

sensors

Intelligent Transportation Related Complex Systems and Sensors

Edited by

Kyandoghene Kyamakya, Jean Chamberlain Chedjou,
Fadi Al-Machot, Ahmad Haj Mosa and Antoine Bagula

Printed Edition of the Special Issue Published in *Sensors*

Intelligent Transportation Related Complex Systems and Sensors

Intelligent Transportation Related Complex Systems and Sensors

Editors

Kyandoghene Kyamakya
Jean Chamberlain Chedjou
Fadi Al-Machot
Ahmad Haj Mosa
Antoine Bagula

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin



Editors

Kyandogherye Kyamakya
Institute for Smart Systems
Technologies, Universitaet
Klagenfurt
Austria

Jean Chamberlain Chedjou
Universitaet Klagenfurt
Austria

Fadi Al-Machot
Department of Applied
Informatics, Universitaet
Klagenfurt
Austria

Ahmad Haj Mosa
Universitaet Klagenfurt
Austria

Antoine Bagula
University of the Western Cape,
ISAT Laboratory
South Africa

Editorial Office

MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Sensors* (ISSN 1424-8220) (available at: https://www.mdpi.com/journal/sensors/special_issues/Transportation_Systems).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

| |
|--|
| LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. <i>Journal Name</i> Year , <i>Volume Number</i> , Page Range. |
|--|

ISBN 978-3-0365-0848-1 (Hbk)

ISBN 978-3-0365-0849-8 (PDF)

© 2021 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license, which allows users to download, copy and build upon published articles, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of our publications.

The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons license CC BY-NC-ND.

Contents

| | |
|--|-----|
| About the Editors | ix |
| Kyandoghene Kyamakya, Jean Chamberlain Chedjou, Fadi Al-Machot, Ahmad Haj Mosa and Antoine Bagula Intelligent Transportation Related Complex Systems and Sensors Reprinted from: <i>Sensors</i> 2021 , <i>21</i> , 2235, doi:10.3390/s21062235 | 1 |
| Fernando Álvarez-Bazo, Santos Sánchez-Cambronero, David Vallejo, Carlos Glez-Morcillo, Ana Rivas and Inmaculada Gallego A Low-Cost Automatic Vehicle Identification Sensor for Traffic Networks Analysis Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 5589, doi:10.3390/s20195589 | 9 |
| Víctor Corcoba Magaña, Wilhelm Daniel Scherz, Ralf Seepold, Natividad Martínez Madrid, Xabiel García Pañeda and Roberto Garcia The Effects of the Driver’s Mental State and Passenger Compartment Conditions on Driving Performance and Driving Stress Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 5274, doi:10.3390/s20185274 | 37 |
| Jacek Oskarbski, Tomasz Kamiński, Kyandoghene Kyamakya, Jean Chamberlain Chedjou, Karol Źarski and Małgorzata Pedzierska Assessment of the Speed Management Impact on Road Traffic Safety on the Sections of Motorways and Expressways Using Simulation Methods Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 5057, doi:10.3390/s20185057 | 69 |
| Qinyu Sun, Yingshi Guo, Rui Fu, Chang Wang and Wei Yuan Human-Like Obstacle Avoidance Trajectory Planning and Tracking Model for Autonomous Vehicles That Considers the Driver’s Operation Characteristics Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 4821, doi:10.3390/s20174821 | 103 |
| Yanbin Guo, Lulu Huang, Yingbin Liu, Jun Liu and Guoping Wang Establishment of the Complete Closed Mesh Model of Rail-Surface Scratch Data for Online Repair Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 4736, doi:10.3390/s20174736 | 131 |
| Mark Richard. Wilby, Juan José Vinagre Díaz, Rubén Fernández Pozo, Ana Belén Rodríguez González, José Manuel Vassallo, and Carmen Sánchez Ávila Data-Driven Analysis of Bicycle Sharing Systems as Public Transport Systems Based on a Trip Index Classification Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 4315, doi:10.3390/s20154315 | 151 |
| Mariusz Kiec, Carmelo D’Agostino and Sylwia Pazdan Impact on Road Safety and Operation of Rerouting Traffic in Rural Travel Time Information System Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 4145, doi:10.3390/s20154145 | 167 |
| Mohammad A. Aljamal, Hossam M. Abdelghaffar, and Hesham A. Rakha Estimation of Traffic Stream Density Using Connected Vehicle Data: Linear and Nonlinear Filtering Approaches Reprinted from: <i>Sensors</i> 2020 , <i>20</i> , 4066, doi:10.3390/s20154066 | 183 |

Jinghui Wang and Hesham Rakha *

Empirical Study of Effect of Dynamic Travel Time Information on Driver Route Choice Behavior
Reprinted from: *Sensors* **2020**, *20*, 3257, doi:10.3390/s20113257 199

Zheng Zhang, Haiqing Liu, Laxmisha Rai and Siyi Zhang

Vehicle Trajectory Prediction Method Based on License Plate Information Obtained from Video-Imaging Detectors in Urban Road Environment
Reprinted from: *Sensors* **2020**, *20*, 1258, doi:10.3390/s20051258 213

Yifeng Wang, Ping Wang, Qihang Wang, Zhengxing Chen and Qing He

Using Vehicle Interior Noise Classification for Monitoring Urban Rail Transit Infrastructure
Reprinted from: *Sensors* **2020**, *20*, 1112, doi:10.3390/s20041112 231

Muhammad Zahid, Yangzhou Chen, Arshad Jamal and Muhammad Qasim Memon

Short Term Traffic State Prediction via Hyperparameter Optimization Based Classifiers
Reprinted from: *Sensors* **2020**, *20*, 685, doi:10.3390/s20030685 249

Ronghua Du, Gang Qiu, Kai Gao, Lin Hu and Li Liu

Abnormal Road Surface Recognition Based on Smartphone Acceleration Sensor
Reprinted from: *Sensors* **2020**, *20*, 451, doi:10.3390/s20020451 271

Donggyun Kim, MyeonGyu Jeong, ByungGuk Bae and Changsun Ahn

Design of a Human Evaluator Model for the Ride Comfort of Vehicle on a Speed Bump Using a Neural Artistic Style Extraction
Reprinted from: *Sensors* **2019**, *19*, 5407, doi:10.3390/s19245407 289

Mohammad Aljamal, Hossam Abdelghaffar and Hesham Rakha

Developing a Neural-Kalman Filtering Approach for Estimating Traffic Stream Density Using Probe Vehicle Data
Reprinted from: *Sensors* **2019**, *19*, 4325, doi:10.3390/s19194325 303

Vahid Tavakkoli, Jean Chamberlain Chedjou and Kyandoghere Kyamakya

A Novel Recurrent Neural Network-Based Ultra-Fast, Robust, and Scalable Solver for Inverting a “Time-Varying Matrix”
Reprinted from: *Sensors* **2019**, *19*, 4002, doi:10.3390/s19184002 321

Mariano Gallo, Giuseppina De Luca, Luca D’Acierno and Marilisa Botte

Artificial Neural Networks for Forecasting Passenger Flows on Metro Lines
Reprinted from: *Sensors* **2019**, *19*, 3424, doi:10.3390/s19153424 341

Adrian Fazekas and Markus Oeser

Spatio-Temporal Synchronization of Cross Section Based Sensors for High Precision Microscopic Traffic Data Reconstruction
Reprinted from: *Sensors* **2019**, *19*, 3193, doi:10.3390/s19143193 355

Qiming Wang, Tao Sun, Zhichao Lyu and Dawei Gao

A Virtual In-Cylinder Pressure Sensor Based on EKF and Frequency-Amplitude-Modulation Fourier-Series Method
Reprinted from: *Sensors* **2019**, *19*, 3122, doi:10.3390/s19143122 377

Yang Wang, Liqiang Zhu, Zujun Yu and Baoqing Guo

An Adaptive Track Segmentation Algorithm for a Railway Intrusion Detection System
Reprinted from: *Sensors* **2019**, *19*, 2594, doi:10.3390/s19112594 397

Hossam M. Abdelghaffar and Hesham A. Rakha
A Novel Decentralized Game-Theoretic Adaptive Traffic Signal Controller: Large-Scale Testing
Reprinted from: *Sensors* **2019**, *19*, 2282, doi:10.3390/s19102282 **419**

Tianyang Dong, Guoqing Zhao, Jiamin Wu, Yang Ye and Ying Shen
Efficient Traffic Video Dehazing Using Adaptive Dark Channel Prior and
Spatial–Temporal Correlations
Reprinted from: *Sensors* **2019**, *19*, 1593, doi:10.3390/s19071593 **439**

**Amir Mehdizadeh, Miao Cai, Qiong Hu, Mohammad Ali Alamdar Yazdi, Nasrin Mohabbati
Kalejahi, Aleksandr Vinel, Steven Rigdon, Karen Davis and Fadel Megahed**
A Review of Data Analytic Applications in Road Traffic Safety. Part 1: Descriptive and
Predictive Modeling
Reprinted from: *Sensors* **2020**, *20*, 1107, doi:10.3390/s20041107 **459**

About the Editors

Kyandoghere Kyamakya is currently a Full Professor of Transportation Informatics and the Deputy Director of the Institute for Smart Systems Technologies at Universitaet Klagenfurt in Austria. He is actively conducting research involving modeling, simulation, and test-bed evaluations for a series of concepts amongst others in the context of intelligent transportation systems. In the research addressing transportation systems, a series of fundamental and theoretical tools from the fields of applied mathematics, electronics, and computer science is either extensively exploited or a source of inspiration for innovative solutions and concepts, including nonlinear dynamics, systems science, machine learning/deep learning, nonlinear image processing, and neurocomputing. He has co-edited more than 6 books, has published more than 100 journal papers, and hundreds of conference papers.

Fadi Al-Machot finished his PhD in Computer Science at Alpen-Adria University Klagenfurt in November 2013 and his habilitation in Applied Computer Science at the University of Lübeck in 2020. As a researcher, he developed different algorithms and approaches in the areas of complex event detection in multimodal sensor networks, advanced driver assistance systems, human cognitive reasoning, and human activity and emotion recognition. His work is patented and published in different international conferences and journals. He is currently a senior data scientist at Leibniz Lung Center—Research Center Borstel.

Ahmad Haj Mosa A researcher and AI developer in the team of Digital Services at PwC Austria. He is also a researcher and a lecturer at the Institute for Smart System Technology (IST) at the University of Klagenfurt, Austria. His research area focus lies on Augmented Intelligence and Explainable Deep Learning, and Self-Driving Cars. And his research interests include machine vision, machine learning, applied mathematics, and neurocomputing. He has developed a variety of methods in the scope of human-machine interaction and pattern recognition.

Antoine Bagula completed his PhD in Communication Systems at the Royal Institute of Technology (KTH), Stockholm, Sweden, and 2 MSc degrees (Computer Engineering—Université Catholique de Louvain (UCL), Belgium and Computer Science—University of Stellenbosch (SUN), South Africa). He is currently a Full Professor and Head of the Department of Computer Science at the University of the Western Cape (UWC), where he also leads the Intelligent Systems and Advanced Telecommunication (ISAT) laboratory. He is a well-published scientist in his research field. His current research interests include Data Engineering including Big Data Technologies, Cloud/Fog Computing and Network Softwarization (e.g. NFV and SDN); The Internet of Things (IoT) including the Internet-of-Things in Motion and the Tactile Internet, Data Science including Artificial Intelligence, Machine Learning with their applications in Big Data Analytics; and Next Generation Networks including 4G/5G.

Intelligent Transportation Related Complex Systems and Sensors

Kyandoghere Kyamakya ^{1,*}, Jean Chamberlain Chedjou ¹, Fadi Al-Machot ², Ahmad Haj Mosa ¹ and Antoine Bagula ³

¹ Institute for Smart Systems Technologies, Universität Klagenfurt, A9020 Klagenfurt, Austria; Jean.Chedjou@aau.at (J.C.C.); ahmad.haj.mosa@pwc.com (A.H.M.)

² Department of Applied Informatics, Universität Klagenfurt, A9020 Klagenfurt, Austria; Fadi.ALMachot@aau.at

³ ISAT Laboratory, University of the Western Cape, Bellville 7535, South Africa; abagula@uwc.ac.za

* Correspondence: kyandoghere.kyamakya@aau.at

Building around innovative services related to different modes of transport and traffic management, intelligent transport systems (ITSs) are being widely adopted worldwide to improve the efficiency and safety of the transportation system. They enable users to be better informed and make safer, more coordinated, and smarter decisions on the use of transport networks. Current ITSs are complex systems, made up of several components/sub-systems characterized by time-dependent interactions among themselves. Some examples of these transportation-related complex systems include road traffic sensors, autonomous/automated cars, smart cities, smart sensors, virtual sensors, traffic control systems, smart roads, logistics systems, smart mobility systems, and many others that are emerging from niche areas. The efficient operation of these complex systems require (i) efficient solutions to the issues of sensors/actuators used to capture and control the physical parameters of these systems as well as the quality of data collected from these systems; (ii) tackling complexities using simulations and analytical modelling techniques; (iii) applying optimization techniques to improve the performance of these systems.

This book out of the Special Issue on Intelligent Transportation Related Complex Systems and Sensors emerges as a result of the crucial need for improving transportation support in different domains and parts of the world by finding solutions to the rich yet non-trial and unexpected behaviour resulting from the complexity of ITS. It includes twenty-four papers, which cover scientific concepts, frameworks, architectures, and various other ideas on analytics, trends, and applications of transportation related data.

The 24 papers/chapters contained in this book propose solutions to various issues and broadly grouped into four classes/parts, namely, the following ones:

1. Traffic safety and security.
2. Autonomy and path planning.
3. Traffic density.
4. Traffic analytics and prediction.

1. Safety and Security

In [1], the authors considered the research gap found in the observability in traffic networks. The work addressed non-definitive plate scanning problems, by using sensors embedded into elements across traffic network to enable technicians reach better conclusions when they deal with traffic network analysis. This is an area of research with very limited number of studies, and the authors in this work proposed (i) an architecture for deploying temporary low-cost sensors across city streets as an alternative of rubber hoses commonly used in elaborate urban mobility plans; (ii) a design for these low-cost, low-energy sensors themselves; (iii) an ideal sensor location model for establishing the best set of network links to achieve the study's aims. To demonstrate the viability of these



Citation: Kyamakya, K.; Chedjou, J.C.; Al-Machot, F.; Haj Mosa, A.; Bagula, A. Intelligent Transportation Related Complex Systems and Sensors. *Sensors* **2021**, *21*, 2235. <https://doi.org/10.3390/s21062235>

Received: 11 March 2021

Accepted: 18 March 2021

Published: 23 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

contributions, a case study with the installation of a set of proposed devices is used by the authors to demonstrate the viability of their contributions.

In [2], the effects of drivers' mental state and passenger compartment conditions on driving performance and driving stress were considered. Factors such as the human error, cognitive capacity, and levels of CO₂ concentration, humidity, and temperature within the vehicles were analysed in terms of their impact on driving. The experimental setting of the study included a survey with 50 drivers, 25 min of drive time using a driving simulator. Information about the drivers' mental state and stress levels were monitored during the test using biometric sensors, while suitable sensors were used to monitor environmental conditions—temperature, humidity, and CO₂ levels. The study revealed that i) the initial level of stress and tiredness of the driver can have a strong impact on stress, driving behaviour, and fatigue and ii) elements such as state of the mind (sadness or happiness) and the conditions of the interior of the vehicle can also impaired driving and affect compliance with traffic regulations.

Reference [3] addressed the issues associated with expert assessments and statistical studies commonly used to evaluate the impact of intelligent transport system (ITS) services on road safety. The work built upon an approach based on surrogate safety measures calculated and calibrated with the use of simulation techniques and a driving simulator to achieve traffic efficiency and road safety. Experiments were conducted to measure the influence of selected scenarios of variable speed limits on the efficiency and safety of traffic on sections of motorways and expressways in various traffic conditions. The presented studies confirmed the positive impact of variable speed limits (VSLs) on the level of road safety and traffic efficiency. Recommendations were then given as well as areas of potential further research.

Building upon an intelligent traffic control system installed in Poland, [4] addressed the issue of safety and traffic operation on roads by (i) analysing the safety level of the entire road network when traffic is rerouted to paths along different road categories, intersections, road environments, and densities of access points and (ii) presenting a comparison between traffic operation and road safety performance, with the aim of predicting the number of crashes for each possible route when considering travel time and delay. The results of the study allow for maximizing safety or traffic operation characteristics, providing an effective tool for the management of rural road systems.

In [5], a review of data analytic applications in road traffic safety was done. The aim was to reduce the start-up burden of data collection and descriptive analytics for statistical modelling and route optimization of risk associated with motor vehicles. Building upon a data-driven bibliometric analysis, the study showed that literature is divided into two disparate research streams: (a) predictive or explanatory models that attempt to understand and quantify crash risk based on different driving conditions and (b) optimization techniques that focus on minimizing crash risk through route/path-selection and rest-break scheduling. Translation of research outcomes between these two streams was limited. The study also (i) presented publicly available high-quality data sources (different study designs, outcome variables, and predictor variables) and descriptive analytic techniques (data summarization, visualization, and dimension reduction) that could be used to achieve safer-routing and provide code to facilitate data collection/exploration by practitioners/researchers; (ii) reviewed the statistical and machine learning models used for crash risk modelling; (iii) showed that (near) real-time crash risks are rarely considered.

Reference [6] also reviewed data analytic applications in road traffic safety with the objective of filling the gap left by [5] between the predictive and prescriptive models pertaining to crash risk prediction and minimization, respectively. The paper reviewed and categorized the optimization and prescriptive analytic models that focused on minimizing crash risk. The review showed that, though majority of works in this segment of the literature are related to the hazardous materials (hazmat) trucking problems, with some exceptions, many studies can also be utilized in non-hazmat scenarios. In an effort to highlight the effect of crash risk prediction models on the accumulated risk obtained

from the prescriptive model, the review presented a simulated example where four risk indicators (obtained from logistic regression, Poisson regression, XGBoost, and neural network) were used in the k -shortest path algorithm. An example demonstrating two major designed takeaways were also presented in the paper. The first revealed that the shortest path may not always result in the lowest crash risk, while the other showed that a similarity in overall predictive performance may not always translate to similar outcomes from the prescriptive models.

With regards to intrusion detection, the work done in [7] revisited the issue of labelling of alarm regions in video surveillance-based intrusion detection systems. The authors proposed a three step adaptive segmentation algorithm to delineate the boundary of the track area with very light computation burden. In the first step of the algorithm, the image was segmented into fragmented regions. During this step, an optimal set of Gaussian kernels with adaptive directions for each specific scene was calculated using Hough transformation to reduce the redundant calculation in the evaluation of the boundary weight for generating the fragmented regions. At the second step of the algorithm, the fragmented regions were combined into local areas using a new clustering rule based on the region's boundary weight and size. Lastly, a classification network is used to recognize the tracked area among all local areas. To achieve fast and accurate classification, a simplified convoluted neural network (based on pre-trained convolution kernels) and a loss function (that can enhance the diversity of the feature maps) were used. Obtained results showed that the proposed method found an effective balance between the segmentation precision, calculation time, and hardware cost of the system.

On a related issue of traffic videos, [8] sought to tackle the issue of restoring traffic videos with different degrees of haziness in a real-time and adaptive manner. They work proposed an efficient traffic video dehazing method using adaptive dark channel prior and spatial-temporal correlations. The work used a haziness flag to measure the degree of haziness in images based on dark channel prior. It then got the adaptive initial transmission value by establishing the relationship between the image contrast and haziness flag. Additionally, the method took advantage of the spatial and temporal correlations among traffic videos to speed up the dehazing process and optimize the block structure of restored videos. Extensive experimental results showed that the proposed method had superior haze removing and colour balancing capabilities for the images with different degrees of haze and was indeed able to restore degraded videos in real time.

2. Autonomy and Path Planning

The development of human-like autonomous driving systems has in recent times gained increased attention from both technology companies and academic institutions alike, as it has the potential to improve the interpretability and acceptance of autonomous vehicle systems. [9] addressed the research gap found in the planning of a safe and human-like obstacle avoidance trajectory, which is one of the critical issues for the development of autonomous vehicles (AVs). The paper proposed an automatic obstacle avoidance system that focused on the obstacle avoidance characteristics of human drivers. Different models for trajectory planning and trajectory tracking were proposed and tested through off-line simulation based on CarSim/Simulink. Simulation results revealed that the proposed trajectory planning and tracking controllers were more human-like under the premise of ensuring the safety and comfort of the obstacle avoidance operation, thus providing a foundation for the development of AVs.

The work done in [10] relied on the license plate data obtained from the massive volume of information collected from video imaging to predict vehicle trajectory. The paper proposed a real-time vehicle trajectory prediction method based on (i) historical trip rules extracted from vehicle license plate data in an urban road environment; (ii) vehicle trip chain acquired on the basis of the topologic graph of the road network, channelization of intersections, and the driving status information at intersections; (iii) a trip chain compensation method based on the Dijkstra algorithm to complement missing data in the

original vehicle license plate. The proposed method was tested using realistic road traffic scenarios with actual vehicle license plate data. Good trajectory prediction results were obtained and revealed an average accuracy of 0.72 for one-step prediction when there are only 200 historical training data samples.

In [11], the authors studied the effect of travel time information on day-to-day driver route choice behaviour. A real-world experimental study designed to have participants repeatedly choose between two alternative routes for five origin–destination pairs over multiple days was performed. The participants were provided with dynamically updated travel time information (average travel time and travel time variability) during the experiment. The results of the study revealed that (i) historical travel time information enhanced behavioural rationality by 10% on the average; (ii) expected travel time information was more effective than travel time variability information in enhancing rational behaviour when drivers had limited experiences; (iii) when drivers lack experience, the faster less reliable route was more attractive than the slower more reliable route; (iv) with cumulative experiences, drivers become more willing to take the more reliable route given that they are reluctant to become risk seekers once experience is gained; (v) the effect of information on driver behaviour differed significantly by participant and trip, which to a large extent, depended on personal traits and trip characteristics.

3. Traffic Density

References [12–14] focused on vehicular and/or human traffic. In [12], the authors estimated traffic stream density by using data from solely connected vehicle (CV) and applying a nonlinear filtration. A particle filter (PF) was developed to produce reliable traffic density estimates using the CV's travel-time measurements. Traffic flow continuity was then used to derive the state equation, while the measurement equation was derived from the hydrodynamic traffic flow relationship. A comparison was done against two PF filtering approaches, namely Kalman filter (KF) and adaptive KF (AKF). Obtained results revealed that (i) the three techniques produce accurate estimates—with the KF, surprisingly, being the most accurate of the three techniques (ii) the accuracy of the PF estimations increased as the number of particles increased and (iii) the accuracy of the density estimate increased as the level of CV market penetration increased.

In a similar work, [13] used the adaptive Kalman filter (AKF) to reliably estimate traffic vehicle count by considering real-time characteristics of system noise. Using only real-time probe vehicle data, the AKF is demonstrated to outperform the traditional Kalman filter by reducing the prediction error by up to 29%. A novel approach was also introduced by the paper wherein AKF was combined with a neural network (AKFNN) to enhance the vehicle count estimates. The results showed that the accuracy of vehicle count estimates was significantly improved when AKFNN was used by up to 26% compared to pure AKF, but the AKF was more sensitive to the initial conditions.

Reference [14] slightly shifted focus to passenger count in rail transportation. The work revisited the issue of forecasting passenger flows on metro lines by proposing the use of artificial neural networks (ANNs). The authors forecasted the number of passenger flows on a metro to be a function of passenger counts at station turnstiles. The study assumes that metro station turnstiles record the number of passengers entering by means of an automatic counting system and that these data are available every few minutes. These data are then used to estimate the onboard passengers on each track section of the line (i.e., between two successive stations). The ANNs were trained using simulation data obtained with a dynamic loading procedure of the rail line and tested using real-scale case scenario of Line 1 of the Naples metro system in Italy. The numerical results showed that the proposed approach was able to forecast the flows on metro sections with satisfactory levels of precision.

4. Analytics and Prediction

These papers focus on analysis of transportation data for the purpose of drawing insights or predicting future events.

Reference [15] addresses the requirement of high level of detail and coverage in traffic data acquisition in the next generation of intelligent transportation systems (ITS). The authors presented appropriate methods with consideration of realistic scale and accuracy conditions of the original data acquisition. The study relied on datasets consisting of timestamp and speed for each individual vehicle used as input data and proposed as a first step, a closed formulation for a sensor offset estimation algorithm with simultaneous vehicle registration. Building upon this initial step, the datasets are fused to reconstruct microscopic traffic data using quintic Beziér curves. The derived trajectories were then used to thoroughly investigate the dependency of the results on the accuracy of the individual sensors. It was found that the proposed method enhanced the usability of common cross-section-based sensors by enabling the deriving of non-linear vehicle trajectories without the necessity of precise prior synchronization.

The focus on [16] lies on the challenging issue of accurate modelling of short-term traffic prediction due to its intricate characteristics, stochastic, and dynamic traffic processes. Existing works in this area followed different modelling approaches focusing on speed, density, or data volume. The study however used (i) state-of-the-art models via hyper-parameter optimization using different machine learning classifiers such as local deep support vector machine (LD-SVM), decision jungles (DJ), multi-layers perceptron (MLP), and CN2 rule induction and (ii) traffic states evaluation based on traffic attributes such as level of service (LOS) horizons and simple if-then rules at different time intervals. The findings of the study revealed that (i) hyper-parameter optimization via random sweep yielded superior results; (ii) the overall prediction performances obtained an average improvement by over 95%, such that the decision jungle and LD-SVM achieved an accuracy of 98.2 and 97.5%, respectively; (iii) the robustness and superior performances of decision jungles (DJ) over other methods.

Still on short distance traffic, [17] proposed a solution to the issue of bicycle sharing systems (BSSs), which are traditionally conceived as a last-mile complement to the public transport system. The paper demonstrated that BSSs can be seen as a public transport system in their own right and built a mathematical framework for the classification of BSS trips. It also used trajectory information to create the trip index, which characterizes the intrinsic purpose of the use of BSS as transport or leisure. By applying the proposed methodology to empirical data from BiciMAD (the public BSS in Madrid, Spain), the authors conducted experiments, which revealed that the obtained trip index was able to correctly distinguish between transport and leisure categories using exhibited statistical and operational features.

In identifying irregular/abnormal road surface conditions, the authors in [18] proposed an efficient and low-cost model, which used the vibration and acceleration sensors in smartphones. The study used an improved Gaussian background model to extract the features of the abnormal pavement and the k-nearest neighbour (kNN) algorithm to distinguish the abnormal pavement types, including pothole and bump. The study also included as feature the influence of vehicles with different suspension characteristics on the detection threshold and proposed an adaptive adjustment mechanism based on vehicle speed. In determining the accuracy of the proposed model, the authors bench-marked their algorithms against real-life field investigation. Obtained results showed that the vibration and acceleration information could indeed reveal the condition of the road surface with an accuracy of up to 96.03% for road surface pothole and 94.12% for road surface bumps. These results show that the proposed road surface recognition method could potentially be utilized to replace special patrol vehicles for timely and low-cost road maintenance.

The study in [19] revisits the costly and time-consuming issue of subjective evaluation of vehicle ride comfort for vehicle development. In contrast to most of the approaches that rely on the use of a regression model between objective metrics and subjective ratings

with an accuracy that is highly dependent on the selection of the objective metrics, the solution proposed in the study used a method that built a correlation model between measurements and subjective evaluations without using predefined features or objective metrics. Using a combination of (i) numerical representation of ride comfort extracted from raw signals based on the idea of the artistic style transfer method, (ii) a correlation model designed based on the extracted numerical representation and subjective ratings, and (iii) a pre-trained neural network, the proposed model was proven to provide much better accuracy than any other correlation models in the literature.

Focusing on vehicle interior noise classification, the study in [20] developed a multi-classification model that has the potential to be used to analyse the causes of abnormal noise using statistical methods and evaluate the effect of rail maintenance work. The work first developed a multi-source data (audio, acceleration, and angle rate) collection framework via built-in sensors in smartphone. It then used the Shannon entropy based on a 1-second window to segment the time-series signals. Forty-five features extracted from the time and frequency domains were used to establish the classifier. The study investigated the effects of balancing the training dataset with the synthetic minority oversampling technique (SMOTE). It then compared and analysed the classification results of importance-based and mutual information-based feature selection methods using a feature set consisting of the top 10 features by importance score. Comparisons with other classifiers indicated that the proposed XGBoost-based classifier ran fast, while maintaining good accuracy.

Reference [21] aimed at bridging the research gap found in the field of rail surface scratching data, where only a limited number of studies have addressed the issue of complete closed mesh model. The paper proposed a model based on a novel triangulation algorithm relying on the topological features of the point-cloud model (PCM) of scratch data. These data were obtained by implementing a scratch-data-computation process following a rail-geometric-feature-fused algorithm of random sample consensus (RANSAC) constructed by 3D laser vision. Using a method that is universal for all types of normal-speed rails in China, the paper presented experimental results showing that the proposed method could accurately acquire the complete closed mesh models of scratch data of one meter of 50 Kg/m-rails within 1 min.

In contrast to the intrusive type cylinder pressure sensor, which has high cost, low reliability, and short life due to severe working environments, [22] proposed the cylinder pressure identification method also called virtual cylinder pressure sensor. In this work, frequency-amplitude modulated Fourier series (FAMFS) and extended-Kalman-filter-optimized (EKF) engine model were used as low-cost, real-time, non-invasive, and highly accuracy alternative solutions. The paper established an iterative speed model based on burning theory and law of energy conservation and used the efficiency coefficient to represent the operating state of engine from fuel to motion. The iterative speed model was associated with the throttle opening value and the crankshaft load. EKF was used to estimate the optimal output of this iteration model, which was utilized to separately compute the frequency and amplitude of the cylinder pressure cycle-to-cycle. A standard engine's working cycle, identified by the 24th order Fourier series was then determined. Using frequency and amplitude obtained from the iteration model to modulate the Fourier series yielded a complete pressure model. Using a commercial engine (EA211) provided by the China FAW Group corporate R&D centre, the proposed method was verified through a test that revealed its high accuracy and real-time capability, with an error percentage for speed of less than 9.6% and the cumulative error percentage of cylinder pressure less than 1.8% when A/F Ratio coefficient is setup at 0.85.

Building upon game theory, [23] presented a novel de-centralized flexible phasing scheme, cycle-free, adaptive traffic signal controller that uses a Nash bargaining game-theoretic framework. The Nash bargaining algorithm was used to optimize the traffic signal timings at each signalized intersection by modelling each phase as a player in a game, where players cooperate to reach a mutually agreeable outcome. The controller is implemented and tested in the integration microscopic traffic assignment and simulation software. Its

performance was then compared with traditional decentralized adaptive cycle length and phase split traffic signal controller, a centralized, fully coordinated, adaptive phase split, cycle length, and offset optimization controller. Using comparisons conducted in the town of Blacksburg, Virginia (38 traffic signalized intersections) and in downtown Los Angeles, California (457 signalized intersections), the results showed significant potential benefits of using the proposed controller over other state-of-the-art centralized and de-centralized adaptive traffic signal controllers on large-scale networks both during uncongested and congested conditions.

Building upon dynamical methods to realize a time-varying matrix inversion, [24] relies on a system of coupled ordinary differential equations (ODEs), constituting a recurrent neural network (RNN) model as a universal modelling framework for realizing a matrix inversion, provided the matrix is invertible. The study in this paper builds around this framework to propose and investigate a new combined/extended method for time-varying matrix inversion that extends both the gradient neural network (GNN) and the Zhang neural network (ZNN) concepts. The newly proposed model has (i) proven that it has exponential stability according to Lyapunov theory, (ii) a much better theoretical convergence speed compared to other previous related methods (namely GNN, ZNN, Chen neural network, and integration-enhanced Zhang neural network or IEZNN), and (iii) a better practical convergence rate when compared to the other models on practical examples and both their respective measured convergence and error rates.

Author Contributions: All authors: participation to the redaction work and overall text reviews; K.K.: general coordination; A.B.: final fine-tunings. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Álvarez-Bazo, F.; Sánchez-Cambronero, S.; Vallejo, D.; Glez-Morcillo, C.; Rivas, A.; Gallego, I. A Low-Cost Automatic Vehicle Identification Sensor for Traffic Networks Analysis. *Sensors* **2020**, *20*, 5589. [[CrossRef](#)] [[PubMed](#)]
2. Magaña, V.C.; Scherz, W.D.; Seepold, R.; Madrid, N.M.; Pañeda, X.G.; Garcia, R. The Effects of the Driver's Mental State and Passenger Compartment Conditions on Driving Performance and Driving Stress. *Sensors* **2020**, *20*, 5274. [[CrossRef](#)] [[PubMed](#)]
3. Oskarski, J.; Kamiński, T.; Kyamakya, K.; Chedjou, J.C.; Żarski, K.; Pędzińska, M. Assessment of the Speed Management Impact on Road Traffic Safety on the Sections of Motorways and Expressways Using Simulation Methods. *Sensors* **2020**, *20*, 5057. [[CrossRef](#)]
4. Kiec, M.; D'Agostino, C.; Pazdan, S. Impact on Road Safety and Operation of Rerouting Traffic in Rural Travel Time Information System. *Sensors* **2020**, *20*, 4145. [[CrossRef](#)]
5. Mehdizadeh, A.; Cai, M.; Hu, Q.; Yazdi, M.A.A.; Mohabbati-Kalejahi, N.; Vinel, A.; Rigdon, S.E.; Davis, K.C.; Megahed, F.M. A Review of Data Analytic Applications in Road Traffic Safety. Part 1: Descriptive and Predictive Modeling. *Sensors* **2020**, *20*, 1107. [[CrossRef](#)]
6. Hu, Q.; Cai, M.; Mohabbati-Kalejahi, N.; Mehdizadeh, A.; Yazdi, M.A.A.; Vinel, A.; Rigdon, S.E.; Davis, K.C.; Megahed, F.M. A Review of Data Analytic Applications in Road Traffic Safety. Part 2: Prescriptive Modeling. *Sensors* **2020**, *20*, 1096. [[CrossRef](#)]
7. Wang, Y.; Zhu, L.; Yu, Z.; Guo, B. An Adaptive Track Segmentation Algorithm for a Railway Intrusion Detection System. *Sensors* **2019**, *19*, 2594. [[CrossRef](#)]
8. Dong, T.; Zhao, G.; Wu, J.; Ye, Y.; Shen, Y. Efficient Traffic Video Dehazing Using Adaptive Dark Channel Prior and Spatial-Temporal Correlations. *Sensors* **2019**, *19*, 1593. [[CrossRef](#)]
9. Sun, Q.; Guo, Y.; Fu, R.; Wang, C.; Yuan, W. Human-Like Obstacle Avoidance Trajectory Planning and Tracking Model for Autonomous Vehicles That Considers the Driver's Operation Characteristics. *Sensors* **2020**, *20*, 4821. [[CrossRef](#)]
10. Zhang, Z.; Liu, H.; Rai, L.; Zhang, S. Vehicle Trajectory Prediction Method Based on License Plate Information Obtained from Video-Imaging Detectors in Urban Road Environment. *Sensors* **2020**, *20*, 1258. [[CrossRef](#)] [[PubMed](#)]
11. Wang, J.; Rakha, H. Empirical Study of Effect of Dynamic Travel Time Information on Driver Route Choice Behavior. *Sensors* **2020**, *20*, 3257. [[CrossRef](#)]
12. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Estimation of Traffic Stream Density Using Connected Vehicle Data: Linear and Nonlinear Filtering Approaches. *Sensors* **2020**, *20*, 4066. [[CrossRef](#)]
13. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Developing a Neural-Kalman Filtering Approach for Estimating Traffic Stream Density Using Probe Vehicle Data. *Sensors* **2019**, *19*, 4325. [[CrossRef](#)]

14. Gallo, M.; De Luca, G.; D’Acierno, L.; Botte, M. Artificial Neural Networks for Forecasting Passenger Flows on Metro Lines. *Sensors* **2019**, *19*, 3424. [[CrossRef](#)] [[PubMed](#)]
15. Fazekas, A.; Oeser, M. Spatio-Temporal Synchronization of Cross Section Based Sensors for High Precision Microscopic Traffic Data Reconstruction. *Sensors* **2019**, *19*, 3193. [[CrossRef](#)] [[PubMed](#)]
16. Zahid, M.; Chen, Y.; Jamal, A.; Memon, M.Q. Short Term Traffic State Prediction via Hyperparameter Optimization Based Classifiers. *Sensors* **2020**, *20*, 685. [[CrossRef](#)]
17. Wilby, M.R.; Díaz, J.J.V.; Pozo, R.F.; González, A.B.R.; Vassallo, J.M.; Ávila, C.S. Data-Driven Analysis of Bicycle Sharing Systems as Public Transport Systems Based on a Trip Index Classification. *Sensors* **2020**, *20*, 4315. [[CrossRef](#)] [[PubMed](#)]
18. Du, R.; Qiu, G.; Gao, K.; Hu, L.; Liu, L. Abnormal Road Surface Recognition Based on Smartphone Acceleration Sensor. *Sensors* **2020**, *20*, 451. [[CrossRef](#)]
19. Kim, D.; Jeong, M.; Bae, B.; Ahn, C. Design of a Human Evaluator Model for the Ride Comfort of Vehicle on a Speed Bump Using a Neural Artistic Style Extraction. *Sensors* **2019**, *19*, 5407. [[CrossRef](#)] [[PubMed](#)]
20. Wang, Y.; Wang, P.; Wang, Q.; Chen, Z.; He, Q. Using Vehicle Interior Noise Classification for Monitoring Urban Rail Transit Infrastructure. *Sensors* **2020**, *20*, 1112. [[CrossRef](#)]
21. Guo, Y.; Huang, L.; Liu, Y.; Liu, J.; Wang, G. Establishment of the Complete Closed Mesh Model of Rail-Surface Scratch Data for Online Repair. *Sensors* **2020**, *20*, 4736. [[CrossRef](#)] [[PubMed](#)]
22. Wang, Q.; Sun, T.; Lyu, Z.; Gao, D. A Virtual In-Cylinder Pressure Sensor Based on EKF and Frequency-Amplitude-Modulation Fourier-Series Method. *Sensors* **2019**, *19*, 3122. [[CrossRef](#)] [[PubMed](#)]
23. Abdelghaffar, H.M.; Rakha, H.A. A Novel Decentralized Game-Theoretic Adaptive Traffic Signal Controller: Large-Scale Testing. *Sensors* **2019**, *19*, 2282. [[CrossRef](#)] [[PubMed](#)]
24. Tavakkoli, V.; Chedjou, J.C.; Kyamakya, K. A Novel Recurrent Neural Network-Based Ultra-Fast, Robust, and Scalable Solver for Inverting a “Time-Varying Matrix”. *Sensors* **2019**, *19*, 4002. [[CrossRef](#)] [[PubMed](#)]



Article

A Low-Cost Automatic Vehicle Identification Sensor for Traffic Networks Analysis

Fernando Álvarez-Bazo ¹, Santos Sánchez-Cambronero ^{1,*}, David Vallejo ²,
Carlos Glez-Morcillo ², Ana Rivas ¹ and Inmaculada Gallego ¹

¹ Department of Civil and Building Engineering, University of Castilla-La Mancha, 13071 Ciudad Real, Spain; fernando.alvarezbazo@uclm.es (F.Á.-B.); ana.rivas@uclm.es (A.R.); inmaculada.gallego@uclm.es (I.G.)

² Department of Technologies and Information Systems, University of Castilla-La Mancha, 13071 Ciudad Real, Spain; david.vallejo@uclm.es (D.V.); Carlos.Gonzalez@uclm.es (C.G.-M.)

* Correspondence: santos.sanchez@uclm.es

Received: 4 September 2020; Accepted: 25 September 2020; Published: 29 September 2020

Abstract: In recent years, different techniques to address the problem of observability in traffic networks have been proposed in multiple research projects, being the technique based on the installation of automatic vehicle identification sensors (AVI), one of the most successful in terms of theoretical results, but complex in terms of its practical application to real studies. Indeed, a very limited number of studies consider the possibility of installing a series of non-definitive plate scanning sensors in the elements of a network, which allow technicians to obtain a better conclusions when they deal with traffic network analysis such as urbans mobility plans that involve the estimation of traffic flows for different scenarios. With these antecedents, the contributions of this paper are (1) an architecture to deploy low-cost sensors network able to be temporarily installed on the city streets as an alternative of rubber hoses commonly used in the elaboration of urban mobility plans; (2) a design of the low-cost, low energy sensor itself, and (3) a sensor location model able to establish the best set of links of a network given both the study objectives and of the sensor needs of installation. A case of study with the installation of as set of proposed devices is presented, to demonstrate its viability.

Keywords: plate scanning; low-cost sensor; sensor location problem; traffic flow estimation

1. Introduction

1.1. The Purpose and Significance of This Paper

The monitoring of traffic in urban networks, whatever their complexity, is a problem that has been tackled for decades. The aim of this monitoring depends on the case and can involve managing the daily traffic flow to perform urban mobility plans. Regarding the techniques and tools to identify and quantify the vehicles on the network, traditional manual recording has been displaced by more sophisticated techniques due to their economy, and also to collect the traffic information with enough performance and quality. Basically, the emerging techniques consist of a sensor or device able to collect a type of information through its interaction with a vehicle or the infrastructure. Therefore, the sensors used for traffic analysis can be classified in different categories according to their physical characteristics, type of collected information, and position with respect to the network among others. In particular, [1] differs between in-vehicle and in-road sensors. The first are those that allow increasing the performance of the driving and the connectivity of the vehicles with their environment. In this, the concepts of communication between vehicle and the vehicular sensing networks (VSN) are called to be important in the improvement of the quality and operability of transportation systems (see [2,3]). The second are those installed in the transportation network and allows the monitoring of the performance of the system and, according to the extracted information, diagnose the problems, improve the resilient and

operational functioning, and inform the users helping them to make better choices. In this paper we mainly focus in this last. Based on the works of [4,5], in-road sensors for traffic network analysis are classified in two main groups according to the characteristics of the data collected (see Figure 1):

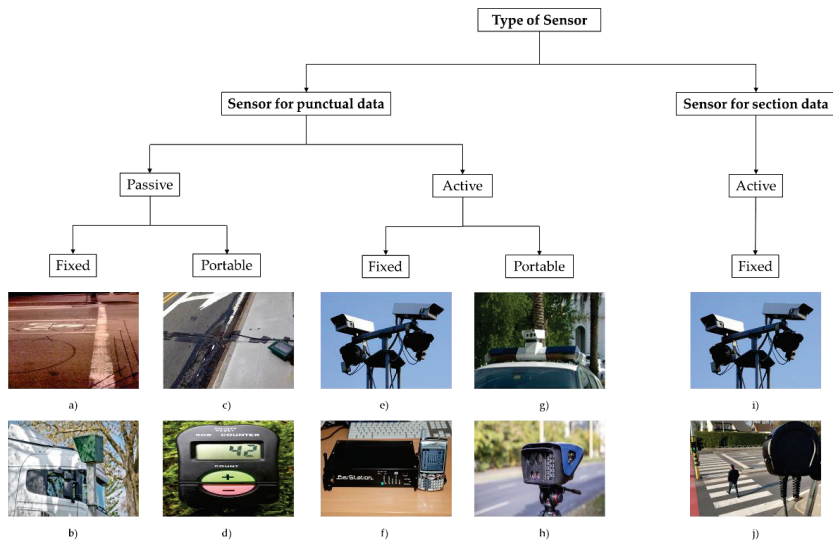


Figure 1. Classification and examples of sensors according to the characteristics of the collected data, their interaction with the vehicle and position on the road: (a) Inductive loop detector; (b) Microwave radar; (c) Rubber hoses detector; (d) Hand electronic counter; (e) and (i) Automatic Number Plate Recognition (ANPR) fixed sensors; (f) Bluetooth sniffer; (g) Police ANPR sensor on vehicle; (h) Police ANPR portable sensor; (j) Bluetooth scanning sensor.

- Sensors for punctual data collect traffic information at a single point of the road, and can be designed to obtain information for each single vehicle (e.g., vehicle presence, speed, or type), or for the vehicles in a defined time interval (e.g., vehicles count, average speed, vehicles occupancy, etc.). In addition, these sensors can be
 - “Passive sensors” do not require any active information provided from a vehicle, i.e., they collect the information when a vehicle is passing in front of the sensor. In particular:
 - “Passive fixed sensors” have a fixed position on the network. This group includes inductive loop detectors, magnetic detectors, pressure detectors, piezoelectric sensors, microwave radars, among others. These sensors are used to manage the traffic and can also be used to elaborate traffic mobility plans using only the already installed fixed sensors if the available budget is limited.
 - “Passive portable sensors” have a fixed position on the network, but they are installed for a defined-short period of time. This group includes counters made with rubber hoses or manual counters that are used for example to elaborate traffic mobility plans completing the information provided by fixed sensors.
 - “Active sensors” require active information from the vehicle to be univocally identified. In fact, these sensors can be included under the term “automatic vehicle identification” (AVI). As well as the passive sensors they can be fixed or portable:
 - “Active fixed sensors” have a fixed position on the network. This group includes automatic number plate recognition (ANPR) sensors, Bluetooth sniffer of bar-coded

tags. Despite these sensors being designed for other purposes far from the traffic network analysis, recent researches have begun to use the data collected by these sensors to estimate traffic flows.

- “Active portable sensors” have to be designed to be installed for a very short period of time to take info from vehicles. As far as we know this kind of sensor has a very limited use for traffic management and more for Police controls, such as ANPR sensors, both those that are temporarily installed on the road and those installed on vehicles (although this latter could be also considered as in-vehicle sensors).
- “Sensors for section data” are those that collect information in different sections on the network providing the number of vehicles traveling from different points in the network, travel times between these points, entrances and exits between reidentification devices, etc. This group includes mainly sensors for license plate recognition, but other approaches that allow vehicle re-identification to match measurements at two (or more) data collection sites that belong to the same vehicle. In general, all these sensors are installed fixed in the network.

The data collected by the sensors can be used for multiple purposes but, since this paper is focused on the topic of traffic flow estimation, only those used as inputs for these models are going to be analyzed. These sensors have to satisfy two objectives: accuracy and coverage [6] and, due to their ease installation and capability of data collection, passive sensors (e.g., fixed loop detectors or portable rubber hoses) have been widely used in mobility studies in large urban areas.

As exposed above, sensors as rubber hoses count the number of vehicles that pass over it, obtaining the needed traffic counts used by traditional methods to estimate origin–destination (O–D), route and link flows on a network. The quality of the results of this estimate may be enough for some cases, but when the technicians or the authorities look for a better degree of observability (or even full observability) of traffic flows to achieve a high quality of estimation, the traffic count data has been proved to be not sufficient. For this, it is expected that these sensors are going to be gradually replaced by new active sensors (as ANPR) that, taking advantage of the available technology and the added value provided by the data, allows the development of models to better estimate the non-observed flows.

1.2. State of the Art of Sensors for ANPR

The automatic number plate recognition (ANPR) system is based on image processing techniques to identify vehicles by their number plates, mainly in real time (for automatic control of traffic rules). In [7] or also in [8] a review is made regarding the most significant research work conducted in this area in recent years.

The general process of automatic number plate recognition can be summarized in several well-defined steps [9,10]. Each step involves a different set of algorithms and/or considerations:

1. Vehicle image capture: This step has a critical impact on the subsequent steps since the final result is highly dependent on the quality of the captured images. The task of correctly capturing images of moving vehicles in real time is complex and subject to many variables of the environment, such as lighting, vehicle speed, and angle of capture.
2. Number plate detection: This step focuses on the detection of the area in which a number plate is expected to be located. Images are stored in digital computers as matrices, each number representing the light intensity of an individual pixel. Different techniques and algorithms give different definitions of a number plate. For example, in the case of edge detection algorithms, the definition could be that a number plate is a “rectangular area with an increased density of vertical and horizontal edges”. This high occurrence of edges is normally caused by the border plates, as well as the limits between the characters and the background of the plate.
3. Character segmentation: Once the plate region has been detected, it is necessary to divide it into pieces, each one containing a different character. This is, along with the plate detection

phase, one of the most important steps of ANPR, as all subsequent phases depend on it. Another similarity with the plate detection process is that there is a wide range of techniques available, ranging from the analysis of the horizontal projection of a plate, to more sophisticated approaches such as the use of neural networks.

4. Character recognition: The last step in the ANPR process consists in recognizing each of the characters that have been previously segmented. In other words, the goal of this step consists in identifying and converting image text into editable text. A number of techniques, such as artificial neural networks, template matching or optical character recognition, are commonly employed to address this challenge. Since character recognition takes place after character segmentation, the recognizer system should deal with ambiguous, noisy or distorted characters obtained from the previous step.

Once the data is collected by the sensors, it has to be properly processed to be used for a great amount of traffic analysis. In particular, focusing on the scope of traffic flow analysis, the data allows to

- Develop models where the observable flows are directly related with the routes followed by the vehicles [11–13]. Since both link flows and O–D flows can be directly derived from route flows, these models are a powerful tool for traffic flow estimation.
- Extract a great amount of information compared with traffic counts which in turn permits developing a model with more flow equations for the same number of variables [14].
- Obtain the full observability of the traffic flows if the budget is sufficient to buy the needed number of sensors [15,16].
- Being combined with other sources of data to improve the results [17].
- Measure other variables as travel times in traffic networks if the location of sensors is adequate [4,18].

An extra step to complement the aforementioned steps is the error recovery that may occur when recognizing plate numbers. This problem is a very important issue to deal with when plate scanning data is used for traffic flow estimation, which some authors have been faced using different approaches [16,19,20].

However, the increasing development of these ANPR systems faces some problems such as: they are fixed sensors and they incur a high cost in terms of hardware [21] (about \$20,000 per camera) and installation and maintenance (about \$4000 per camera). This makes necessary to develop new architectural approaches that allow these types of services to be deployed on a larger scale to face transportation problems such as urban mobility plans. It is worth noting the survey collected in [22], which analyzes the sensors to monitor traffic from the point of view of various criteria, including cost. In this study, it is highlighted that the new sensors tend to be of reduced dimensions, of low energy consumption and that, with a certain number of them, it is possible to design and configure a sophisticated wireless sensor network (WSN) that can cover multiple observations in a certain region [23,24].

Regarding existing software libraries and tools focused on automatic plate recognition, “OpenALPR” (2.5.103) [25] stands out. This open source library, written in C++, is able to analyze images and video streams to automatically identify license plates. The generated output is a text representation that comprises the set of characters associated with each one of the identified plate numbers. The hardware required to run OpenALPR depends on the number of frames per second that the system must handle. From a general point of view, a resolution of 5–10 fps is required for low-speed contexts (under 40 km/h), 10–15 fps for medium speed contexts (40–72.5 km/h), and 15–30 fps for high speed contexts (over 72.5 mph). The library requires significant computing power, with the use of several multi-core processors at 3 GHz to process images at 480 p in low-speed contexts. From the point of view of the success rate, OpenALPR represents the software library with the best results on the market (more than 99% success in a first estimation [26]).

“Plate Recognizer” (1.3.8) [27] offers cloud-based license plate recognition services for projects with special needs such as diffuse, low-light, or low-resolution imaging. The cloud processing pricing

plan offers different configurations per processing volume. There are also other specific purpose platforms for automatic license plate identification in the market, such as “SD-Toolkit” (1.2.50) [28], “Anyline” (24) [29], or the framework “Eocortex” (3.1.39) [30].

In recent years, conventional ANPR systems are strengthening their services through the use of AI techniques [31]. “Intelli-Vision” (San Jose, CA, USA) [32], the company that offers intelligent image analysis services using AI and deep learning techniques, has specific license plate recognition services that can be integrated, via an existing SDK, in Intel processors or provided as a web service in the cloud. The Canadian company “Genetec” (Montreal, Quebec, Canada) [33] announced, at the end of 2019 an ANPR camera that includes an Intel chip designed to feed neural networks improving the identification of license plates at high speed or in bad weather conditions.

Finally, it is very important to keep in mind if the ANPR systems can respect the users’ privacy rights in the entire process in which the vehicle data is collected according to the different locations all along the network [34]. All this means that, when designing a type of sensor that can be implemented in an architecture that serves to monitor the traffic network, the cost criteria for manufacturing and installation, operability and resilience, and information processing must be taken into account.

1.3. Contributions of This Paper

It is being seen how the sensors based on the capture of vehicle images constitute an efficient traffic monitoring system for its features. However, there is still a challenge in terms of manufacturing and installation costs, since well-designed equipment and materials are required in terms of performance and functionality to face different network conditions [19,34]. This is a very important challenge because the large number of papers published by researchers in recent years (see [35] or [4] for a good review), stated that in order to achieve good traffic flow estimation results, a large number of sensors has to be installed. Even when trying to minimize this number the model developed in [36] proposed to install 200 ID-sensors to obtain the full observability of a real size city with 2526 links. Depending on the case of study, this can be an unaffordable cost. In addition, the sensor location models have to be designed to take into account the particular characteristics of installation of the type of sensor to be used.

Therefore, all the context exposed in this section motivates the preparation of this original paper, whose main contributions are as follows:

- A novel architecture to deploy low-cost sensor networks able to automatically recognize plate numbers, which can be temporarily installed on city streets as an alternative to rubber hoses commonly used in the elaboration of urban mobility plans.
- A design of a low-cost, low-energy sensor composed of a number of hardware components that provides flexibility to conduct urban mobility experiments and minimize the impact on maintenance, installation, and operability.
- A methodology to locate the sensors able to establish the best set of links of a network given both the study objectives and of the sensor needs of installation. This model integrates the estimation of traffic flows from the data obtained by the proposed sensors and also establishes the best set of links to locate them taking into account the special characteristics of its installation. Furthermore, using the proposed methodology, we have proved that the expected quality of the traffic flow estimation results are very similar if the sensor can be located in any link compared with avoiding links with certain problems to install the sensor.

The rest of the paper is organized as follows: in Section 2, the proposed low-cost sensor and its associated system for traffic networks analysis are deeply described. In Section 3 the proposed system is applied in a pilot project in Ciudad Real (Spain). Finally, some conclusions are provided in Section 4.

2. The Proposed Low-Cost ANPR System for Traffic Networks Analysis

This section deals with the description of the proposed system which is composed of three elements: (1) the proposed architecture to deploy the sensor networks, (2) the devised low-cost sensor prototype, and (3) the adopted method to decide the best set of links where the sensors have to be installed.

2.1. Architecture to Deploy Low-Cost Sensor Networks

2.1.1. General Overview

Figure 2 shows the multi-layer architecture designed to deploy low-cost sensor networks for automatic license plate detection. The use of a multi-layer approach ensures the scalability of the architecture, as it is possible to carry out modifications in each of the layers without affecting the rest. In particular, the architecture is composed of three layers:

1. The perceptual layer, which integrates the self-contained sensors responsible for image capture. Each of these sensors integrates a low-power processing device and a set of low-cost devices that carry out the image capture. The used camera enables different configurations depending on the characteristics of the urban environment in which the traffic analysis is conducted.
2. The smart management layer, which provides the necessary functionality for the definition and execution of traffic analysis experiments. This layer integrates the functional modules responsible for the configuration of experiments, the automatic detection of license plates, from the images provided by the sensors of the perceptual layer, and the permanent storage of information in the system database.
3. The online monitoring layer, which allows the visualization, through a web browser, of the evolution of an experiment as it is carried out. Thanks to this layer, it is possible to query the state of the different sensors of the perceptual layer, through interactions via the smart management layer.

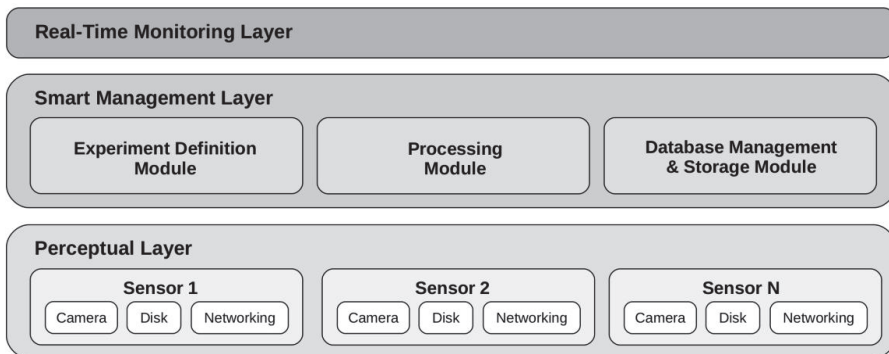


Figure 2. Multi-layer architecture for the deployment of low-cost sensor networks for automatic license plate recognition.

2.1.2. Perceptual Layer

The perceptual layer is the lowest-level layer of the proposed architecture. It contains the set of basic low-cost processing sensors that will be deployed in the physical environment to perform the image capture. In this layer, these sensors are not aware of the existence of the rest of the sensors. In other words, each sensor is independent of the others and its responsibility is limited to taking pictures at a certain physical point, sending them to the upper-level layer, and, periodically, notifying

they are working properly. Since the sensor design represents a major component of the proposed architecture, a detailed description of its main characteristics is done in Section 2.2.

2.1.3. Smart Management Layer

The smart management layer provides the functionality needed to (i) facilitate the deployment of low-cost sensor networks and the execution of experiments, (ii) process the images captured by the perceptual layer sensors, performing automatic license plate detection, and (iii) persistently store all the information associated with an experiment for further forensic analysis.

This layer of the architecture follows a Platform-as-a-Service (PaaS) model, i.e., using the infrastructure deployed in the cloud, which provides the computational needs of the traffic analysis system. This approach makes it possible to offer a scalable solution that responds to the demands of the automatic license plate detection system, avoiding the complexity that would be introduced by deploying our own servers to provide functional support.

Particularly, “Google App Engine” [37] has been used as a functional support for the system’s server, using the Python language to develop the different components of the system and the Flask web application framework to handle web requests. The information retrieval with respect to the sensors of the perceptual layer is materialized through web requests, so that these can ask for the initial configuration of a sensor, or send information, as the so-called “control packages”, as the state of a traffic analysis experiment evolves.

In this context, the control package concept stands out as the basic unit of information to be handled by the smart management layer. The control package is composed of the following fields:

- Client ID. Unique ID of the sensor that sends the packet within the sensor network.
- Timestamp. Temporal mark associated with the time when the sensor captured the image.
- Latency. Latency regarding the previously sent control packet. Its value is 0.0 for the first control packet.
- Plates. List of candidate plate numbers, together with their respective confidence values, detected in the image captured by the sensor.
- Image. Binary serialization of the image captured by the sensor.

There are three different modules in this layer, which are detailed next:

1. Experiment definition module: This module is responsible for managing high-level information linked to a traffic analysis experiment. This information includes the start/end times of the experiment and the configuration of the parameters that guide the operation of the perceptual layer sensors. This configuration is retrieved by each of these sensors through a web request when they start their activity, so that it is possible to adjust it without modifying the status of the sensors each time it is necessary.
2. Processing module: This module provides the functionality needed to effectively perform automatic license plate detection. Thus, the input of this module is a set of images, in which vehicles can potentially appear, and the output is the set of detected plates, together with the degree of confidence associated with those detections. In the current version of the system, the commercial, web version OpenALPR library is used [26]. This module is responsible for attending the image analysis requests made by the sensors of the perceptual layer. Both the images themselves and the license plate detections associated with them are reported to the database management and storage module.
3. Database management and storage module: This module allows the permanent storage of all the information associated with a traffic analysis and automatic license plate detection experiment. At a functional level, this module offers a forensic analysis service of all the information generated as a result of the execution of an experiment.

It is important to note that the processing module offers two modes of operation: (i) online and (ii) offline. In the online mode, the processing module carries out an online analysis of the images obtained from the perceptual layer, parallelizing the requests received by them to provide results in an adequate time. In contrast, the offline mode of operation is designed to analyze large sets of images associated with the past execution of a traffic analysis experiment.

2.1.4. Online Monitoring Layer

The general objective of this layer is to facilitate the monitoring, in real time, of the evolution of a traffic analysis experiment. In order to facilitate the use of the system and avoid the installation of software by the user, the interaction through this layer is made by means of a web browser. From a high level point of view, the online monitoring layer offers the following functionality:

- Overview of the system status: Through a grid view, the user can visualize a subset of sensors in real time. This view is designed to provide a high level visual perspective of the sensors deployed in a traffic analysis experiment. It is possible to configure the number of components of the grid.
- Analysis of the state of a sensor: This view makes it possible to know the status, in real time, of one of the previously deployed sensors (see Figure 3). In addition to visualizing the last image captured by the sensor, it is possible to obtain global statistics of the obtained data and the generated information if an online analysis is performed.

In both cases, the information represented in this layer, through a web browser, is obtained by making queries to the layer of the immediately lower level, that is, the smart management layer. The latter, in turn, will obtain the information from the perceptual layer, where the sensors deployed in the physical scenario are located.



Figure 3. Visualization obtained from one of the sensors. (To protect personal data, the first three digits of the license plate have been blurred.)

2.1.5. Systematic Requirements

This subsection presents a well-defined set of systematic requirements provided by the devised architecture, considering the practical deployment of low-cost sensor networks for ANPR. These requirements are as follows:

- Scalability, defined as the architectural capacity and mechanisms provided to integrate new components.
- Availability, defined as the system robustness, the detection of failures, and the consequences generated as a result of these failures.
- Evolvability, defined as the system response when making software or hardware modifications.
- Integration, defined as the capacity of the architecture to integrate new devices.
- Security, defined as the ability to provide mechanisms devised to deal with inadequate or unauthorized uses of the deployed sensor networks.
- Manageability, defined as the capacity to interact between the personnel responsible to conduct the experiments and the software system.

Regarding scalability, the architecture proposed in this work provides support (i) when new low-cost sensors need to be integrated and (ii) when new physical locations need to be monitored. The integration of new sensors is carried out in the perceptual layer. Thus, this systematic requirement is guaranteed thanks to the existing independence between sensors. As mentioned before, each sensor is responsible for a single physical point. Similarly, when a new physical location needs to be added, then it is only necessary to deploy a new sensor which, in turn, will send information to the upper-level layer and will notify whether it is working properly. This is why integration is also guaranteed in terms of adding new devices when they are required. In other words, this requirement is strongly related to scalability of the devised architecture.

The systematic requirement named availability has been achieved thanks to the adopted cloud-based approach, since it is easy to incorporate multiple layers of license plate analysis so that processing errors are identified. Although the currently deployed system only uses OpenALPR, the architecture easily allows the incorporation of other license plate identification platforms that minimize potential errors. On the other hand, all processing sensors run the same software on the same hardware. Replacing a sensor implies changing its identifier and the server address that are specified in the configuration file stored in the memory stick. In other words, replacing a faulty sensor is a simple and straightforward task. This decision is related to the systematic requirement evolvability.

With respect to security, multiple methods have been considered to protect the information exchanged between the different components of the architecture, ensuring its integrity. Particularly, the extension hypertext transfer protocol secure (HTTPS) has been used to guarantee a secure communication so that the information is encrypted using secure sockets layer (SSL). Finally, regarding manageability, the devised architecture aims at facilitating the deployment process of sensor networks. In fact, there is a component, named experiment module definition, which has been specifically designed to address this systematic requirement. As previously stated in Section 2.1.3, it is possible to set up experiments and adjust the configuration of the sensors in a centralized way, without having to individually modify the internal parameters of every single sensor.

2.2. Low Cost Sensor Prototype Description

2.2.1. Production Cost

From a hardware point of view, each low-cost sensor (€62.27) is composed of the following components (see Figure 4):

- Raspberry Pi Zero W: €11.97
- Raspberry Pi Camera Module V2.1: €23.63
- Power bank PowerAdd Slim2 5000 mAh: €8.29
- 3D plastic box (PLA): €1.18
- MicroSD card U1 16 GB: €6.49
- Memory stick 128 MB: €1.97
- Tripod: €10.74

Raspberry Pi is a low-cost single board computer running open source software. The multiple versions of the board employ a Broadcom processor (ARM architecture) and a specific camera connector. Thanks to the use of this hardware, the versions of the Raspberry Pi OS (formerly called “Raspbian”), derived from the GNU/Linux distribution Debian, can be used. Thus, the development in a number of general-purpose programming languages is possible.

For the development of the sensor previously introduced, the version of the board called Pi Zero W has been used, which incorporates the Broadcom BCM2835 microprocessor. This has a single-core processor running at 1 GHz, 512 MB of RAM, a VideoCore IV graphics card, and a MicroSD card as a storage device. Based on the Pi Zero model, this version offers Wi-Fi connectivity, which allows online monitoring. In the conducted tests, the connectivity with the cloud has been done by using 3G/4G connection sensors, using the existing institutional Wi-Fi network of the University of Castilla-La Mancha whenever possible.

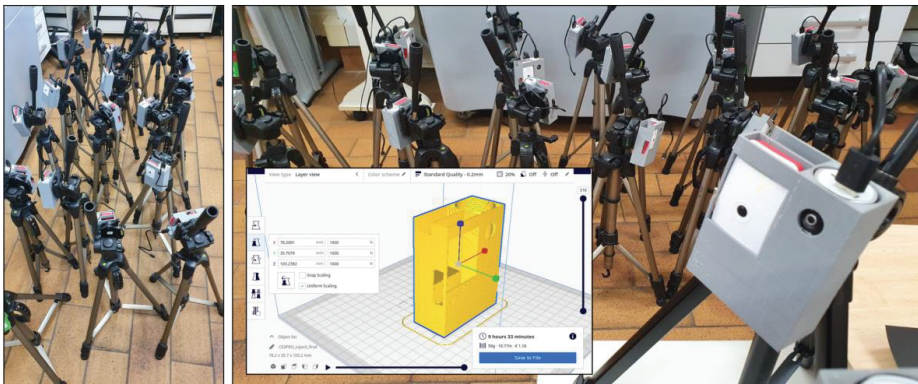


Figure 4. Prototype of the designed sensor.

The 8 megapixel Raspberry Pi Camera V2.1 features Sony’s IMX219RQ image sensor with high sensitivity to harsh outdoor lighting conditions, with fixed pattern noise and smear reduction. The connection is made using the camera’s serial interface port directly to the CSI-2 bus via a 15-pin flat cable. The camera automatically performs black level, white balance, and band filter calibrations, as well as automatic luminance detection (for changing conditions) of 50 Hz in hardware. In the configuration of each sensor, the resolution with which each photograph is taken can be specified, up to the maximum of 3280×2464 pixels.

The used Lithium battery holds a capacity of 5000 mAh, with an output of 5 V/2.1 A and a very small size and weight ($100 \times 33 \times 31$ mm, 195 g).

The installed operating system is based on Debian Buster, with kernel version 4.19. The installation image has a size of 432 MB which, once installed on the system partition, uses 1.7 GB. The current version of the prototype uses a UHS Speed Class 1(U1) microSD card, with a write speed of 10 MB/s required to record high-definition pictures in short intervals. Each 8 MP photograph may require around 4 MB in jpg format if stored at full resolution (depending on the scene complexity and lighting conditions).

In the conducted tests, each sensor made the captures with a resolution of 1024×720 pixels. Each image occupied an average of 412 KB, size that was reduced to 151 KB after the optimization process with capture sub-regions. The capture frequency was established to 1 image per second. This requires a disk storage of 1.4 GB for every hour of capture without optimization. Thus, with more than 14 GB available on the SD card for data, it is possible to store more than 8 h of images without optimization, and more than 24 h by defining capture sub-regions.

The 128 MB Flash Drive is used to store the processing sensor configuration parameters, such as the unique identifier of the processing sensor, the address of the web server associated with the intelligent experiment management layer, and the network configuration.

For the integration of all hardware components of the system, a basic prototype has been made using 3D printing, with a size of $103 \times 78 \times 35$ mm, and a unit cost of 1.18 (59 g of PLA of 1.75 mm).

The cost of the designed sensor is similar to some of the low-cost sensors discussed in [22]. However, the offered functionality can be compared to commercial systems with a significantly greater cost. Plus, the devised architecture enhances the global functionality of the sensor networks deployed from the architecture, and this is a major improvement regarding existing work in the literature.

2.2.2. Energy Cost

The energy cost of the system depends mainly on the use of the processor. In the deployed system, the most expensive computational stage is done in the cloud, so three working states can be defined in the sensors: the “idle” mode, in which the sensor is waiting for work orders, the “capture” mode, in which the sensor accesses the camera and saves the image in the local storage, and the “networking” mode, which optimizes the image with the defined sub-regions and sends them to the smart management layer.

Table 1 summarizes the power consumption between different versions of Raspberry Pi (all five versions). The ZeroW version was chosen because it provides wireless connectivity (not available on Zero), and because of the very low power consumption it requires (0.6 Wh in idle mode, 1 Wh in capture mode, and 1.19 Wh in networking mode). In this way, a small 10 W solar panel could be enough to provide the energy required by the sensor.

Table 1. Power consumption comparison in mAh of different versions of Raspberry Pi.

| | Zero | ZeroW | A | B | Pi2B | Pi3B |
|------------|------|-------|-----|-----|------|------|
| Idle | 100 | 120 | 140 | 360 | 230 | 230 |
| Capture | 140 | 200 | 260 | 420 | 290 | 290 |
| Networking | - | 230 | 320 | 480 | 350 | 350 |

2.2.3. Maintenance, Installation, and Operability

The use of a general purpose processor, such as the Broadcom BCM2835, facilitates rapid prototyping, as well as the integration of existing software modules. In particular, the integration of the functionality offered to the smart management layer is done in a straightforward way thanks to this approach.

On the other hand, the impact of maintenance costs and the addition of new functionality is minimized by using a cloud-based approach where each sensor is configured through specific parameters. A unique identifier and server address are specified for each sensor. From the server, the sensor receives a JSON message with the parameters to be used in each analysis experiment. By using this configuration package per sensor, it is possible to adjust the specific capture configuration of each sensor in the network, based on its position, weather conditions, or lighting level at each time of day. For example, a sensor that may be better positioned to identify license plates will be able to take lower resolution captures (saving processing costs) than a sensor that is located further away from the traffic. Even the same sensor may need to make higher resolution captures in adverse weather situations, such as rain or fog.

The JSON message has the same format:

```
{
  "begTime": "2020-06-10T09:00:00",
  "endTime": "2020-06-10T11:00:00",
  "resolution": "1024x720",
```

```

"mode": "manual",
"exposure_time": 1000,
"freq_capture": 1000,
"iso": 320,
"rectangle_p1": [
    280,
    262
],
"rectangle_p2": [
    1024,
    574
]
}

```

The fields `begTime` and `endTime` indicate the date and time of the start and the end of the capture session. Resolution indicates the capture resolution of the sensor with values supported by the hardware up to a maximum of 3280×2464 pixels. If the field `mode` is set to `manual`, it is possible to indicate the shutter speed or exposure time, which defines the amount of light that enters the camera sensor. The parameter `exposure_time` defines the fraction of a second (in the form $1/\text{exposure seconds}$) that the light is allowed to pass through. The field `freq_capture` indicates the number of milliseconds that will pass between each capture. The field `iso` defines the sensitivity of the sensor to light (low values for captures with good light level). Finally, the fields that begin with the keyword `rectangle` allow us to define capture sub-regions within an image. The upper left and lower right corners define the valid capture rectangle within the image. The rest of the pixels are removed from the image, facilitating the transmission of the image through the network and avoiding storage and processing costs in regions where plate numbers will never appear (see Figure 5).



Figure 5. Example of definition of clipping parameters in capture sub-regions in three sensors of the deployed system, with comparative analysis of storage size for each frame. (To protect personal data, the first three digits of the license plate have been blurred).

The use of parameters that are used to define sub-regions in the captured images, their size can be drastically reduced. Any 3G connection is more than enough to cover the bandwidth requirements of

each processing sensor, without any loss of image quality. Even under more adverse transmission conditions (such as Enhanced Data rates for GSM Evolution (EDGE) or General Packet Radio Service (GPRS) coverage with maximum speeds between 114 and 384 Kbps), the frame could be stored using a higher level of JPG compression without significant loss of image quality (up to a level of 65 would be acceptable), and therefore without putting at risk the identification of the license plate (see Figure 6).



Figure 6. Different compression levels of the JPG standard. With values below 60, with significant loss of high frequency information, the image quality significantly compromises the success rate of the license plate detection algorithms. (To protect personal data, the first three digits of the license plate have been blurred).

2.2.4. Information Processing

The proposed architecture, and particularly the smart management layer that was previously discussed, improves the processing costs, offering results that can be in real time or with programmed offline execution. In this way, the use of the platform as a whole can even be shared between different sets of sensors, avoiding the unnecessary complexity of local management at the level of each sensor or group of sensors.

From the point of view of information processing, it is possible to minimize the information traffic between the image analysis system (in the cloud) and the processing sensors. As a way of example, Figure 7 shows how the sensors can make fewer requests by encoding multiple captures into one single image.



Figure 7. Detection of vehicles in a 3 × 3 image matrix, which allows, by means of a single sending to the cloud, to summarize 9 s of traffic analysis of a given sensor. The system detects both the number plate and certain characteristics of the vehicle, assigning a confidence value to each detection. (To protect personal data, the first three digits of the license plate have been blurred).

2.3. Methodology to Locate ANPR Sensors in a Traffic Network

Having described the sensors to be located and its operating system, the next step is to determine their best locations on the network. To do this, given (1) a reference demand and traffic flow conditions; (2) a traffic network, defined by a graph (N, A) , where N is the set of nodes and A is the set of links; and (3) the budget of the project (i.e., a number of available sensors), the next aim is to obtain the locations that allow obtaining the best possible traffic flow estimation. Depending on the number of sensors to be located, we can achieve total or partial observability of the network according to the flow conditions and the number of routes modelled on it. The suitable locations for these sensors are determined from the use of two algorithms that integrate the previous three elements. In this section, these two algorithms are described.

2.3.1. Algorithm 1: Traffic Network Modelling

The method used to build an appropriate network model, given a graph (N, A) , for traffic analysis using plate-recognition based data is the one proposed in [13]. We assume that every node of the network can be the origin and the destination of trips, and therefore the classic zone-based D–D matrix has to be transformed into a node-based O–D matrix used as reference. This matrix is assigned to the network using a route enumeration model. Then, a route simplification algorithm is proposed based on transferring to adjacent nodes the generated or attracted (reference) demand of those nodes that generate or attract fewer trips than a given threshold. Figure 8 shows the operation of this first algorithm that involves the modeling of the network, and whose steps are described below.

- **INPUT:** A traffic network (N, A) and its link parameters (links cost, links capacities, etc.); an out-of-date O–D matrix defined by traffic zones; capacities of links to attract and generate trips; the k parameter of the MNL assignment model; and the threshold flow (F_{thres}) to simplify the node-based O–D matrix and its corresponding routes.
- **OUTPUT:** A set Q of representative routes of the network and its corresponding route flow f_q^0 , and a set of “real” data necessary to check the efficiency of the estimation.

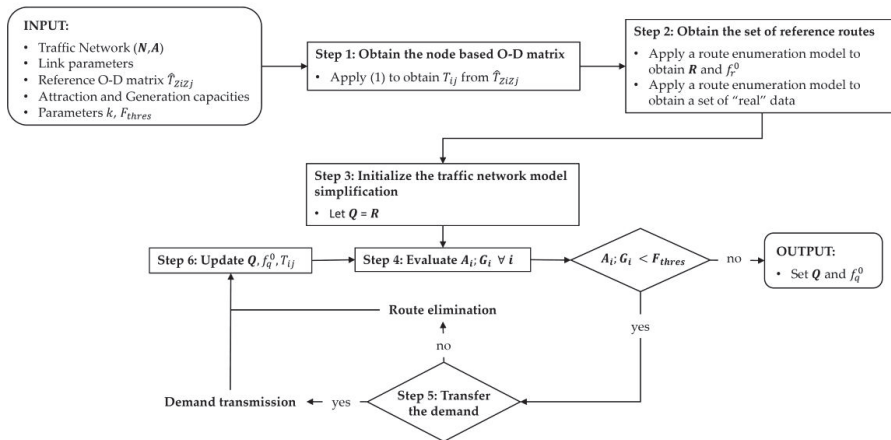


Figure 8. Flowchart of the algorithm defined for the traffic network modelling.

STEP1: Obtain the node-based O–D matrix: Given an O–D matrix by traffic zones in the network, and from some data on the attraction and trip generation capacities of the links that form it (see [13] for more details), it is possible to obtain an extended O–D matrix by nodes, defined as follows:

$$T_{ij} = \hat{T}_{ZiZj} P A_i P G_j \quad (1)$$

where T_{ij} is the number of trips from node i to node j ; \hat{T}_{ZiZj} is the number of trips from zone of node i to the zone of node j ; PA_i is the proportion of attracted trips at node i ; and PG_j is the proportion of generated trips at node j which depends on its capacity to attract or to generate trips.

- STEP2: Obtain the set R of reference routes: After defining the O–D matrix, an enumeration model, based on Yen’s k -shortest path algorithm [38], is used to define those k -shortest routes between nodes, which are then assigned to the network through a MNL Stochastic User assignment model. This model makes it possible to build an “exhaustive reference set of routes” R between nodes, with its respective route flows f_r^0 , which will be operated by the algorithm, and whose size will vary according to the value adopted by the parameter k . Along with these reference data, other data considered as “real” will also be defined that will serve to check the effectiveness of the model in the flow estimation results obtained from the information collected by the sensors.
- STEP3: Initialize the traffic network model simplification: The intention of this step is to adapt the modelled traffic network as close to the actual network as possible, simplifying those routes by O–D pairs whose attraction/generation trip flow is below a given threshold flow value. To do this, we initialize the set Q of modelled routes to the set R of reference routes.
- STEP4: Evaluate the trip generations or attractions of the nodes: The algorithm evaluates the trips generated and/or attracted of each node of the network and compares them with the threshold flow value F_{thres} . If there is any node that holds this condition, go to Step 5, otherwise the algorithm ends and a simplified set of routes Q will be obtained, whose size will be a function of the value of the F_{thres} flow considered.
- STEP5: Transfer the demand: The node that meets with the condition in Step 4, would lose its generated/attracted demand, which would also imply that no route could begin or end from that node. Therefore, it will be necessary to transmit these flow routes to another node close enough (which could receive or emit demand) with an implicit route, whenever possible. If the demand transfer could not be carried out, the evaluated demand is lost and all the involved routes as well.
- STEP6: Update the set of routes: Q . The set Q and its associated flows f_q^0 have to be updated with the deleted or updated routes. The O–D Matrix T_{ij} must also be updated. Go to Step 4.

2.3.2. Algorithm 2: The ANPR Sensor Location Model

After defining the traffic network, i.e., the set of routes and its associated reference flows, both are introduced in the location model so that from these, and with the particularities of the sensor to be used, this model allows us to obtain a set of links, SL , to locate a certain number of sensors to collect data able to obtain the best possible estimation of the remaining flows of the network. This can be a difficult combinatorial problem to solve, especially when it is required to locate sensors in large networks with a great number of existing routes (this justifies the use of the set of routes Q instead of set R). Next, we propose an iterative problem-solving process to find the best possible solution given a series of restrictions. Figure 9 shows the operation of this second algorithm, and whose steps are described below.

- **INPUT:** A traffic network (N,A) ; Sets of routes R and Q , with their associated flows; the budget to install sensors B ; an optional set of non-candidate scanned links NSL ; and maximum number of iterations to be performed $iter^{max}$.
- **OUTPUT:** The set of scanned links SL^{best} and the evolution of the $RMARE$ value according to the performed iterations.

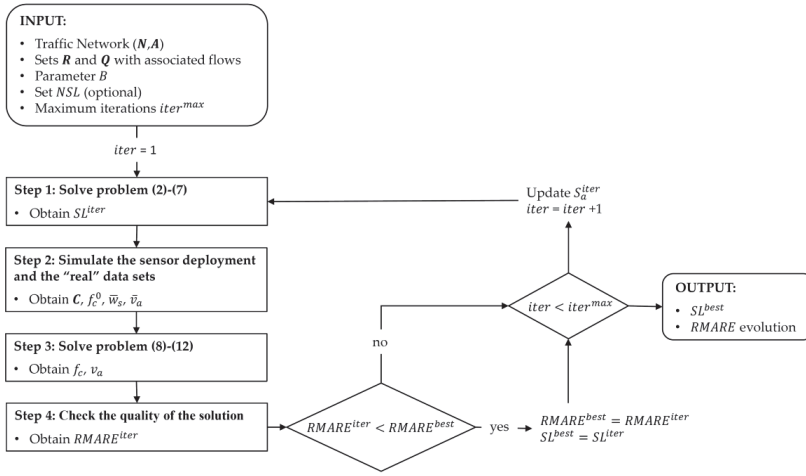


Figure 9. Flowchart of the algorithm defined for the ANPR sensor location model.

STEP1: Solve the optimization problem: The following optimization problem has to be solved:

$$\max_{z_a, y_q} M = \sum_{q \in Q} f_q^0 y_q \quad (2)$$

subjected to

$$\sum_{a \in A} P_a z_a \leq B \quad \forall a \in A \quad (3)$$

$$\sum_{a \in A} \delta_a^q z_a \geq y_q \quad \forall q \in Q \quad (4)$$

$$\sum_{a \in A} z_a \geq y_q \quad \forall (q, q1) \in Q^2 \mid q > q1; \sum_{a \in A} \delta_a^q \delta_a^{q1} > 0 \quad (5)$$

$$z_a = 0 \quad \forall a \in NSL \quad (6)$$

$$\sum_{a \in A} S_a^{iter} z_a \leq \sum_{a \in A} S_a^{iter} \quad \forall iter \in I \quad (7)$$

The objective function (2) maximizes the distinguished route flow in terms of f_q^0 ; y_q is a binary variable equal to 1 if a route can be distinguished from others and 0 otherwise. Constraint (3) satisfies the budget requirement, where z_a is a binary variable that equals 1 if link a is scanned and 0 otherwise. This constraint guarantees that we will have a number of scanned links with a cost P_a for link a that does not exceed the established limited budget B . Constraint (4) ensures that any distinguished route contains at least one scanned link. This constraint is indicated by the parameter δ_a^q , which is the element of the incidence matrix. Constraint (5) is related to the previous constraint since it indicates the exclusivity of routes: a route q must be distinguished from the other routes in at least one scanned link a . If $\delta_a^q + \delta_a^{q1} = 1$, this means that the scanned link a only belongs to route q or route $q1$. If $\sum z_a \geq y_q$ and $y_q = 1$, then at least one scanned link has this property; on the other hand, if $y_q = 0$, then the constraint always holds. Constraint (6) is an optional constraint that allows a link to not be scanned if it belongs to a set of links not suitable for scanning NSL . This restriction will make the binary variable z_a equal to 0, and therefore a sensor cannot be located on it. The intention of defining this constraint will be discussed with more detail in the next section. Finally, since

this model is part of an iteration process (see [13] for more details), an additional constraint (7) is proposed, which allows us to obtain different solutions of SL sets for each iteration performed through the definition of S_a^{iter} , which is a matrix that grows with the number of iterations I , in which each row reflects the set SL resulting from each iteration carried out up to then by the model. Therefore, if an element of S_a^{iter} is 1, means that link a was proposed to be scanned in the solution provided on iteration $iter$ and 0 otherwise. Each iteration keeps the previous solutions and does not permit the process to repeat a solution in future iterations. That is, each iteration carried out by the algorithm is forced to search for a different solution SL^{iter} with the same objective function (2).

- STEP2: Simulate the sensor deployment and the “real” data sets: After obtaining the set SL^{iter} , the numerical simulation of it on the traffic network is carried out with the flow conditions given by an assumed “real” condition. One of the main features introduced in that algorithm is the possibility of working with a set of routes not fixed. Until now, the sensor location and flow estimation models have been formulated considering a set of existing fixed and non-changing routes. In the proposed model, each set SL may allow to obtain different observed set of combinations of scanned links ($OSCSL$) used by the vehicles (i.e., sets of links where vehicles have been registered), denoted by s . Since the modelled network and routes are not always the same as in reality, not all sets in $OSCSL$ are compatibles with set of routes Q and hence new routes have to be added conforming a new global set C that encompasses the routes in Q and the new ones, with their associated flows f_c^0 . To define these new sets from new routes discovered from this simulation, the algorithm looks for and assimilates their compatibility with those routes from set R that were eliminated in the simplification step of Algorithm 1 (see [13] for more detail). With this step, each set s of observed combinations of scanned links will provide the observed flow values \bar{w}_s as the input data for the estimation model. In addition, apart from allowing to quantify the flow in routes from the scanned sets of links, these sensors behave as traffic counters in the link where they are installed, making it easier to quantify the flow in the link as well \bar{v}_a .
- STEP3: Obtain the remaining traffic flows: In this step, a traffic flow estimation of the remaining flows is performed where route (f_c) and links (v_a) flows are obtained from reference flows (f_c^0) and the observed flows (\bar{w}_s and \bar{v}_a). We propose to use a Generalized Least Squares (GLS) optimization problem [14,15], as follows:

$$\min_{f_c; v_a} Z = \sum_{c \in C} U_c^{-1} \left(\frac{f_c - f_c^0}{f_c^0} \right)^2 + \sum_{a \in SL} Y_a^{-1} \left(\frac{v_a - \bar{v}_a}{\bar{v}_a} \right)^2 \tag{8}$$

subjected to

$$\bar{w}_s = \sum_r \beta_s^c f_c; \quad \forall s \in S \tag{9}$$

$$v_a = \sum_r \delta_a^c f_c; \quad \forall a \in A \tag{10}$$

$$f_c \geq 0; \quad \forall c \in C \tag{11}$$

$$v_a \geq 0; \quad \forall a \in A \tag{12}$$

where U_c^{-1} and Y_a^{-1} are the inverses of the variance–covariance matrices corresponding to the flow in route C and the observed flow in link a ; \bar{w}_s is the observed flow in each set $OSCSL$; f_c is the estimated flow of routes in set C ; β_s^c and δ_a^c are the corresponding incidence matrices of relationship between observed link sets s and links a with routes.

- STEP4: Check the quality of the solution. Once the flow estimation problem has been resolved, the quality of the solution in absolute terms, can be quantified as follows:

$$RMARE = \frac{1}{n} \sum_{a \in A} \frac{|v_a - v_a^{real}|}{v_a^{real}} \quad (13)$$

where $RMARE$ is the root mean absolute value relative error; n is the number of links in the network; and v_a and v_a^{real} are the estimated flow and (assumed) real flow for link a . Such error is calculated over the link flows since the number of them remain constant regardless of the network simplification and the SL set used. Each value of $RMARE$ indicates the quality of the estimation by using the set SL for the traffic network. As said above, due to the complexity of the problem, it has not a unique solution so we propose to evaluate a great amount of combined solutions in an iterative process. This iterative process, which is shown in Figure 9, is carried out since Step 1 a number of iterations equal to the maximum considered $iter^{max}$. For the solution found in the first iteration, the value of $RMARE$ will be considered as the best, but in the following iteration, the algorithm could find another solution with lower value of and it will be considered as best. All the solution found and tested in each iteration are stored in S_a^{iter} matrix, which grows in size during the performance. Finally, the best solution or set SL^{best} , for the established conditions, will be the one provided with the lower $RMARE$ value.

3. The Application of the Proposed System in a Pilot Project

In this section, the proposed low-cost system for traffic network analysis is applied in a pilot project in a real network to demonstrate its viability and also to test the influence that some inputs of the Algorithm 1 (i.e., the network modelling) have in the results of the Algorithm 2 (i.e., the expected traffic flow estimation quality).

3.1. Description of the Project and Particularities about the Position of Sensors in the Streets

The network chosen to develop the pilot project was the traffic network of the University Campus of Ciudad Real (Spain), delimited in Figure 10, consisting of 75 nodes and 175 links. To consider the influence that the other districts have on this network, links connected to the contour were also modelled. With this, the set of links, its capacity and cost characteristics are established (these characteristics are available by requesting them to the corresponding author). The O–D matrix $\hat{T}_{Z_i Z_j}$ was first defined, where each zone Z contains a certain set of nodes (see Table 2) resulting in a total of 15 zones as shown in Figure 10, while the reference O–D matrix by zones is shown in Table 3.

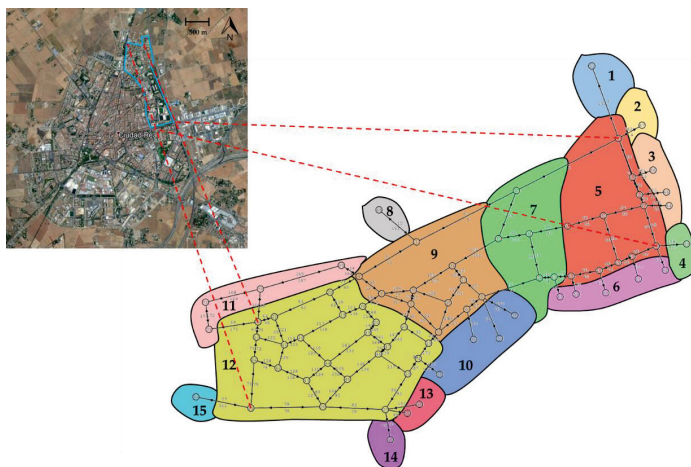


Figure 10. Traffic network to be modelled for analysis: Plan view of the urban area of Ciudad Real and the delimited Campus area to be modelled; Ciudad Real Campus network and its division in 15 traffic zones.

Table 2. List of nodes on the traffic network and their associated zones.

| Node | Zone | Node | Zone | Node | Zone | Node | Zone | Node | Zone | Node | Zone |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 5 | 14 | 5 | 27 | 12 | 40 | 12 | 53 | 14 | 66 | 3 |
| 2 | 7 | 15 | 5 | 28 | 12 | 41 | 12 | 54 | 13 | 67 | 2 |
| 3 | 9 | 16 | 5 | 29 | 9 | 42 | 12 | 55 | 13 | 68 | 8 |
| 4 | 9 | 17 | 5 | 30 | 5 | 43 | 12 | 56 | 10 | 69 | 11 |
| 5 | 9 | 18 | 5 | 31 | 5 | 44 | 12 | 57 | 10 | 70 | 11 |
| 6 | 9 | 19 | 5 | 32 | 7 | 45 | 12 | 58 | 6 | 71 | 11 |
| 7 | 9 | 20 | 12 | 33 | 7 | 46 | 12 | 59 | 6 | 72 | 11 |
| 8 | 9 | 21 | 12 | 34 | 9 | 47 | 12 | 60 | 6 | 73 | 10 |
| 9 | 9 | 22 | 12 | 35 | 9 | 48 | 12 | 61 | 6 | 74 | 12 |
| 10 | 7 | 23 | 12 | 36 | 9 | 49 | 9 | 62 | 6 | 75 | 10 |
| 11 | 5 | 24 | 12 | 37 | 9 | 50 | 12 | 63 | 4 | | |
| 12 | 5 | 25 | 12 | 38 | 12 | 51 | 1 | 64 | 3 | | |
| 13 | 5 | 26 | 12 | 39 | 12 | 52 | 15 | 65 | 3 | | |

Table 3. O–D trip matrix per defined zones.

| Zone | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | Total |
|-------|-----|-----|----|-----|-----|------|-----|-----|-----|-----|-----|-----|-----|-----|------|-------|
| 1 | - | 269 | 0 | 31 | 21 | 115 | 17 | 0 | 68 | 0 | 10 | 23 | 60 | 217 | 137 | 968 |
| 2 | 125 | - | 0 | 83 | 13 | 97 | 0 | 0 | 0 | 25 | 0 | 45 | 23 | 36 | 8 | 455 |
| 3 | 0 | 0 | - | 170 | 0 | 64 | 0 | 17 | 0 | 35 | 0 | 0 | 0 | 28 | 0 | 314 |
| 4 | 124 | 160 | 0 | - | 156 | 83 | 118 | 0 | 0 | 0 | 46 | 0 | 166 | 224 | 0 | 1077 |
| 5 | 70 | 89 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 118 | 0 | 40 | 5 | 322 |
| 6 | 43 | 118 | 0 | 70 | 20 | - | 46 | 32 | 0 | 0 | 2 | 0 | 0 | 48 | 28 | 407 |
| 7 | 50 | 12 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 9 | 0 | 17 | 2 | 90 |
| 8 | 3 | 4 | 0 | 0 | 0 | 31 | 0 | - | 0 | 47 | 12 | 34 | 25 | 16 | 0 | 172 |
| 9 | 15 | 26 | 48 | 62 | 0 | 55 | 0 | 0 | - | 105 | 39 | 73 | 64 | 13 | 22 | 522 |
| 10 | 30 | 12 | 0 | 55 | 0 | 37 | 0 | 95 | 375 | - | 0 | 0 | 57 | 21 | 0 | 682 |
| 11 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 26 | 0 | 37 | 3 | 82 | |
| 12 | 245 | 29 | 0 | 0 | 0 | 154 | 0 | 44 | 0 | 0 | 0 | - | 0 | 0 | 28 | 500 |
| 13 | 105 | 39 | 0 | 181 | 0 | 106 | 0 | 0 | 0 | 0 | 0 | - | 63 | 0 | 494 | |
| 14 | 0 | 83 | 0 | 271 | 98 | 461 | 75 | 37 | 26 | 83 | 50 | 0 | - | 0 | 1184 | |
| 15 | 80 | 25 | 0 | 0 | 123 | 144 | 31 | 0 | 52 | 10 | 21 | 0 | 0 | - | 486 | |
| Total | 898 | 874 | 48 | 923 | 431 | 1347 | 287 | 225 | 521 | 305 | 180 | 328 | 395 | 760 | 233 | 7755 |

From a technical point of view, installing each traffic sensor in the streets of a city can be a complex task depending on the configuration and characteristics of the network and the elements that configure it. In the case of links or linear elements, the configuration of their infrastructure and the flow conditions, such as its distribution and intensity along a day, may condition the choice of one or another location, or even consider whether or not a link is a candidate for a sensor to be located. This section deals with the specific problems of installing the sensor in some links.

After a first test of the sensor in the streets of the project, we found a set of links that, due to their physical characteristics, may difficult the sensor to be installed. As shown in Figure 11a in violet, this set is formed by the links that make up the external corridor that connects the ends of the network, which is one of the main arterials of the city. In this type of links (see images 1 and 2 in Figure 11b), the vehicles can reach higher speeds and flow densities, which can make it difficult to capture the data because the license plate is not read correctly due to the occlusion of other vehicles as these links have two lanes per direction. Installing sensors in links where their characteristics make such a task difficult, may involve higher installation and/or operating costs, increasing the possibility that the data that they collect may have errors that may disturb the results of the analysis and the estimation of the remaining flows. The problems that wrong readings of plate license may involve on the flow estimation results have been studied in detail by [20]. Despite these links being very important since the greatest flow of vehicles takes place in them as they are one of the most important arterial corridors of the city. Therefore, the effects of not locating a sensor on them must be investigated.

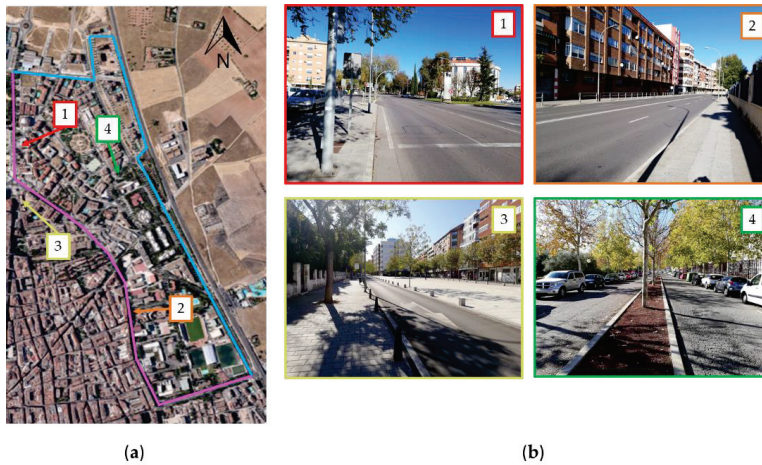


Figure 11. (a) Representation of the corridor, in violet, where the greatest flows and difficulties in installing a sensor take place; (b) Images of several links on the network.

With respect to the rest of links (see images 3 and 4 in Figure 11b), the results obtained from the sensor were very satisfactory resulting all of them suitable and hence being eligible. The flow conditions make feasible the correct identification of the vehicles regardless of its speed and flow density.

To sum up, the sensor location model described in Algorithm 2, has to consider the possibility of avoiding some links which, despite the fact that the greatest flow is concentrated by them, their characteristics make their installation difficult. This may have an impact on the results, since it seeks to obtain the best estimate of flows in the network with the best combination of scanned links. This observation is considered in the sensor location model with the inclusion of constraint (6), which has been described as a restriction that considers that for the arcs belonging to the *NSL* set, their binary variable is null, and therefore they are not suitable for having a sensor installed. Considering this topic can put a risk in obtaining better or worse estimation results, so an analysis is necessary to show that, by avoiding these links, the expected results of the traffic estimation can be similar. Next section below deals with a deeper analysis.

Finally, within the links that are suitable to be scanned, it is important to assess the different locations in them to obtain the correct reading of the license plates (see Figure 12). Here it is necessary to consider the orientation of the sensor with respect to the flow (i.e., recorded from the front or rear of the vehicle); the presence of fixed elements or obstacles present that make it difficult to identify the vehicle; the lighting among others.

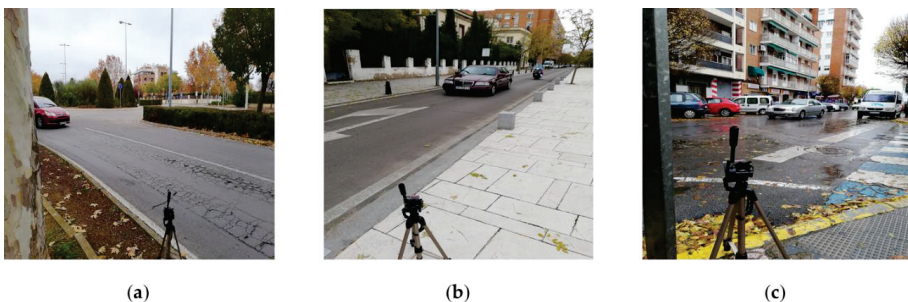


Figure 12. Examples of installed sensor at strategic points: (a) Exit from an intersection and enter a link with two lanes; (b) Link with a single lane with unidirectional flow; (c) Entrance to a link with a single lane.

3.2. Analysis of the Results

To obtain the traffic network model, we have applied the proposed Algorithm 1, where, in addition to the above described input data, the important values of k and F_{thres} need to be defined. Therefore, with object to check the network simplification effects (Steps 4 to 6) on the estimation results (obtained with the Algorithm 2), it was decided to vary the value of the threshold flow F_{thres} , establishing values equal to 10, 15, 20, 25, and 30 trips per hour. Regarding the k parameter used in the enumeration model of Step 2, there are usually certain discrepancies between transportation analysts and engineers about its best value. High values are usually rare to find in the literature due to the high computational cost that it would entail, and also because the existence of more than 3–4 routes per O–D is very unlikely [39]. For this pilot project, it was considered to select reasonable values of k equal to 2 and 3, whose effects on results will be analyzed in the next sections. In Step 1, a 15×15 matrix by zones \hat{T} shown in Table 2 was transformed into a matrix discretized by nodes T with size 75×75 , resulting in a total of 608 O–D pairs. To obtain the set of existing routes assumed to be “real”, the node-based O–D matrix T is affected by a random uniform number (0.9–1.1), and the assignment was done using $k = 4$ with to obtain the respective “real” link flows.

Regarding Algorithm 2, some of its inputs come from the outputs of Algorithm 1 (the traffic network and the routes). Due to the budget restrictions of the project (related to B parameter), an amount of 30 sensors was set to be used in the network. Therefore, for the different models studied, a fixed value B equal to 30 has been considered. Note that in relation to the number of links in the network, this quantity may be insufficient to obtain total observability, but it will be interesting to see to what degree of good estimation it is possible to obtain.

To sum up, in this section, three analysis of results are carried out:

- An analysis of the effect on the estimation of flows is performed when considering different k values for the definition of set R (Step 2 of Algorithm 1). We have considered two values: 2 and 3.
- An analysis over the variation of the value of the threshold flow F_{thres} that is used in the simplification algorithm is done (Steps 4 to 6 of Algorithm 1). Depending on the value of this threshold value, the degree of simplification of the network will be greater or lesser, affecting the number of considered routes in Q .
- An analysis to verify the effect of considering a certain set of links as not suitable to locate the scanning sensor (Equation (6) in Step 1 of Algorithm 2).

3.2.1. Effect of Varying the k Parameter

Vary the k parameter means more or less number of routes in the modelled traffic network are considered, conforming part of the information with which the model must work. The consideration of such a parameter in this project has been through the use of a route enumeration algorithm, selecting values of 2 and 3 for the example presented. For this first analysis, it has been considered to analyze a not very simplified network scenario, considering a F_{thres} equal to 10, i.e., all the nodes that attract or generate less than 10 trips lose its condition of origin and/or destination.

An important aspect that has been studied in this first analysis is related to the consideration of a set of links $NSL \in A$, where the cost and difficulties of installing a scanning sensor are greater than other links in the network. For the shake of brevity, we have decided to undertake a joint study of the k parameter influence and the effects of including some conflicting links in set NSL . A first scenario (Model A) where all the links have the same opportunity of install a sensor, which means that all the links have the same cost P_a equal to 1. In the second scenario (Model B) a certain set of links (those corresponding to corridor shown in Figure 11a), are included in set NSL so a sensor cannot be installed in them.

Figure 13 shows the effects of these considerations on the results of the model. There are four well-differentiated lines in pairs, one assigning a k equal to 2 and another equal to 3. Each jump in the graph means that the location model has found a better set of scanned links SL that improves the

solution in terms of error, and the horizontal sections mean that the model has not been able to find a better solution.

It is observed that considering a higher value of k , the results of the model are better in terms of the error in the estimation of traffic flows. We clearly see how a $k = 3$ obtains quite better results than considering a $k = 2$ due to the existence of a higher number of routes per each O–D pair. In particular, when considering $k = 2$, we are operating with a set R of 2074, as opposed to the 2943 routes considering $k = 3$. To define the set of “real” routes and their associated flows, a value of $k = 4$ has been considered, resulting in a set of 4274 routes in total.

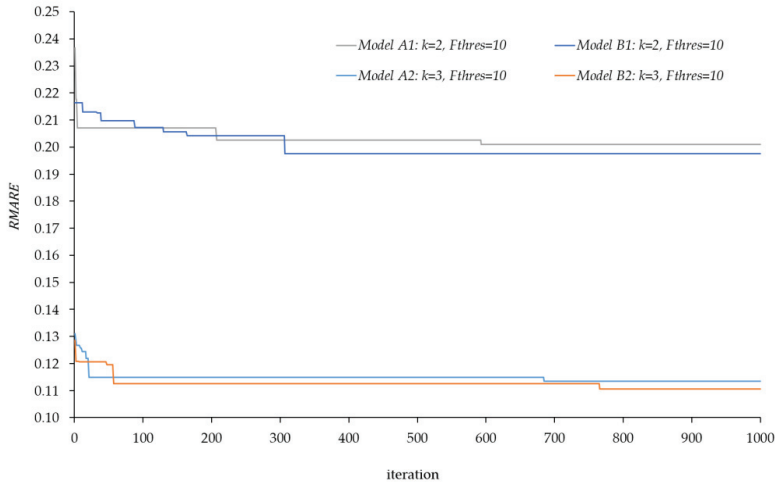


Figure 13. Evolution of $RMARE$ value for the different cases varying the k parameter and the threshold flow.

The most interesting demonstration arises when Model A and Model B are compared. Despite considering a certain amount of links in set NSL , the results of both models reach almost the same $RMARE$ value. We therefore see, in this particular case, how considering or not certain links to install the sensors does not produce a relevant difference in the estimation error to be obtained. In view of this, the following analysis will only consider Model B to avoid installation problems.

Table 4 shows the best SL sets obtained for each case after completing the iterative process. In it, the links that are common in both sets are marked in bold, seeing how a certain amount of them remains fixed, and the others are changing due to the modification of the location model through the constraint (6). This is clearly seen in Figure 14, where the optimal locations of the sensors are outlined in 30 of the links that make up the network. In this it is seen how a set of sensors, marked in blue, are located in NSL , i.e., when Model A was used. For Model B, it is seen how those sensors are moved to other links, now marked in orange. This change in location leads to an improvement in the estimation results, indicating that there would be no problem locating sensors in links in which, despite having a lower flow, there is a greater probability of obtaining data with lower mistakes.

Table 4. Best set of links (SL) sets obtained from the variability analysis of k parameter.

| Model | Scanned Link Set SL | $RMARE$ |
|-------|---|---------|
| A1 | 1 2 3 5 10 27 31 42 43 47 48 50 51 56 58 60 85 89 100 110 137 150 151 152 153 154 155 156 158 163 | 0.2010 |
| B1 | 1 3 4 10 20 31 38 43 50 54 63 64 77 84 85 90 94 137 150 151 152 153 154 155 156 157 159 161 162 163 | 0.1976 |
| A2 | 1 2 4 5 10 38 42 43 47 48 50 51 56 58 77 85 89 97 100 110 137 151 152 153 154 155 156 158 162 173 | 0.1149 |
| B2 | 1 3 4 8 10 31 38 43 50 63 81 84 85 90 91 92 100 110 137 150 151 152 153 155 156 157 159 160 162 163 | 0.1106 |

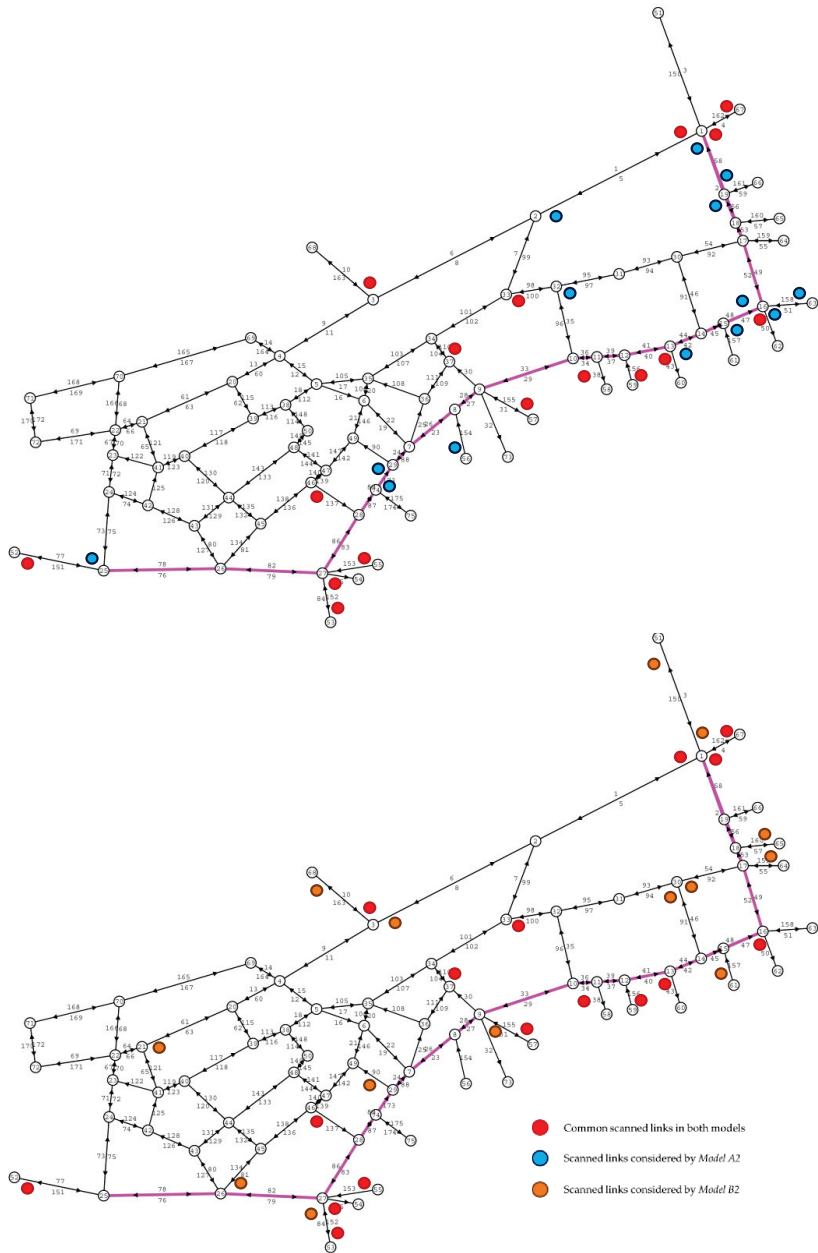


Figure 14. Upper figure: Optimal sensor locations considering $k = 3$, $F_{thres} = 10$, and all links as candidates to be scanned; Lower figure: Optimal sensor locations considering $k = 3$, $F_{thres} = 10$, and a certain set non-candidate scanned links (NSL) of links as no candidates to be scanned.

3.2.2. Effect of Network Simplification

When the value of F_{thres} is small, the proposed methodology will do a smaller simplification of the network, and therefore it is expected to lead to a lower error in the estimation of traffic flows. As F_{thres}

increases, there will be a greater degree of simplification, and therefore greater error in the estimation. Figure 15 shows this effect all the cases modelled with $k = 3$. It is observed that lower F_{thres} values, and therefore less simplification, tend to smaller error values. In any case, depending on the F_{thres} value, the graphs reached to a certain convergence after having performed multiple iterations with the proposed location model. For example, we see how the best solution is achieved with a minimum error difference when considering a F_{thres} equal to 10 or 15 and for F_{thres} equal to 20, 25, and 30.

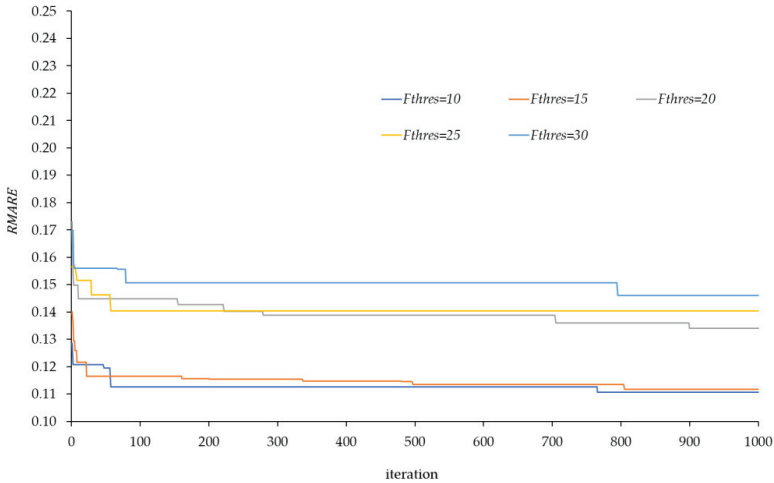


Figure 15. Evolution of RMARE value for the different cases varying the threshold flow.

The effects of variation in threshold flow are shown in Table 5. In it, a first column is defined for each evaluated case, and a second that collects the number of routes in the set R , which is the same for all of them since the same value of $k = 3$ was used; third column collects the number of routes set Q once set R has been simplified with the value of F_{thres} ; a fourth column that includes the number of additional routes included when locating the sensors in the best set obtained for each case; and a last column that considers all the routes in C used in the estimation model. In this table, it can be seen that, with less simplification, the set of routes in C with which we work is greater, and therefore the estimations are better. As the simplification increases, more routes are simplified and this means that, by locating the sensors, a greater number of routes are recovered, but a set C on similar order of magnitude.

Table 5. Number of routes that appear according to the considered F_{thres} value.

| F_{thres} | Number of Routes in R | Number of Routes in Q | Added Routes Compatibles with SL | Number of Routes in C |
|-------------|-------------------------|-------------------------|------------------------------------|-------------------------|
| 10 | 2943 | 2896 | 23 | 2919 |
| 15 | 2943 | 2816 | 30 | 2846 |
| 20 | 2943 | 2398 | 31 | 2429 |
| 25 | 2943 | 2175 | 41 | 2216 |
| 30 | 2943 | 2090 | 47 | 2137 |

Finally, Table 6 indicates the SL sets obtained for the cases where F_{thres} is equal to 10 and 15, since they offer the best results and where there is little difference in the best RMARE estimation error. We see how for both sets, there is only a difference of 8 links from the 30 considered, the rest being common in both. For both sets, constraint (6) is considered in the location model, and therefore no links belonging to NSL appears. Furthermore, this tells us how, depending on how the network is

modeled, one set or another may be obtained, with small differences but which may influence the observability and estimation results of the network.

Table 6. Best *SL* sets obtained from the variability analysis of *k* parameter.

| F_{thres} | Scanned Link Set <i>SL</i> | | | | | | | | | | | | | | | | | | | | <i>RMARE</i> | | | | | | | | | | |
|-------------|----------------------------|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|
| 10 | 1 | 3 | 4 | 8 | 10 | 31 | 38 | 43 | 50 | 63 | 81 | 84 | 85 | 90 | 91 | 92 | 100 | 110 | 137 | 150 | 151 | 152 | 153 | 155 | 156 | 157 | 159 | 160 | 162 | 163 | 0.1106 |
| 15 | 3 | 4 | 6 | 7 | 19 | 25 | 31 | 38 | 43 | 50 | 60 | 72 | 84 | 85 | 90 | 91 | 92 | 137 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 160 | 161 | 162 | 163 | 0.1117 |

4. Conclusions

This paper presents a proposal for deployment of a low-cost sensor network for automated vehicles plate recognition in a pilot project in Ciudad Real (Spain). For this, three main tools were needed: (1) the architecture to deploy the sensors, (2) a low-cost sensor prototype, and (3) a methodology to decide the best location for the sensor.

Regarding the deployment of sensors and the sensors themselves, one of the main features to highlight is that the total cost is very low in terms of the following elements:

- **Production/Manufacturing:** The unit cost of the hardware components for the realization of the prototype is less than €60 (considering the tripod as an extra accessory). In the case of integration for large scale manufacturing, these could be significantly reduced.
- **Installation:** The sensors have a very low energy consumption, which allows their deployment in any location and without specific energy supply infrastructure. The platform allows adapting the sensor parameters (resolution, lighting levels, shutter speed, and compression level) to the specific needs of each location.
- **Maintainability and scalability:** The proposed architecture allows working with any existing ANPR library in the market by delegating tasks between processing layers, as well as their combination to improve the overall success rate. The detection stage is delegated to the smart management layer, reducing overall costs, and providing more scalable and efficient solutions.

In addition, the deployed sensor is completely decoupled from the specific license plate identification platform used. This allows to change the platform if the user found any other better. In particular, the used platform identifies besides the license plate number, the vehicle's manufacturer, model, and color data. This information can be used in the overall analysis of traffic flows with a view to reducing possible errors in the identification of the number plate and will be developed in the future by the authors.

The third tool used in this paper is a methodology to determine the location in the traffic network of the designed sensors. To this, we have proposed the use of two algorithms which aim to achieve a good enough quality of the traffic flow estimation to be done (in terms of low *RMARE* value) with the ANPR data collected by the sensors.

The model was applied to the traffic network of a pilot project considering a deployment of 30 sensors analyzing whether or not to install the proposed sensors on some links due to the difficulty of its installation. The results were very positive since the expected quality of the estimation results is very similar to those obtained when allowing the sensor to be located in any link. The main advantage is that avoiding those conflictive links we expect a reduction obtaining errors of reading vehicle plates.

The influence of other parameters of the model were also analyzed such as the number of routes used as reference and the degree of network simplification. The analysis of the results shows that considering a greater number of reference routes, represented by means of the parameter *k*, leads to a better estimation of the flows in terms of achieving a smaller *RMARE*. However, a high value for *k* would imply working with a network with a large number of routes, which would have a high computational cost. In reference to network simplification, a medium–low degree of network simplification leads to a good performance of the methodology in terms of the error obtained in the estimation step.

Author Contributions: Conceptualization, all authors; methodology, F.Á.-B. and S.S.-C.; location model, F.Á.-B., S.S.-C., A.R., and I.G.; architecture of sensor deployment, D.V., C.G.-M., and S.S.-C.; validation, all authors; formal analysis, F.Á.-B. and S.S.-C.; investigation, F.Á.-B. and D.V.; resources, A.R. and I.G.; data curation, S.S.-C. and C.G.-M.; writing—review and editing, F.Á.-B., S.S.-C., D.V., and C.G.-M.; project administration, A.R. and S.S.-C.; and funding acquisition, A.R. and S.S.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded thanks to the financial support of the Spanish Ministry of Economy and Competitiveness in relation to project TRA2016-80721-R (AEI/FEDER, UE).

Acknowledgments: The authors acknowledge Miguel Carrión (University of Castilla-La Mancha) and the university's technical staff for providing computer resources. We also want to acknowledge the editor and the two anonymous reviewers whose valuable comments have contributed to improve this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Guerrero-Ibáñez, J.A.; Zeadally, S.; Contreras-Castillo, J. Sensor Technologies for Intelligent Transportation Systems. *Sensors* **2018**, *18*, 1212. [[CrossRef](#)] [[PubMed](#)]
- Wang, J.; Jiang, C.; Han, Z.; Ren, Y.; Hanzo, L. Internet of Vehicles: Sensing-Aided Transportation Information Collection and Diffusion. *IEEE Trans. Veh. Technol.* **2018**, *67*, 3813–3825. [[CrossRef](#)]
- Wang, J.; Jiang, C.; Zhang, K.; Quek, T.Q.; Ren, Y.; Hanzo, L. Vehicular Sensing Networks in a Smart City: Principles, Technologies and Applications. *IEEE Wirel. Commun.* **2017**, *25*, 122–132. [[CrossRef](#)]
- Gentili, M.; Mirchandani, P.B. Review of optimal sensor location models for travel time estimation. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 74–96. [[CrossRef](#)]
- Auberlet, J.-M.; Bhaskar, A.; Ciuffo, B.; Farah, H.; Hoogendoorn, R.; Leonhardt, A. Data collection techniques. In *Traffic Simulation and Data. Validation Methods and Applications*; CRC Press: Boca Raton, FL, USA, 2014; pp. 5–32.
- Li, X.; Ouyang, Y. Reliable sensor deployment for network traffic surveillance. *Transp. Res. Part B Methodol.* **2011**, *45*, 218–231. [[CrossRef](#)]
- Anagnostopoulos, C.-N.; Anagnostopoulos, I.; Psoroulas, I.; Loumos, V.; Kayafas, E. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 377–391. [[CrossRef](#)]
- Sonavane, K.; Soni, B.; Majhi, U. Survey on automatic number plate recognition (anr). *Int. J. Comput. Appl.* **2015**, *125*, 1–4. [[CrossRef](#)]
- Patel, C.; Shah, D.; Patel, A. Automatic number plate recognition system (anpr): A survey. *Int. J. Comput. Appl.* **2013**, *69*, 21–33. [[CrossRef](#)]
- Ullah, F.; Anwar, H.; Shahzadi, I.; Rehman, A.U.; Mehmood, S.; Niaz, S.; Awan, K.; Khan, A.; Kwak, D. Barrier Access Control Using Sensors Platform and Vehicle License Plate Characters Recognition. *Sensors* **2019**, *19*, 3015. [[CrossRef](#)]
- Castillo, E.F.; Conejo, A.J.; Menéndez, J.M.; Jiménez, M.D.P. The observability problem in traffic network models. *Comput. Civ. Infrastruct. Eng.* **2008**, *23*, 208–222. [[CrossRef](#)]
- Cerrone, C.; Cerulli, R.; Gentili, M. Vehicle-ID sensor location for route flow recognition: Models and algorithms. *Eur. J. Oper. Res.* **2015**, *247*, 618–629. [[CrossRef](#)]
- Sánchez-Cambronero, S.; Álvarez-Bazo, F.; Rivas, A.; Gallego, I. A new model for locating plate recognition devices to minimize the impact of the uncertain knowledge of the routes on traffic estimation results. *J. Adv. Transp.* **2020**, *2020*, 1–20. [[CrossRef](#)]
- Castillo, E.F.; Nogal, M.; Rivas, A.; Sánchez-Cambronero, S. Observability of traffic networks. Optimal location of counting and scanning devices. *Transp. B Transp. Dyn.* **2013**, *1*, 68–102. [[CrossRef](#)]
- Mínguez, R.; Sánchez-Cambronero, S.; Castillo, E.F.; Jiménez, M.D.P. Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks. *Transp. Res. Part B Methodol.* **2010**, *44*, 282–298. [[CrossRef](#)]
- Istin, C.; Pescaru, D.; Doboli, A. Stochastic Model-Based Heuristics for Fast Field of View Loss Recovery in Urban Traffic Management Through Networks of Video Cameras. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 895–907. [[CrossRef](#)]
- Castillo, E.F.; Rivas, A.; Jiménez, M.D.P.; Menéndez, J.M. Observability in traffic networks. Plate scanning added by counting information. *Transp.* **2012**, *39*, 1301–1333. [[CrossRef](#)]

18. Sanchez-Cambronero, S.; Jiménez, M.D.P.; Rivas, A.; Gallego, I. Plate scanning tools to obtain travel times in traffic networks. *J. Intell. Transp. Syst.* **2017**, *21*, 390–408. [CrossRef]
19. Castillo, E.F.; Gallego, I.; Menendez, J.M.; Rivas, A. Optimal Use of Plate-Scanning Resources for Route Flow Estimation in Traffic Networks. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 380–391. [CrossRef]
20. Sánchez-Cambronero, S.; Castillo, E.; Menéndez, J.M.; Jiménez, M.D.P. Dealing with Error Recovery in Traffic Flow Prediction Using Bayesian Networks Based on License Plate Scanning Data. *J. Transp. Eng.* **2011**, *137*, 615–629. [CrossRef]
21. U.S. Department of Transportation. Available online: <https://www.itsbenefits.its.dot.gov/ITS/benecost.nsf/ID/20746487B947B3358525797C006528CF> (accessed on 31 July 2020).
22. Bernaś, M.; Placzek, B.; Korski, W.; Loska, P.; Smyła, J.; Szymała, P. A Survey and Comparison of Low-Cost Sensing Technologies for Road Traffic Monitoring. *Sensors* **2018**, *18*, 3243. [CrossRef]
23. Carrabs, F.; Cerulli, R.; D’Ambrosio, C.; Gentili, M.; Raiconi, A. Maximizing lifetime in wireless sensor networks with multiple sensor families. *Comput. Oper. Res.* **2015**, *60*, 121–137. [CrossRef]
24. Du, R.; Gkatzikis, L.; Fischione, C.; Xiao, M. On Maximizing Sensor Network Lifetime by Energy Balancing. *IEEE Trans. Control. Netw. Syst.* **2017**, *5*, 1206–1218. [CrossRef]
25. Github. Available online: <https://github.com/openalpr/openalpr> (accessed on 31 July 2020).
26. OpenALPR. Available online: <https://www.openalpr.com/benchmarks.html> (accessed on 31 July 2020).
27. Plate Recognizer. Available online: <https://platerrecognizer.com/> (accessed on 31 July 2020).
28. SD-Toolkit. Available online: https://www.sd-toolkit.com/products_sdtanpr_windows.php (accessed on 31 July 2020).
29. Anyline. Available online: <https://anyline.com/> (accessed on 31 July 2020).
30. Eocortex. Available online: <https://eocortex.com/products/video-management-software-vms/license-plate-recognition> (accessed on 31 July 2020).
31. Izidio, D.; Ferreira, A.; Medeiros, H.; Barros, E. An embedded automatic license plate recognition system using deep learning. *Des. Autom. Embed. Syst.* **2020**, *24*, 23–43. [CrossRef]
32. IntelliVision. Available online: <https://www.intelli-vision.com/license-plate-recognition/> (accessed on 31 July 2020).
33. Genetec. Available online: <https://www.genetec.com/solutions/all-products/autovu> (accessed on 31 July 2020).
34. Handscombe, J.; Yu, H. Low-Cost and Data Anonymised City Traffic Flow Data Collection to Support Intelligent Traffic System. *Sensors* **2019**, *19*, 347. [CrossRef] [PubMed]
35. Gentili, M.; Mirchandani, P. Locating sensors on traffic networks: Models, challenges and research opportunities. *Transport. Res. Part C* **2012**, *24*, 227–255. [CrossRef]
36. Hadavi, M.; Shafabi, Y. Vehicle identification sensors location problem for large networks. *J. Intell. Transp. Syst.* **2019**, *23*, 389–402. [CrossRef]
37. Google. Available online: <https://cloud.google.com/appengine> (accessed on 31 July 2020).
38. Yen, J. Finding the k-Shortest Loopless Paths in a Network. *Manag. Sci.* **1971**, *17*, 661–786. [CrossRef]
39. Owais, M.; Moussa, G.; Hussain, K. Sensor location model for O/D estimation: Multi-criteria meta-heuristics approach. *Oper. Res. Perspect.* **2019**, *6*, 1–12. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

The Effects of the Driver's Mental State and Passenger Compartment Conditions on Driving Performance and Driving Stress

Víctor Corcoba Magaña ^{1,*}, Wilhelm Daniel Scherz ², Ralf Seepold ^{2,3},
Natividad Martínez Madrid ^{3,4}, Xabiel García Pañeda ¹ and Roberto Garcia ¹

¹ Department of Computer Science, University of Oviedo, 33003 Oviedo, Spain; xabiel@uniovi.es (X.G.P.); garciaroberto@uniovi.es (R.G.)

² Ubiquitous Computing Lab, Department of Computer Science, University of Technology, Business and Design Konstanz, 78462 Konstanz, Germany; wscherz@htwg-konstanz.de (W.D.S.); ralf@ieee.org (R.S.)

³ Institute of Digital Medicine, I.M. Sechenov First Moscow State Medical University, 119435 Moscow, Russia; nati@ieee.org

⁴ IoT Lab, School of Informatics, Reutlingen University, 72762 Reutlingen, Germany

* Correspondence: corcobavictor@uniovi.es; Tel.: +34-985-182-277

Received: 30 July 2020; Accepted: 11 September 2020; Published: 15 September 2020

Abstract: Globalization has increased the number of road trips and vehicles. The result has been an intensification of traffic accidents, which are becoming one of the most important causes of death worldwide. Traffic accidents are often due to human error, the probability of which increases when the cognitive ability of the driver decreases. Cognitive capacity is closely related to the driver's mental state, as well as other external factors such as the CO₂ concentration inside the vehicle. The objective of this work is to analyze how these elements affect driving. We have conducted an experiment with 50 drivers who have driven for 25 min using a driving simulator. These drivers completed a survey at the start and end of the experiment to obtain information about their mental state. In addition, during the test, their stress level was monitored using biometric sensors and the state of the environment (temperature, humidity and CO₂ level) was recorded. The results of the experiment show that the initial level of stress and tiredness of the driver can have a strong impact on stress, driving behavior and fatigue produced by the driving test. Other elements such as sadness and the conditions of the interior of the vehicle also cause impaired driving and affect compliance with traffic regulations.

Keywords: driving safety; driving emotions; driving stress; lifestyle; sensors; heart rate

1. Introduction

In 2018, there were 102,299 traffic accidents with victims in Spain, with 1806 people losing their lives [1]. Most of these accidents happened in cities. Distractions were the main cause of fatal accidents at 32%, speed was at 22% and alcohol or drug consumption was at 21%. Most accidents are due to human error [2,3]. Other factors such as the environment or the vehicle involve less hazard [4].

In the literature, we find many works that evaluate driving performance and the drivers' physiological and cognitive states [5,6]. For example, in [5], the authors proposed a method to determine a driver's relative stress level based on analyzing physiological data and artificial intelligence. Twenty-four drivers participated in the experiment. The authors monitored the participants in real driving for at least 50 min. The results showed that the conductivity of the skin and the pulse metrics are those that most correlate with the level of stress. In [6], the changes in the muscles of the shoulder and the neck were analyzed while the participant drove in a driving simulator. Professional and non-professional drivers participated in the study. The conclusions were that, in both cases, the drivers

suffered from fatigue after driving for 15 min. Many researchers have found a strong relationship between emotions and traffic accidents. Researchers in [7] conducted a cluster analysis on responses to a survey by more than 1500 college students about potentially annoying driving-related situations. The authors noted that men were more angered by the police presence and slow driving. On the other hand, women were more angered by illegal behavior and obstructions. Finally, they concluded that knowing the driver's anger could be of great help in reducing traffic accidents. Negative emotions, such as fear, anger, disgust or sadness, increase the probability of manifesting dangerous behaviors while driving. In [8], the authors analyzed the influence of mood on risk perception and attitude. The authors conducted an experiment where they induced different emotions in the driver by watching videos. The results showed that negative emotions significantly increase the perception of risk by the driver, but also cause an inappropriate attitude. In [9], the authors focused on studying how sadness affects driving performance. Sixteen drivers participated in the experiment. They used a driving simulator. The results showed that drivers with sadness make more driving mistakes than neutral drivers. Furthermore, the driving times is also longer. In [10], the authors explore the use of affective interfaces in vehicles. The researchers conducted an online survey on emotional situations on the road. The results showed that drivers who experienced negative situations require better information management and a high degree of automation. In contrast, drivers with positive emotions prefer a more genuine driving mode.

The state of the interior of the vehicle is another factor to consider. Other aspects such as the number of occupants in the vehicle or adequate ventilation should also be taken into account. It has been shown that a high concentration of CO₂ decreases cognitive ability and increases fatigue. In [11], the authors analyzed the performance in a cognitive test when the participants were exposed to different levels of CO₂. Participants obtained significantly better cognitive scores when the CO₂ level was lower than 1400 ppm. Similar results were obtained in [12]. In this work, the researchers analyzed the effects of CO₂ on decision making. Twenty-two people participated in the experiment who took a decision-making test. In addition, the participants also filled out a questionnaire about their health and perceived air quality. The authors found that there was a moderate worsening of decision-making performance at 1000 ppm of CO₂.

Autonomous vehicles could be a solution to reduce traffic accidents caused by human errors. However, on the one hand, these solutions still have a high cost and are not accurate enough [13]. In addition, in most countries, there are still no laws or regulations for this type of vehicle [14,15]. On the other hand, the autonomous vehicles that have been developed to date require that the driver maintain attention on the road and if an exceptional event takes place, he or she must take control. The first accidents have already occurred due to a lack of attention and the need for traditional and autonomous vehicles to coexist. In [16], the author used a Tesla Model S vehicle for six months. In this article, he analyzed the problems that these vehicles present for the driver in real driving conditions. The researcher points out that these vehicles can improve safety and driving comfort, but also can present challenges due to the transition between automatic and manual driving mode. Finally, a series of guidelines are proposed to improve the way in which this type of vehicle interacts with the driver. Reducing human errors while driving remains a significant concern.

The emotions experienced by the driver can be caused by events that occur during the driving task or before. In [17], McMurray analyzed driving reports. The researcher concluded that drivers who were in the process of a divorce had a higher probability of suffering traffic accidents and violating traffic rules than other drivers. Similar results were obtained by [18], where the authors examined how stressful environments and psychological characteristics of the driver affect driving behavior. The results showed that stress significantly increases the risk of traffic accidents. The researchers also highlighted the importance of experiential knowledge acquired without instruction to present good driving behavior. In [19], the authors found that financial problems increase the likelihood of suffering a traffic accident. Therefore, events unrelated to the driving task influence stress and driving skills.

Many of the works that analyze the driver's mental state and its effect on driving are based on artificially inducing a certain emotion using music, images or words [20]. For example, in [21], the researchers studied the effect of joy, sadness and anger on driving behavior using induced emotions. The results showed that negative emotions cause dangerous driving behaviors. The authors also observed that, in some cases, the emotional state did not affect driving when the workload was high. In [22], the authors conducted an experiment where participants were induced with sadness, anger or neutral emotions. Participants with induced anger or sadness made more driving mistakes than participants induced with neutral emotions.

Originally, the works that analyzed drivers' emotions used subjective measures [7]. These methods require the direct intervention of the participant. Therefore, the samples are obtained at a low rate. In recent years, the psychological methods have become less expensive and intrusive due to wearables. These devices allow us to monitor the driver continuously and without requiring his direct involvement. [23]. During driving, the use of non-intrusive devices is essential so as not to affect driving performance or cause safety problems [24,25]. In [26], we can find a review of the solutions to monitor a driver's psychological state.

Contribution

Most of the works documented in the literature focus on ascertaining the emotions of the driver while driving. There are some studies in which the driver's previous state is analyzed, but they are minimal. Furthermore, they do not investigate how emotions together with factors such as the interior state of the vehicle (temperature, humidity and CO₂ concentration) or the music the driver listens to affect driving and the level of stress. Another problem we have found in previous works is that they artificially induced moods, which could lead to inaccuracies in the results.

The objective of this work is to analyze which elements affect driving behavior and stress levels, focusing on the drivers' initial emotions, their characteristics and the comfort inside the car. The conclusions of this analysis can be used to include the emotional component in driving assistants. Most of them are limited to warning the driver when they invade the opposite lane or exceed the speed limit.

2. Related Work on Stress Detection

Stress is defined as a state of physical, psychological or emotional health experienced by a person when the perceived or actual demand requires a high number of resources [27]. Stress appears when the demand for mental workload exceeds the capabilities of the subject [28]. Stress may be accompanied by other emotions such as anxiety [29]. However, it is not always bad. Healey et al. [5] classified stress into two types: eustress and distress. Eustress encourages people to achieve a high level of performance. If the level of stress is too low, it can cause drivers to suffer fatigue and drowsiness, and they may lose control of the vehicle [30]. Distress appears when there is an excess in the level of demand that surpasses the capacity of the person and consequently discourages the driver [31]. The level of stress experienced while driving can be affected by four factors: the physical and mental condition of the driver, road and traffic conditions, vehicle condition and external disturbances. This paper focuses on the driver's mental state and vehicle condition (music tempo and CO₂ concentration).

Stress detection methods can be classified into four categories:

- Self-report questionnaire assessment;
- Physiological measures;
- Driving behavior monitoring;
- Visual-based and speech detection.

The self-report questionnaires analyze the driver's behavior and strategies for coping with different types of stressful events. In addition, the characteristics of the driver are very important [32]. Data such as age or accident history have a strong relationship with stress. Drivers who have suffered

more traffic accidents are more likely to feel anxiety and develop post-traumatic stress disorders [33]. One of the most used questionnaires in research is the driver behavior inventory (DBI) [34]. In this questionnaire, stress is defined by five elements: driving aggression, dislike of driving, tension and frustration connected with successful or unsuccessful overtaking, irritation when overtaken and heightened alertness and concentration. There are also many other questionnaires such as: the driving stress inventory (DSI) [35], stress arousal checklist (SACL) [36] and Dundee stress state questionnaire (DSSQ) [37]. In the case of workload measurement, NASA load index (NASA-TLX) [38] and driving activity load index (DALI) [39] are the most widely used. In the experiments, several of them can be used with different objectives. For example, in [40], the participants completed the DSI questionnaire before the test to estimate their vulnerability to stress. They then completed the DSSQ questionnaire to analyze the stress and workload caused by the task.

Stress detection models based on physiological signals allow us to objectively monitor the driver's stress level in real time. They mainly use the heart rate signal, skin conductance, skin temperature and the encephalogram [41,42]. The main disadvantage of these methods is that they require the use of sensors, which increases the cost and reduces the number of potential participants. In addition, these solutions can cause discomfort if they are intrusive. However, in recent years, wearable devices have been developed that can monitor the driver without affecting mobility and at a relatively low cost. An example widely used in research is the Empatica E4 [43]. These portable devices are not as accurate as medical devices. However, there is a strong correlation between them, and they are valid for measuring stress and conducting long-term studies [44,45].

In the literature, we find many proposals of stress detection based on these types of signals. A strong relationship between driving stress and heart rate and blood pressure was reported in [32,46]. In [47], the authors proposed a binary logistic regression model to predict driving stress. This method uses galvanic skin response data obtained in real road driving situations to predict whether driver stress will be high or low. GSR data were collected using a wearable device (Empatica E4). The main advantage of this solution is that it is non-intrusive so it can be used in real driving. The authors achieved an accuracy higher than 80% using this model. In [48], the authors wanted to analyze the relationships between driving stress, traffic conditions and road types. The authors proposed using electrodermal activity (EDA) signals to estimate the levels of driving stress taking into account the road type and traffic conditions. The classification model developed was based on the data collected by a driver in real road driving conditions for 60 min a day for 21 days. The results showed that traffic conditions and road type are factors that influence driving stress.

Proposals using physiological signals can detect driver stress in real time using artificial intelligence algorithms [5,49]. Galvanic skin response (GSR) and heart rate variability (HRV) are considered the best indicators of stress in real time [5]. However, we should take into account the latency that in the case of skin conductivity can be up to 1.4 s [50].

A different alternative to using these sensors is to analyze the driver's face and speech. This avoids having to wear sensors. For example, the authors in [51] used visual-based thermography to detect facial skin temperature. In [52], the authors proposed to analyze facial expression using an NIR camera. The drawback is that good illumination is required to achieve accuracy in stress detection. Voice speech is another variable that can be helpful for detecting stress. In [53], the authors analyzed the changes in pitch of the subjects to detect stress. The problem with this type of approach is that it requires the driver to perform additional tasks while driving in order to make the voice recording, which could cause distractions [54]. In addition, noise inside the vehicle cabin could make it difficult to detect stress [55].

There are some proposals to detect stress based on driving behavior. The authors in [56] highlighted that the autonomic system (ANS) and driving style change when the level of stress is high. Stressful events can be detected by analyzing the corrections the driver makes with the steering wheel and the pedals of the vehicle. In [57], the researchers proposed a system that monitors the turning patterns of the steering wheel and recognizes lanes and accelerating patterns in order to detect stress.

Finally, there are proposals that combine the use of physiological signals with vehicle telemetry (steering wheel movement, acceleration, deceleration). In [58], the authors presented a wearable glove system for monitoring stress while driving. The proposal extracted features of photoplethysmography (PPG) and inertial measurement unit (IMU) sensors located in the glove to assess the stress events. The proposal was able to detect stress events with an accuracy rate of over 95% using an SFS-SVM classifier with the RBF kernel function. The main limitation of this device is that participants cannot change the position of their hands on the steering wheel during the driving test.

3. Materials and Methods

In this section, we will describe the materials and the procedure to carry out the experiment. We present the sensors (Figure 1) used to monitor driving stress, to evaluate driving performance and to obtain the state of the simulation environment (temperature, humidity and CO₂ level). Driving stress is tracked using an Empatica E4 wristband and a Polar H10 chest band. The environment is supervised using the Netatmo device. We also define the measurements that we will use to evaluate the drivers and their driving behavior from the data gathered by the sensors. In addition, the test scenario will be detailed, explaining the simulator used, as well as the music that the driver listened to during the driving task. It will specify the survey completed by the drivers before and after the test to ascertain their characteristics, their opinion of the experiment and their physical and mental state.



Figure 1. Sensors used in the experiment.

3.1. Heart Signal

We have used several sensors in this work to measure stress objectively. One of the vital signs most used in research on driving is the heart rate variability (HRV). There are numerous studies where this biosignal is measured due to its strong correlation with stress, and the fact that it can be obtained in a non-intrusive way [5].

Heart rate variability can be analyzed in two different domains: time and frequency. Time domain analysis of the HRV signal consists of measuring the mean or standard deviation of the time intervals between consecutive heartbeats. Frequency domain analysis is a method based on the amount of heart signal found in two different frequency bands. In the case of heart rate analysis, the ranges are (0.04–0.15 Hz) and (0.15–0.4 Hz). In this work, we use the following measurements obtained from the heart rate signal that have been widely used in the literature [59]:

- pNN50 (%): this is the number of consecutive heartbeats differing more than 50 ms divided by the total number of measured heartbeats and expressed as a percentage. This variable decreases when driving stress is high.

- LF/HF: this is the low-frequency (LF) power (0.04–0.15 Hz) modulated by the sympathetic and parasympathetic nervous system divided by the high-frequency (HF) power (0.15–0.4 Hz) associated with the parasympathetic nerve activity. This ratio captures the global sympathovagal balance [25]. A high LF/HF ratio means sympathetic dominance, which happens when driving stress is elevated.

There are several non-intrusive devices which obtain the heart rate. Many of them are based on the photo-plethysmography sensor that allows us to measure the blood volume pulse. This type of device has improved significantly in recent years. However, they are very sensitive to movement and the level of pressure on the skin [60]. Another type of sensor that allows measuring the heart rate non-intrusively is a chest band with electrodes. These solutions achieve higher accuracy than the photo-plethysmography sensor [61,62].

In our case, the heart signal is obtained using a Polar H10 chest band. Polar H10 is the successor to the Polar H7 device. Polar H10 introduces improvements to measure heart rate variability. This device can offer precision for measuring the time between successive heartbeats (also called RR interval) similar to that obtained by a Holter ECG [61]. Polar H10 was connected wirelessly to the EliteHRV[®] app [63] running on a Google Pixel 3a. This app directly receives the RR intervals and uses a proprietary algorithm to correct artifacts such as ectopic beats or signal noise to present a more valid signal for heart rate analysis (HRV). We exported the calculated HRV values (pNN50 and LF/HF ratio) from the web dashboard provided by this app. It is also important to mention that before starting the heart rate measurement, there is a 30 s sensor stabilization period. Table 1 shows the Polar H10 specifications.

Table 1. Polar H10 specifications. Data from [64].

| | |
|------------------------------|--------------------------------------|
| Battery Type | CR 2025 |
| Battery sealing ring | O-ring 20.0 × 0.90 Material Silicone |
| Battery lifetime | 400 h |
| Sampling rate | 1 Hz |
| Operating temperature | −10 °C to +50 °C/14 °F to 122 °F |
| Connector material | ABS, ABS + GF, PC, Stainless steel |
| Strap material | 38% Polyamide, 29% Polyurethane, 20% |

3.2. Skin Conductivity

The variation of skin conductivity is linked to the sympathetic nervous system [62]. When the driver has high stress, the activity of sweat glands is triggered by postganglionic sudomotor fibers. The result is a change in the skin conductivity response (SCR) that can be measured by applying a low constant voltage. The SCR amplitude can be used as an indicator of sympathetic activity [33].

Skin conductivity is monitored using an Empatica E4 wristband. Table 2 [65] shows the characteristics of the sensors that the Empatica E4 device integrates. This device is certified as CE Medical class 2a [66] and has been validated in many works [67,68]. It includes a photo-plethysmography sensor that allows us to measure the blood volume pulse. It also has a galvanic sensor to measure sympathetic nervous system arousal as well as to derive features related to stress, engagement and excitement. The wristband features are a 3-axis accelerometer to capture motion-based activity and an infrared ray which reads peripheral skin temperature. This device has been designed for continuous, real-time data acquisition.

Table 2. Empatica E4 specifications. Data from [65].

| | |
|-----------------------------|--|
| PPG sensor | <ul style="list-style-type: none"> • Sampling frequency 64 Hz • LEDs: Green (2 LEDs), Red (2 LEDs) Photodiodes: 2 units, total 15.5 mm² sensitive area • Sensor output resolution 0.9 nW/Digit |
| EDA sensor | <ul style="list-style-type: none"> • Sampling frequency: 4 Hz • Resolution: 1 digit ~900 pSiemens • Range: 0.01 μSiemens–100 μSiemens |
| Infrared thermopile | <ul style="list-style-type: none"> • Sampling frequency: 4 Hz • Range: −40 ... 115 °C • Resolution: 0.02 °C • Accuracy \pm0.2 °C within 36–39 °C |
| 3-Axis accelerometer | <ul style="list-style-type: none"> • Sampling frequency: 32 Hz • Range \pm2 g |
| E4 operating range | <ul style="list-style-type: none"> • Relative Humidity 60 \pm 25% H.R. |
| Water resistance | <ul style="list-style-type: none"> • IP 22 |
| Memory | <ul style="list-style-type: none"> • Device storage capacity exceeds 60 recording hours. |
| Connectivity | <ul style="list-style-type: none"> • Bluetooth LE • Operating range: 10 m |
| Size | <ul style="list-style-type: none"> • 44 \times 30 \times 16 mm |

However, the calculation of the amplitude is not trivial. Usually, the SCRs overlap each other. In the standard peak detection method (trough-to-peak), the SCR amplitude is obtained by calculating the difference between the peak and the previous trough of the skin conductance data. This results in an underestimation of the amplitude of subsequent SCRs. The degree of underestimation depends on the amplitude and proximity of the preceding SCRs. There are different proposals in the literature to avoid this problem. In this paper, a deconvolution approach [35] is used, which separates skin conductivity data into continuous signals of tonic and phasic activity. This algorithm allows us to represent the overlapping SCRs by compact impulses, thus avoiding the underestimation problem. To that end, we use Ledalab 3.4.9 [69], which is recommended by Empatica. Before the signal deconvolution by continuous decomposition analysis, we pre-process it to eliminate high-frequency noise by applying a smoothening filter consisting of a 4-sample Gaussian window.

3.3. Environments

Temperature, humidity and CO₂ concentration are variables that influence comfort and safety [70]. Vehicles tend to circulate in areas that are heavily contaminated. Many drivers close the windows and use the air conditioning in order to avoid polluting gases. However, the air that comes from this system is not clean. Besides, the reduced space of the vehicle causes a high amount of CO₂ to accumulate due to the passengers themselves. If the level of CO₂ inside the vehicle is very high, the driver may suffer from dizziness and nausea [71].

Temperature and humidity are other factors that can induce fatigue in the driver [72]. In the past, temperature was a significant cause of traffic accidents [73]. Currently, most vehicles integrate an air conditioning system. However, it is very difficult to adjust it correctly because the thermal sensation is different for each passenger [74]. An inadequate temperature, either too high or too low, causes a significant worsening of driving performance [75]. In order to monitor the interior of the vehicle, we used a Netatmo Healthy Home Coach [76]. This system allows us to obtain the air temperature, relative humidity and CO₂ concentration. The measurements are taken every five minutes, and are uploaded to the cloud instantly. The data are processed internally using proprietary

Netatmo algorithms. We directly downloaded the temperature, humidity and CO₂ values using a Python script. We could not obtain the raw data. Regarding the validity of the use of the device to measure the CO₂ concentration, there are several works where it has been verified, providing that a calibration has been previously performed [77,78].

The Home Coach Netatmo device allows manual calibration of the CO₂ and temperature sensor. The calibration of the CO₂ sensor of the device was carried out for 8 h inside a room without any occupants following the manufacturer’s instructions [79]. The temperature of the room during calibration was 25 °C. The temperature sensor was calibrated by calculating the average difference between the measurement obtained by the Netatmo device and the value offered by a weather station belonging to the State Meteorological Agency (AEMET), located in Asturias. The samples were collected over 7 days at 11 A.M. As a result, we fitted the temperature by +0.1 °C. In order to obtain the temperature reading using the Netatmo device at the location of the weather station, an external Samsung power bank was used. The Netatmo device has been validated by other authors previously obtaining good accuracy when the value of the air temperature under investigation is close to the air temperature in which the manual calibration occurs. In our case, the temperature during the calibration and test drive was very similar. Table 3 shows the specifications of the air quality sensor.

In our experiment, the temperature and humidity remained constant and we analyzed the CO₂ concentration. In the literature, we find works where the CO₂ concentration inside the vehicle cabin is analyzed [80]. However, we have not found works which study how a high CO₂ concentration influences driving stress and driving behavior.

Table 3. Specifications of the Netatmo Indoor Air Quality Monitor. Data from [81].

| | | |
|------------------------------------|-------------------|--|
| Temperature | Range Accuracy | 0 °C to 50 °C ±0.3 °C |
| Humidity | Range Accuracy | 0 to 100% ±3% |
| CO₂ | Range Accuracy | 0 to 5000 ppm ±50 ppm (from 0 to 1000 ppm) or ±5% (from 1000 to 5000 ppm) |
| Sound meter | | Ranges from: 35 to 120 dB |
| Records frequency | | Every 5 min |
| Connectivity specifications | | Wi-Fi 802.11 b/g/n compatible (2.4 GHz) Supported security: Open/WEP/WPA/WPA2-personal (TKIP and AES) |
| Size | | 45 × 45 × 155 mm |

3.4. Driving Simulator

This experiment was carried out using the “City Car Driving” simulator [82]. The simulator uses advanced car physics to achieve a realistic car feeling and a high-quality render engine for graphical realism. The simulator implements German traffic rules and warns drivers if they fail to comply with some of these. Traffic density can also be adjusted with the simulator. The drivers’ behavior and pedestrians’ behavior are sometimes erratic as in a real environment. The vehicles can collide with the player’s car or with each other. Pedestrians sometimes cross the road in the wrong places. The scene selected for the experiment is named “Old District” and is characterized by narrow streets with simple crossing places and clear traffic patterns. This driving simulator was developed to train novice drivers in driving schools. It saves a log file with all the traffic rules that the driver violated as well as events such as traffic accidents.

The execution of the driving simulator on a computer and features are included in Table 4. Three 27-inch screens were connected to the computer. To operate the vehicle, we employed a Logitech G29 [83]. This device is an electronic steering wheel designed for driving video games with realistic

force feedback. It includes a set of three pedals and a gearbox, and it allows us to archive an immersive perception in the virtual environment. Table 5 shows the specifications of the device.

Table 4. Specifications of the PC on which the driving simulator is run.

| | |
|------------------|-------------------------|
| Model | Alienware Area-51 R4 |
| Processor | Intel Core i7-7800X |
| Chipset | Intel X299 PCH |
| Memory | 16 GB DDR4 2666 MHz |
| GPU | 2 X Geforce 1080 TI SLI |
| Storage | 128 GB SanDisk M.2 SSD |

Table 5. Specifications of the G29. Data from [84].

| | |
|----------------------|--|
| Wheel | Rotation: 900 degrees lock-to-lock Hall-effect steering sensor Dual-Motor Force Feedback Overheat safeguard |
| Pedals | Nonlinear brake pedal Patented carpet grip system Textured heel grip Self-calibrating |
| Size | Wheel: 270 × 260 × 278 mm Pedals: 167 × 428.5 × 311 mm |
| Connection | USB 2.0 |
| Compatible OS | Windows 10, 8.1 Windows 8 or Windows 7 macOS 10.10 Playstation 4 or Playstation 3 |

In order to evaluate driving performance, we have developed a program based on the SFML library [85] that captures the angle of rotation of the steering wheel and the pressure applied by the participant on the pedals. In this study, we have defined the following variables to assess driving behavior:

- **Harsh braking:** this is the percentage of time that the driver stopped abruptly concerning the total braking time. We have considered that the driver brakes sharply when the deceleration is -2.5 m/s^2 or more. This value is considered by many authors as abrupt [86].
- **Braking time:** this is the time of the total driving time (25 min) that the driver was pressing the brake pedal and is expressed as a percentage.
- **Harsh acceleration:** this is the percentage of time that the driver sped up abruptly with respect to the total acceleration time. We have considered that the driver accelerates sharply when the value is 1.5 m/s^2 or more. This value is considered by many authors as abrupt [86].
- **Acceleration time:** this is the time of the total driving time (25 min) that the driver was pressing the accelerator pedal and is expressed as a percentage.

3.5. Music Tempo

Many people listen to music when they are driving. Studies show that music influences human behavior. In supermarkets, fast music causes customers to move faster through the store [87]. In bars, fast music makes people consume their drinks quickly [88]. The music tempo also causes an effect on the speed and accuracy of the tasks. In [89], fast music increased the rate and accuracy of mathematical computations in stock market environments. In driving, fast music also instigates the driver to drive faster [90]. However, in music, many parameters can affect the driver, such as the genre of music, instruments or volume, but the tempo is one of the most important in driving.

In the experiment, the participants listened to music through headphones. Drivers could adjust the volume according to their preferences to avoid discomfort. We created two playlists on Spotify [91]. One includes music with a slow tempo (65–71 bpm), and the other contains audio tracks with a fast tempo (155–188 bpm). Each of the participants was randomly assigned one of the two lists. Further, the song “Sonata for Two Pianos in D major” from Mozart was used to relax the participant at the beginning of the experiment. All songs were reproduced with the best sound quality that Spotify allows (OGG, 320 Kbps). In addition to the music, the drivers listened through headphones to the sound of the vehicle’s engine.

3.6. Survey

The participants completed two surveys: one at the beginning of the experiment and another at the end. The purpose of the pre-test survey was to obtain driver characteristics and the emotional and physical states. The survey contains questions about the level of stress, fatigue and sadness that the driver feels before the driving test. The participant should respond using a Likert scale. A Likert scale is a psychometric scale used in educational and social sciences research that employs questionnaires [92]. The Likert scale is composed of a set of statements (items). Participants are asked to show their level of agreement (from strongly disagree to strongly agree) with the given statement (items) on a metric scale. In our case, the scale is between 1 and 5, where 1 means that he or she does not suffer from that symptom or emotion and 5 that he or she develops it to a high degree.

The post-test survey was focused on ascertaining the emotional and physical states after completing the driving task. The objective was to check if the driving task had emotionally affected the driver. As previously mentioned, it comprises queries about the level of stress, fatigue and sadness that the driver feels but in this case after the driving test. This survey also includes questions about the degree of realism of the simulator, the satisfaction level of the drivers with their driving performance and the environmental conditions (temperature, noise and humidity). These questions will allow us to check if the subjective opinion of the participant corresponds to the data gathered by the sensors and if the simulator is realistic enough to infer that in a real environment, we would obtain similar results.

3.7. Procedure Description

First, the sensors were fitted to the participant, who then completed the initial survey. Then, he or she listened to Mozart’s Sonata for Two Pianos in D major using headphones. This track has been selected because it improves mental function [93]. The objective of this phase is to be relaxed before the driving test and to stabilize the sensors. A total of 50 drivers with an average age of 31.76 years (max: 57, min: 18; std. dev.: 10.48) and driving experience of 11.28 years (max 40, min: 1, std. dev: 10.24) participated in the experiment. The participants drove for 25 min. In the driving test, the heart signal, the skin conductivity and the environment (temperature, humidity and CO₂ level) were monitored. The music and the sounds of the vehicle were listened to through headphones. The drivers had to complete the routes proposed by the GPS of the driving simulator. Each route has a length of 5 km and its level of difficulty is comparable because the concentration of vehicles and pedestrians is the same in all cases. The driving simulator assigns points to the participant at the beginning of the route. Each time an infraction is committed, points are deducted. When the score is zero, the route must be repeated. This allows the participant to be focused on the driving task as if in a real environment [94]. The driving time is 25 min to have enough time for the stress data to be valid [95].

A statistical analysis was conducted using R (version 3.6.0) in order to obtain conclusions from the data. We have used the Student’s test or Wilcoxon’s test for independent samples depending on whether the hypothesis of normality is verified or not. The significance level was set at 0.05. Therefore, if the *p* value is less than 0.05, we assume that there are significant differences between the analyzed groups.

4. Results

4.1. Effects of Initial Stress

The drivers have been grouped into two sets, “stressed” and “non-stressed”, according to the initial level of stress. The drivers indicated their stress level using a Likert scale with values between 1 and 5, where 1 means that they are not suffering from stress and 5 that they have a lot of stress. The “stressed” group is made up of 21 drivers. These indicated in the initial survey that their stress level was equal to or higher than 4. The “non-stressed” group consists of 29 drivers. These drivers indicated a stress level equal to or less than 3. In order to analyze if there are significant differences between the two groups, we conducted a Student’s test or a Wilcoxon’s test for independent samples, depending on whether or not the normality hypothesis is verified. We use $p < 0.05$ as the significance level.

Table 6 shows the variables related to stress during driving. The participants who initially indicated that they had stress also obtained values associated with high stress during the driving test. We have found significant differences in two of the three variables analyzed. The result of Wilcoxon’s test is $Z = -3.116$, $p < 0.05$ for pNN50, $Z = -3.803$, $p < 0.05$ for LF/HF ratio and $Z = -3.491$, $p < 0.05$ for SCR amplitude.

Table 6. Heart rate variability and skin conductivity during the driving test grouped by initial stress level.

| | | Stressed | Non-Stressed | <i>p</i> Value |
|---------------|----------------|--------------|--------------|----------------|
| pNN50 | Average Value | 5.08% | 17.95% | 0.002 |
| | Median Value | 2.90% | 10.88% | |
| | Std. Deviation | 7.03% | 19.96% | |
| | P25 | 1.07% | 3.20% | |
| | P75 | 5.52% | 23.25% | |
| LF/HF | Average Value | 6.83 | 3.61 | <0.001 |
| | Median Value | 6.28 | 2.83 | |
| | Std. Deviation | 2.64 | 2.80 | |
| | P25 | 4.95 | 1.22 | |
| | P75 | 8.86 | 5.93 | |
| SCR Amplitude | Average Value | 0.55 μ S | 0.25 μ S | <0.001 |
| | Median Value | 0.48 μ S | 0.11 μ S | |
| | Std. Deviation | 0.35 μ S | 0.31 μ S | |
| | P25 | 0.33 μ S | 0.08 μ S | |
| | P75 | 0.71 μ S | 0.26 μ S | |

Stress also has consequences on driving behavior. Stressed drivers accelerate and brake more frequently and intensively than other drivers, as can be seen in Table 7. The difference in driving behavior is especially important in harsh accelerations and decelerations. The percentage of sudden accelerations is six times higher compared to unstressed drivers and twice as high in the case of sudden braking. The result of Wilcoxon’s test is $Z = -5.376$, $p < 0.05$ for harsh braking, $Z = -2.428$, $p < 0.05$ for braking time and $Z = -5.063$, $p < 0.05$ for harsh acceleration. In the case of the acceleration time, neither the normality hypothesis nor the equality hypothesis of variances are rejected. Therefore, we carry out a Student’s test whose result is $t(48) = 2.703$, $p < 0.05$.

Figure 2 shows the degree of compliance with traffic rules grouped by initial stress level. We have found significant differences between stressed drivers and non-stressed drivers in “Speed limit exceeded” ($Z = -5.184$, $p < 0.05$), “Do not yield to a pedestrian in a crosswalk” ($Z = -2.695$, $p < 0.05$) and “Crossing the lane markings illegally” ($Z = -2.588$, $p < 0.05$). Drivers who are initially stressed often drive at high speed, invade the opposite lane to overtake other vehicles and do not stop at crosswalks.

Table 7. Driving behavior grouped by initial stress level.

| | | Stressed | Non-Stressed | p Value |
|---------------------------|----------------|----------|--------------|---------|
| Harsh braking | Average Value | 24.43% | 9.12% | <0.001 |
| | Median Value | 25.25% | 7.47% | |
| | Std. Deviation | 6.38% | 6.20% | |
| | P25 | 19.36% | 4.95% | |
| | P75 | 28.80% | 10.10% | |
| Braking time | Average Value | 24.55% | 17.86% | 0.015 |
| | Median Value | 24.05% | 17.11% | |
| | Std. Deviation | 6.77% | 9.27% | |
| | P25 | 21.94% | 9.05% | |
| | P75 | 29.26% | 25.35% | |
| Harsh Acceleration | Average Value | 8.16% | 1.37% | <0.001 |
| | Median Value | 6.48% | 0.48% | |
| | Std. Deviation | 4.83% | 1.87% | |
| | P25 | 6.14% | 0.15% | |
| | P75 | 12.58% | 1.65% | |
| Acceleration time | Average Value | 67.01% | 61.018% | 0.009 |
| | Median Value | 68.67% | 61.34% | |
| | Std. Deviation | 7.69% | 7.40% | |
| | P25 | 60.43% | 55.11% | |
| | P75 | 72.94% | 66.02% | |

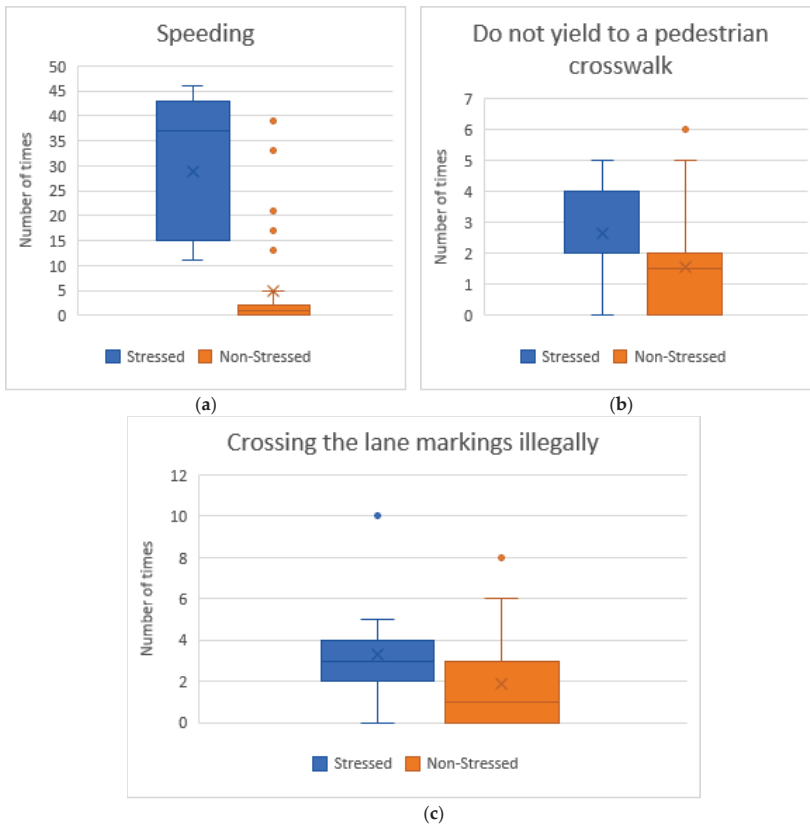


Figure 2. Average number of traffic rules broken grouped by initial stress level: (a) Speeding; (b) Do not yield to a pedestrian at a crosswalk; and (c) Crossing the lane markings illegally.

Figure 3 compares the difference between initial and final fatigue for the two groups of drivers. These values were obtained from the pre-test and post-test surveys. On the one hand, stressed drivers suffer an important increase in the fatigue level after completing the driving test. The tiredness grew by 20%. In the case of drivers with low initial stress, the level of tiredness scarcely changed. On the other hand, at the beginning of the driving experiment, we found no significant differences in the fatigue level between the two groups of drivers analyzed. The result of Wilcoxon's test is $Z = -1.491$, $p > 0.05$. However, we observed significant differences at the end of the experiment. The result of Wilcoxon's test is $Z = -4.545$, $p < 0.05$.

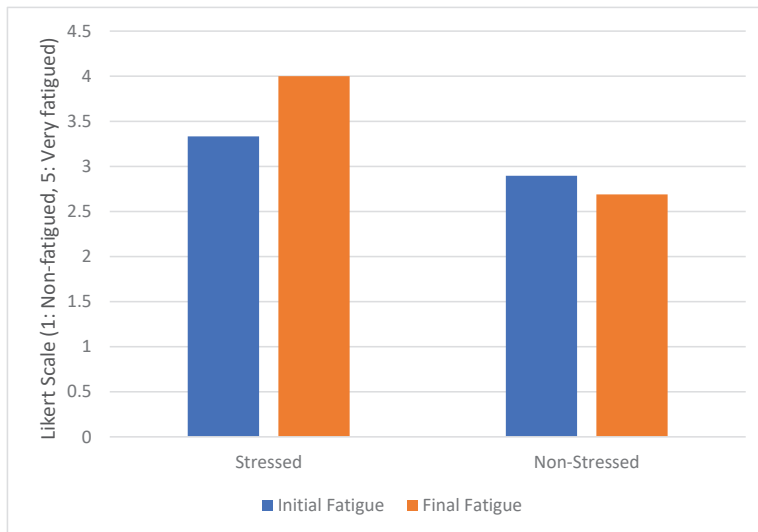


Figure 3. Fatigue evolution grouped by initial stress level.

4.2. Effects of Sadness

The drivers have been grouped into two sets according to the sadness level. The drivers indicated their sadness level using a Likert scale with values between 1 and 5, where 1 means that they are very happy and 5 indicates that they are very sad. The group of drivers with sadness is composed of 17 drivers. These indicated in the initial survey that their sadness level was equal to or higher than 4. The non-sadness group is formed by 33 drivers who rated their level of unhappiness with a value equal to or less than 3.

Table 8 shows the variables related to stress during driving. Drivers who show sadness are also those who have a higher level of stress. However, the differences are not significant. The result of Wilcoxon's test is $Z = -0.881$, $p > 0.05$ for pNN50, $Z = -0.522$, $p > 0.05$ for LF/HF and $Z = -0.420$, $p > 0.05$ for SCR amplitude.

Table 9 presents the acceleration and deceleration values obtained by the drivers. We observe that drivers with sadness accelerate sharply more times than drivers without sadness, although the difference is not significant. The result of the Student's test is $t(48) = 2.001$, $p > 0.05$. No significant differences were found either in the rest of the parameters.

Table 8. Heart rate variability and skin conductivity during the driving test grouped by sadness level.

| | | Sadness | Non-Sadness | <i>p</i> Value |
|----------------------|----------------|--------------|--------------|----------------|
| pNN50 | Average Value | 10.90% | 13.39% | 0.384 |
| | Median Value | 3.94% | 6.66% | |
| | Std. Deviation | 17.94% | 16.72% | |
| | P25 | 1.12% | 2.61% | |
| | P75 | 11.58% | 17.08% | |
| LF/HF | Average Value | 4.68 | 5.11 | 0.647 |
| | Median Value | 4.40 | 4.67 | |
| | Std. Deviation | 3.10 | 3.21 | |
| | P25 | 2.10 | 2.65 | |
| | P75 | 7.19 | 7.68 | |
| SCR Amplitude | Average Value | 0.45 μ S | 0.34 μ S | 0.682 |
| | Median Value | 0.26 μ S | 0.26 μ S | |
| | Std. Deviation | 0.41 μ S | 0.32 μ S | |
| | P25 | 0.12 μ S | 0.10 μ S | |
| | P75 | 0.89 μ S | 0.48 μ S | |

Table 9. Driving behavior grouped by sadness level.

| | | Sadness | Non-Sadness | <i>p</i> Value |
|---------------------------|----------------|---------|-------------|----------------|
| Harsh braking | Average Value | 16.36 | 15.23 | 0.757 |
| | Median Value | 16.69 | 14.39 | |
| | Std. Deviation | 11.03 | 9.34 | |
| | P25 | 7.47 | 5.89 | |
| | P75 | 26.69 | 24.03 | |
| Braking time | Average Value | 20.73 | 20.64 | 0.935 |
| | Median Value | 23.74 | 21.05 | |
| | Std. Deviation | 8.29 | 9.30 | |
| | P25 | 14.47 | 11.45 | |
| | P75 | 26.40 | 29.26 | |
| Harsh Acceleration | Average Value | 6.06 | 3.28 | 0.051 |
| | Median Value | 5.51 | 1.02 | |
| | Std. Deviation | 5.36 | 4.26 | |
| | P25 | 1.09 | 0.33 | |
| | P75 | 9.06 | 6.18 | |
| Acceleration time | Average Value | 64.55 | 63.16 | 0.565 |
| | Median Value | 61.04 | 63.15 | |
| | Std. Deviation | 8.35 | 7.90 | |
| | P25 | 57.45 | 59.45 | |
| | P75 | 70.91 | 68.67 | |

Figure 4 captures the average number of traffic accidents. Drivers with sadness suffer traffic accidents more often than the group of drivers without sadness. The difference between the two groups is especially relevant, as the group with sadness is involved in four times as many accidents as the group of drivers without sadness. The result of Wilcoxon's test is $Z = -4.741$, $p < 0.05$.

Figure 5 compares the level of fatigue before and after the driving test. Drivers suffering from sadness increased their fatigue level by 11.5% compared to their initial value. In the case of drivers without sadness, fatigue increased by 7.5%. However, we found no significant differences between both groups at the beginning and at the end of the experiment. On the one hand, the result of Wilcoxon's test in the initial survey is $Z = -0.361$, $p > 0.05$. On the other hand, the result of Wilcoxon's test in the post-experimental survey is $Z = -1.472$, $p > 0.05$.

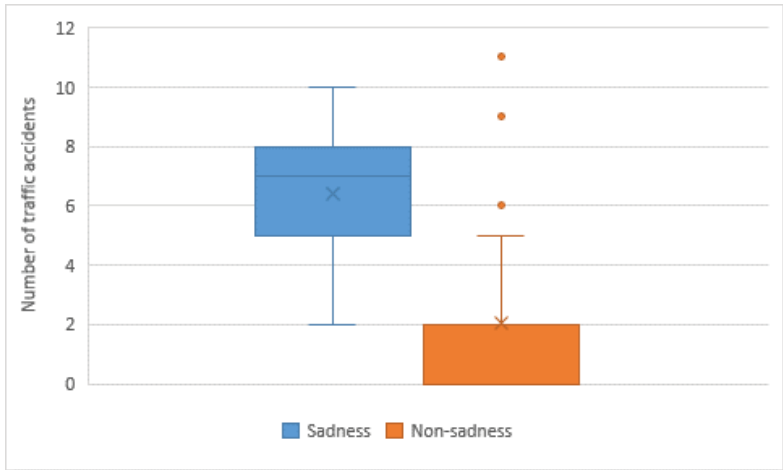


Figure 4. Number of traffic accidents grouped by sadness level.

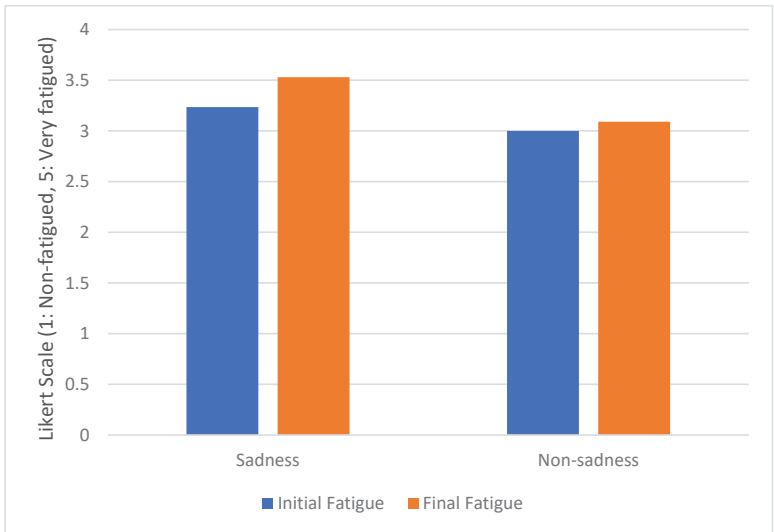


Figure 5. Fatigue evolution grouped by sadness level.

4.3. Effects of Fatigue

In order to analyze this factor, we have divided the samples into two groups. The drivers indicated their initial fatigue level using a Likert scale with values between 1 and 5, where 1 means that they are very vigorous and 5 indicates that they are very tired. The non-fatigue group consists of 36 drivers. These drivers showed a tiredness level equal to or less than 3. The fatigue group is made up of 14 drivers. These indicated in the initial survey that their fatigue level was equal to or higher than 4.

Table 10 reveals the average stress level during the test grouped by the initial fatigue level. The results indicate that tired drivers suffer more stress while driving than the other drivers. The variable pNN50 is eight times lower in the group of drivers who are tired, and the LF/HF ratio and SCR amplitude are twice as high. Low values of pNN50 and high values of LF/HF ratio and SCR amplitude

are correlated with high stress. In all variables, the differences are significant. The result of Wilcoxon's test is $Z = -4.905, p < 0.05$ for pNN50, $Z = -4.127, p < 0.05$ for LF/HF and $Z = -3.297$ for SCR.

Table 10. Heart rate variability and skin conductivity during driving test grouped by tiredness level.

| | | Fatigue | Non-Fatigue | <i>p</i> Value |
|----------------------|----------------|--------------|--------------|----------------|
| pNN50 | Average Value | 1.47% | 16.85% | <0.001 |
| | Median Value | 1.10% | 9.45% | |
| | Std. Deviation | 1.45% | 18.32% | |
| | P25 | 0.60% | 3.89% | |
| | P75 | 2.15% | 21.66% | |
| LF/HF | Average Value | 7.96 | 3.80 | <0.001 |
| | Median Value | 7.48 | 3.08 | |
| | Std. Deviation | 1.82 | 2.76 | |
| | P25 | 6.83 | 1.77 | |
| | P75 | 8.50 | 4.96 | |
| SCR Amplitude | Average Value | 0.62 μ S | 0.28 μ S | 0.001 |
| | Median Value | 0.61 μ S | 0.16 μ S | |
| | Std. Deviation | 0.35 μ S | 0.31 μ S | |
| | P25 | 0.28 μ S | 0.08 μ S | |
| | P75 | 1.00 μ S | 0.44 μ S | |

Driving behavior is also affected by this state. Table 11 shows the use of the accelerator and brake. Acceleration time and braking time is higher for tired drivers than for rested drivers. The differences are significant. The result of the Student's test is $t(48) = 2.905, p < 0.05$ for acceleration time and $t(48) = 3.754, p < 0.05$ for braking time. This means that the drivers are continuously making speed corrections and increasing fuel consumption. No significant differences have been found in the case of abrupt maneuvers, although both average and median values are higher for tired drivers.

Table 11. Driving behavior grouped by level of tiredness.

| | | Fatigue | Non-Fatigue | <i>p</i> Value |
|---------------------------|----------------|---------|-------------|----------------|
| Harsh braking | Average Value | 16.16% | 15.23% | 0.757 |
| | Median Value | 16.69% | 14.39% | |
| | Std. Deviation | 11.03% | 9.34% | |
| | P25 | 7.47% | 5.89% | |
| | P75 | 26.69% | 24.03% | |
| Braking time | Average Value | 27.39% | 18.06% | <0.001 |
| | Median Value | 26.50% | 17.37% | |
| | Std. Deviation | 5.33% | 8.65% | |
| | P25 | 24.07% | 10.09% | |
| | P75 | 29.51% | 24.29% | |
| Harsh Acceleration | Average Value | 5.74% | 3.64% | 0.166 |
| | Median Value | 5.83% | 1.19% | |
| | Std. Deviation | 5.24% | 4.55% | |
| | P25 | 0.81% | 0.36% | |
| | P75 | 6.85% | 6.31% | |
| Acceleration time | Average Value | 68.54% | 61.72% | 0.006 |
| | Median Value | 68.83% | 60.96% | |
| | Std. Deviation | 7.36% | 7.48% | |
| | P25 | 62.79% | 55.37% | |
| | P75 | 75.22% | 67.93% | |

Figure 6 captures the number of broken driving rules in which there are significant differences between fatigued and non-fatigued drivers. The result of Wilcoxon's test is $Z = -4.402, p < 0.05$ for "Stopping over the crosswalk" and $Z = -3.459, p < 0.05$ for "Do not yield to a pedestrian at a crosswalk". Tired drivers stop over the crosswalk 4.5 more times more than the rest of the drivers. Furthermore,

they did not yield to a pedestrian at a crosswalk two times more. This could increase the likelihood of running over a pedestrian.

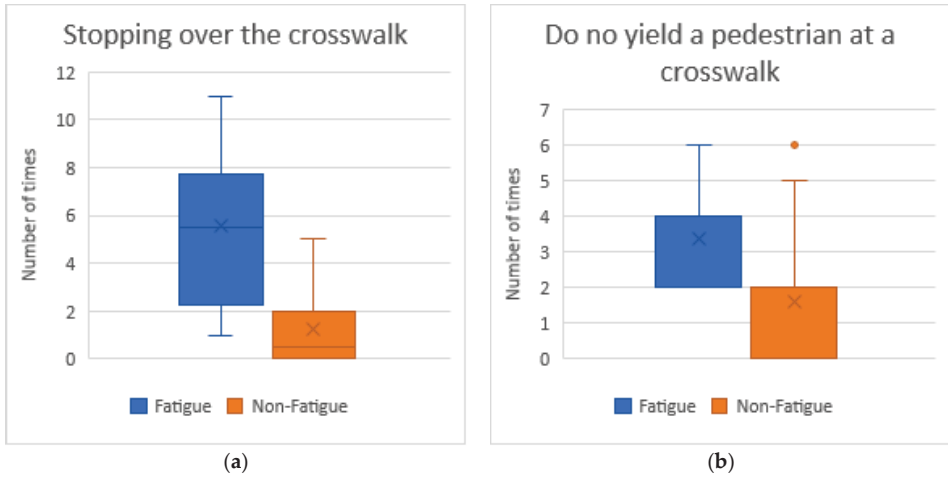


Figure 6. Average number of traffic rules broken grouped by tiredness level: (a) Stopping over the crosswalk; and (b) Do not yield to a pedestrian at a crosswalk.

Figure 7 compares the difference between initial and final fatigue for the two groups of drivers. On the one hand, there is a significant increase in the fatigue level of the non-tired drivers. Tiredness increased by 13.79% after completing the driving test. On the other hand, in the case of tired drivers, the average fatigue value decreases by 8.39%. This could be because for some participants, the driving test is like a leisure activity. Despite this, the level of fatigue manifested by the drivers who were initially tired remains significantly higher than that of the drivers who initially did not feel tired. The result of Wilcoxon’s test is $Z = -3.105, p < 0.05$.

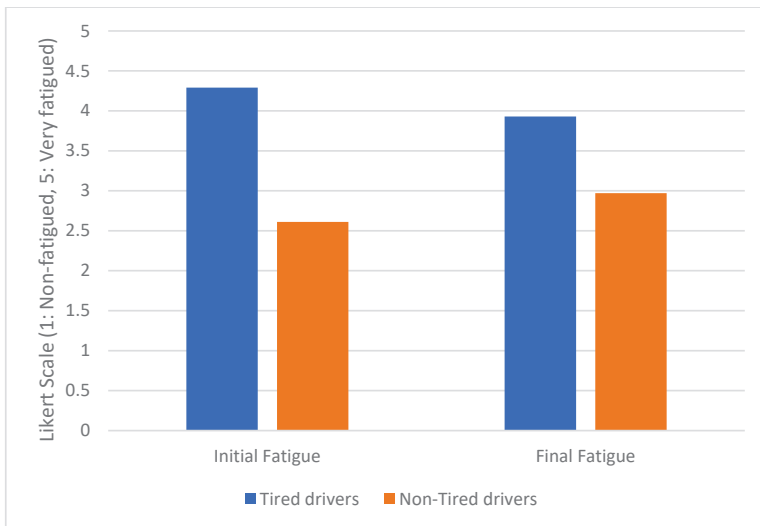


Figure 7. Fatigue evolution.

4.4. Effects of CO₂ Concentration

In order to analyze this factor, we have divided the samples into two groups. One group consists of 29 drivers who drove with an average CO₂ value of less than 1400 ppm. We have chosen this threshold because it has been shown in many articles [11] that differences in cognitive performance appear above this value. The average value of CO₂ concentration of this group was 319.67 ppm (max: 562.55 ppm, min: 149.8 ppm, std. dev: 119.17 ppm). The second group is made up of 21 drivers who drove with an average CO₂ value equal to or higher than 1400 ppm. The average value of CO₂ concentration of this group was 1572.96 ppm (max: 1734.56 ppm, min: 1434.81 ppm, std. dev: 107.43 ppm). The average temperature value during all the tests was 25.27 °C (maximum = 26.71 °C, minimum = 24.12 °C, standard deviation = 0.63 °C) and the average humidity was 50.64% (maximum = 58.13%, minimum = 48.35%, standard deviation = 3.11%).

Table 12 captures the value of the variables associated with stress. The difference between groups is not significant. The result of Wilcoxon's test is $Z = -0.147, p > 0.05$ for pNN50, $Z = -0.88, p > 0.05$ for LF/HF and $Z = -0.364, p > 0.05$ for SCR amplitude.

Table 12. Heart rate variability and skin conductivity during driving test grouped by CO₂ level.

| | | CO ₂ ≥ 1400 ppm | CO ₂ < 1400 ppm | <i>p</i> Value |
|----------------------|----------------|----------------------------|----------------------------|----------------|
| pNN50 | Average Value | 15.21% | 10.61% | 0.891 |
| | Median Value | 3.94% | 5.57% | |
| | Std. Deviation | 20.11% | 14.41% | |
| | P25 | 1.75% | 2.61% | |
| | P75 | 28.31% | 10.93% | |
| LF/HF | Average Value | 5.01 | 4.94 | 0.938 |
| | Median Value | 4.67 | 3.53 | |
| | Std. Deviation | 3.32 | 3.08 | |
| | P25 | 2.65 | 2.49 | |
| | P75 | 6.76 | 7.28 | |
| SCR Amplitude | Average Value | 0.39 μS | 0.37 μS | 0.723 |
| | Median Value | 0.29 μS | 0.20 μS | |
| | Std. Deviation | 0.34 μS | 0.37 μS | |
| | P25 | 15.21% | 10.61% | |
| | P75 | 3.94% | 5.57% | |

Table 13 shows driving behavior. The results indicate that the driver brakes more frequently when the passenger compartment has a high concentration of CO₂. The difference is significant between the two groups (high and low CO₂ level). The result of Wilcoxon's test was $Z = -3.843, p < 0.05$ for braking time. A high CO₂ concentration causes drowsiness and a lack of concentration. The participant's cognitive capacity is reduced and he or she responds more slowly to events that happen on the road.

Consequently, as we can see in Figure 8, the driver violates more traffic regulations and is involved in a higher number of traffic accidents. We found significant differences in "Crossing the lane markings illegally" ($Z = -2.478, p < 0.05$), "Not stopping at a red light" ($Z = -2.752, p < 0.05$) and "Traffic accidents" ($Z = -2.105, p < 0.05$). Figure 8 captures the traffic rules broken and traffic accidents grouped by CO₂ level. We can see how drivers who are exposed to high concentrations of CO₂ invade the opposite lane 95% more than the rest of the drivers. Moreover, they respect traffic lights less. The group of drivers who drive with a high concentration of CO₂ ignored the red lights 1.14 times on average, while the drivers who drive with a low CO₂ level passed red lights 0.59 times. Frequent decelerations along with non-compliance with traffic regulations result in a sharp increase in the number of accidents of the group with the high CO₂ concentration. These drivers suffer 1.87 times more accidents than the rest of the drivers.

Table 13. Driving behavior grouped by CO₂ level.

| | | CO ₂ ≥ 1400 ppm | CO ₂ < 1400 ppm | p Value |
|--------------------|----------------|----------------------------|----------------------------|---------|
| Harsh braking | Average Value | 18.66% | 13.30% | 0.070 |
| | Median Value | 17.45% | 9.57% | |
| | Std. Deviation | 10.25% | 9.05% | |
| | P25 | 9.15% | 5.80% | |
| | P75 | 26.69% | 20.58% | |
| Braking time | Average Value | 26.40% | 16.52% | <0.001 |
| | Median Value | 27.47% | 14.90% | |
| | Std. Deviation | 7.69% | 7.31% | |
| | P25 | 22.74% | 9.87% | |
| | P75 | 31.24% | 23.93% | |
| Harsh Acceleration | Average Value | 3.85% | 4.50% | 0.930 |
| | Median Value | 4.24% | 1.65% | |
| | Std. Deviation | 3.83% | 5.44% | |
| | P25 | 0.48% | 0.33% | |
| | P75 | 6.54% | 7.26% | |
| Acceleration time | Average Value | 63.53% | 63.70% | 0.943 |
| | Median Value | 61.04% | 63.15% | |
| | Std. Deviation | 7.87% | 8.22% | |
| | P25 | 59.22% | 57.45% | |
| | P75 | 70.36% | 69.43% | |

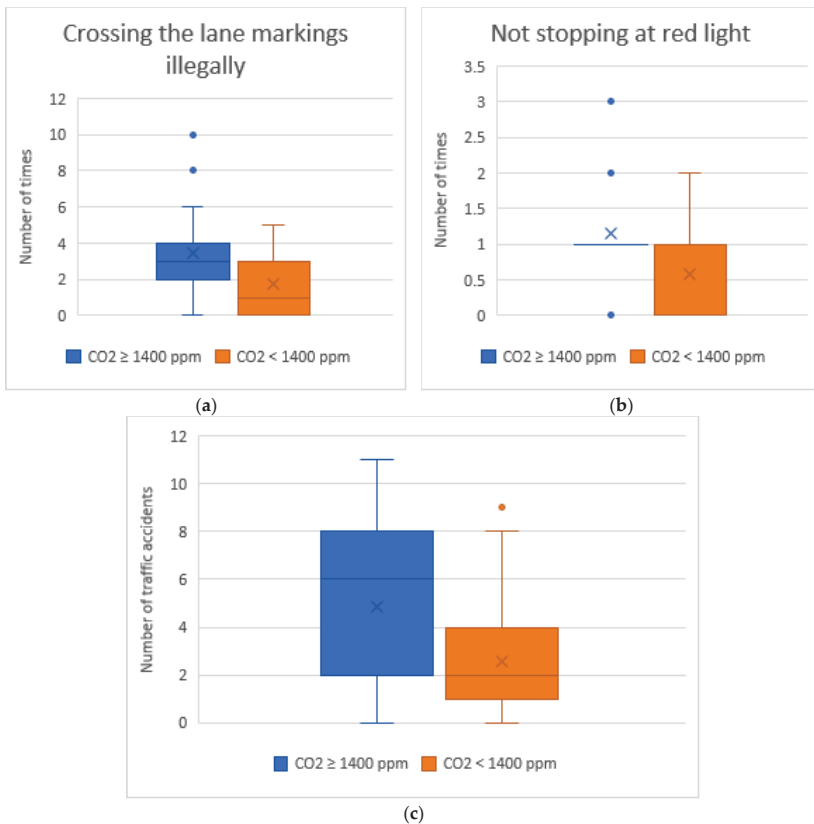


Figure 8. (a) Number of times drivers cross the lane markings illegally; (b) number of times that drivers do not stop at a red light; and (c) number of traffic accidents.

Figure 9 captures the initial and final level of fatigue for each group of drivers. The results show that when the CO₂ concentration is high, the fatigue level increases by 12% compared to the initial value, while when the CO₂ level is low, the level of fatigue does not change.

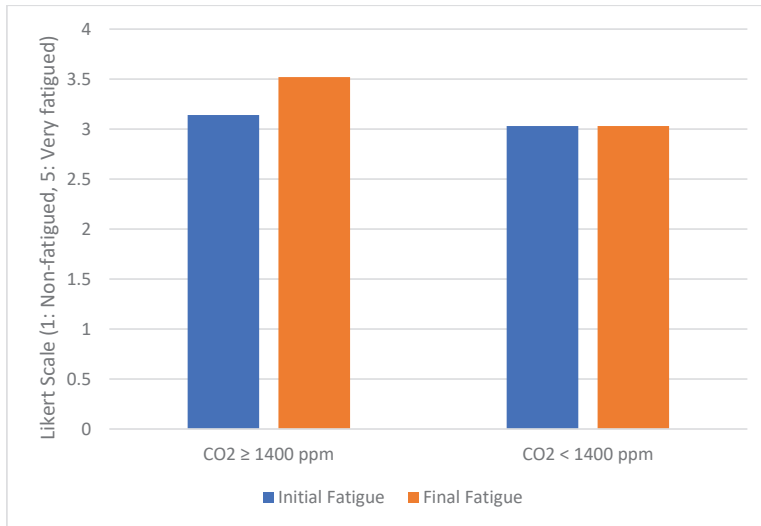


Figure 9. Evolution of fatigue grouped by CO₂ level.

4.5. Effects of Music Tempo

As in the previous analyses, the driving samples were divided into two groups. One group is made up of 23 drivers who listened to slow tempo music. The other group consists of 27 participants, but in this case the music was fast tempo music.

Table 14 captures the value of the variables related to stress. In the case of drivers listening to fast-paced music, pnn50 and LF/HF ratio are higher than drivers listening to slow music, although the differences are not significant. The result of Wilcoxon's test is $Z = -0.049$, $p > 0.05$ for pNN50, $Z = -0.457$, $p > 0.05$ for LF/HF and $Z = -0.886$, $p > 0.05$ for SCR amplitude.

Table 14. Heart rate variability and skin conductivity during the driving test grouped by music tempo.

| | | Slow Tempo | Fast Tempo | <i>p</i> Value |
|---------------|----------------|--------------|--------------|----------------|
| pNN50 | Average Value | 12.30% | 12.76% | 0.969 |
| | Median Value | 3.94% | 7.10% | |
| | Std. Deviation | 17.20% | 17.15% | |
| | P25 | 2.41% | 1.44% | |
| | P75 | 12.72% | 16.80% | |
| LF/HF | Average Value | 4.65 | 5.23 | 0.657 |
| | Median Value | 4.95 | 3.53 | |
| | Std. Deviation | 2.32 | 3.73 | |
| | P25 | 3.01 | 1.95 | |
| | P75 | 6.26 | 8.74 | |
| SCR Amplitude | Average Value | 0.40 μ S | 0.36 μ S | 0.381 |
| | Median Value | 0.35 μ S | 0.25 μ S | |
| | Std. Deviation | 0.33 μ S | 0.38 μ S | |
| | P25 | 0.14 μ S | 0.08 μ S | |
| | P75 | 0.44 μ S | 0.44 μ S | |

Table 15 captures driving behavior grouped by music tempo. We observed that the average values of the four variables analyzed are higher in the case of drivers who listen to fast-paced music than participants who listen to slow-paced music. This means that drivers with fast-paced music show a more aggressive driving style, although we have only found significant differences in acceleration time. The result of the Student’s test is $t(48) = -2.891, p < 0.05$. Likewise, we have found significant differences in the violation of speed limits, as can be seen in Figure 10. The result of Wilcoxon’s test is $Z = -1.980, p < 0.05$. As future work, we want to conduct more experiments to verify whether the differences between the driving behavior variables are significant if the number of participants is increased.

Table 15. Driving behavior grouped by music tempo.

| | | Slow Tempo | Fast Tempo | <i>p</i> Value |
|--------------------|----------------|------------|------------|----------------|
| Harsh braking | Average Value | 14.48% | 16.45% | 0.428 |
| | Median Value | 15.51% | 12.99% | |
| | Std. Deviation | 8.81% | 10.72% | |
| | P25 | 6.03% | 7.05% | |
| | P75 | 19.33% | 26.57% | |
| Braking time | Average Value | 20.04% | 21.21% | 0.649 |
| | Median Value | 21.94% | 22.74% | |
| | Std. Deviation | 9.38% | 8.58% | |
| | P25 | 10.75% | 14.68% | |
| | P75 | 26.50% | 29.45% | |
| Harsh Acceleration | Average Value | 3.56% | 4.79% | 0.876 |
| | Median Value | 1.65% | 2.03% | |
| | Std. Deviation | 4.02% | 5.38% | |
| | P25 | 0.55% | 0.20% | |
| | P75 | 5.37% | 7.10% | |
| Acceleration time | Average Value | 60.33% | 66.44% | 0.006 |
| | Median Value | 59.56% | 68.05% | |
| | Std. Deviation | 7.70% | 7.24% | |
| | P25 | 57.04% | 62.70% | |
| | P75 | 61.75% | 70.86% | |

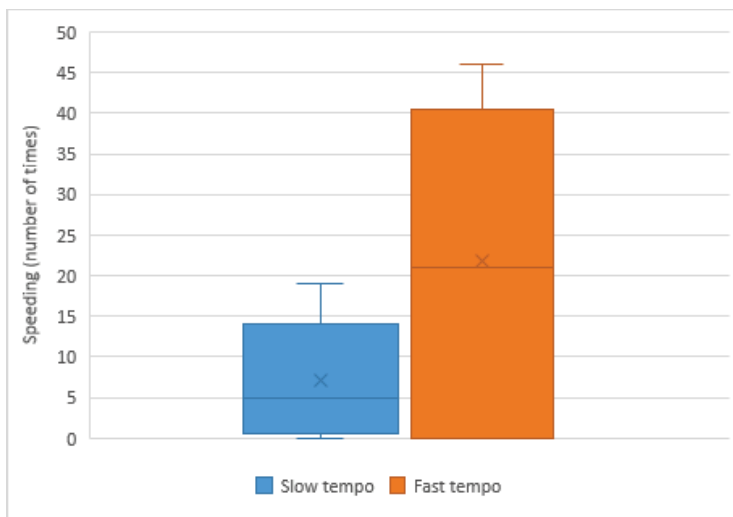


Figure 10. Number of times the driver exceeds the speed limit.

Figure 11 shows the level of initial and final fatigue for the two groups of drivers using a Likert scale, where 1 means no fatigue and 5 a lot of fatigue. The level of fatigue only increased by 2.8% for drivers who listened to music at a slow pace. In contrast, drivers who listened to fast-paced music suffered a significant increase in the level of fatigue (by 7.5%). These results are consistent with those obtained by [95]. In this study, fast music deteriorated the level of fatigue.

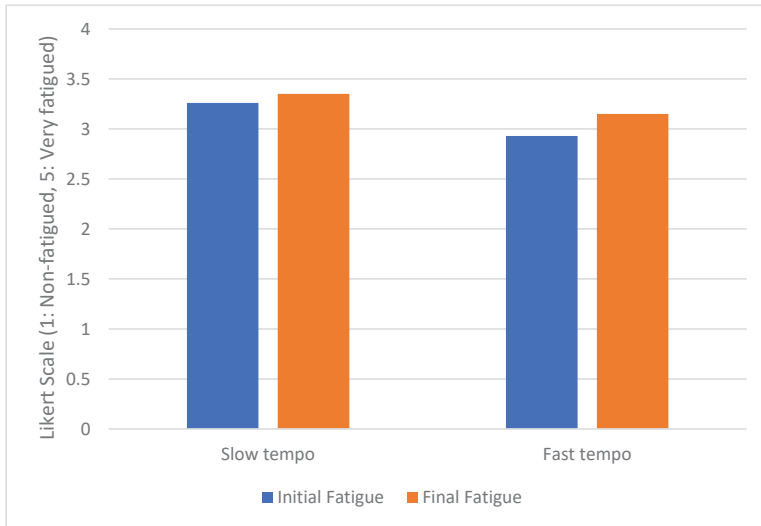


Figure 11. Fatigue evolution grouped by music tempo.

4.6. Multivariate Analysis

Linear ANOVA models have been calculated for each of the factors analyzed: initial stress, sadness, initial fatigue, CO₂ concentration and music tempo. In all models, the p value is less than 0.05. Therefore, we can state that the independent variables reliably predict the dependent variable. Table 16 shows the models with an adjusted R-squared (R^2) higher than 55%. Adjusted R-squared is a statistic that gives information about the goodness of fit of a model. R-squared is defined as the fraction of the variance in the dependent variable that is explained by the model. The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The higher the adjusted R-squared value, the more the model fits the real data. In Table 16, the labels labeled as “COEFFICIENT” are the values for the regression equation for predicting the dependent variable from the independent variables. Finally, the p value is a probability. It gauges the likelihood that the coefficient is not significant, so smaller is better. In our case, we consider that there is significance when the value is less than 0.05.

We can see that the best model is obtained in “Speeding”, where the adjusted R^2 is higher than 70%. In view of the coefficient’s values and the p value, we can state that the initial stress level as well as the initial fatigue and fast-paced music significantly increase the number of times the speed limits are surpassed. We can also point out that both sadness and a high concentration of CO₂ do not seem to influence speeding. In these two independent variables, the p values are higher than 0.05.

Table 16. Results of multivariate analysis with high R^2 .

| | Factor | Coefficient | p Value | R^2 |
|--------------------|------------------------------------|-------------|-----------|--------|
| LF/HF | High Initial Stress | 2.744 | <0.001 | 56.69% |
| | Sadness | -1.405 | 0.032 | |
| | Tiredness | 4.152 | <0.001 | |
| | High CO ₂ Concentration | -0.672 | 0.273 | |
| | Fast Music | 1.138 | 0.062 | |
| HARSH ACCELERATION | High Initial Stress | 6.936 | <0.001 | 57.56% |
| | Sadness | 2.315 | 0.020 | |
| | Tiredness | 0.586 | 0.574 | |
| | High CO ₂ Concentration | -2.074 | 0.028 | |
| | Fast Music | 1.469 | 0.108 | |
| HARSH BRAKING | High Initial Stress | 14.234 | <0.001 | 63.79% |
| | Sadness | -1.215 | 0.506 | |
| | Tiredness | 4.509 | 0.026 | |
| | High CO ₂ Concentration | 2.567 | 0.145 | |
| | Fast Music | 2.978 | 0.086 | |
| SPEEDING | High Initial Stress | 23.641 | <0.001 | 71.72% |
| | Sadness | -3.884 | 0.162 | |
| | Tiredness | 6.209 | 0.042 | |
| | High CO ₂ Concentration | -0.538 | 0.838 | |
| | Fast Music | 16.003 | <0.001 | |

In the LF/HF ratio, we found that the initial stress level along with fatigue contributes to the occurrence of stress during driving. It is important to highlight the strong relationship between initial fatigue and the possibility of stress while driving, where the coefficient value is 4.152. In the case of the other two variables related to driving stress (pNN50 and SCR amplitude), the same thing happens, but we have not included them in the table because the adjusted R-squared value is lower than 50%. Finally, we can observe that when the driver suffers sadness, the value of the LF/HF ratio decreases, meaning less stress in driving. The p value for sadness is lower than 0.05 and the coefficient is -1.405. This could be explained because the drivers are focused on their own problems. Extremely low driving stress is also not good for safety because it could cause drowsiness [5].

The results of the “Harsh braking” variable are very similar to the “LF/HF” variable. However, the p value of sadness is higher than 0.05. Therefore, in this context, it does not significantly affect the model. The driver who is initially tired or stressed does not react early enough to road events, forcing aggressive maneuvers and increasing driving stress.

Regarding sudden accelerations, they characterize an aggressive driving style which appears especially when the driver is stressed. The coefficient value is 6.936 and the p value is lower than 0.05. Sadness is also an emotion that contributes. The coefficient value is 2.315 and the p value is lower than 0.05. People with sadness often adopt an aggressive driving style and a certain degree of passiveness that causes increased fuel consumption and can annoy other drivers [22]. On the contrary, a high concentration of CO₂ decreases harsh accelerations. The coefficient value is -2.074 and the p value is lower than 0.05. This could be due to the possible appearance of drowsiness [96].

5. Discussion and Limitations of Our Experiment

In our experiment, the initial level of stress and fatigue has a strong impact on driving behavior and driving stress. The relationship between stress and road safety has been verified by many authors [97,98]. Several studies have corroborated that a high level of stress increases errors and traffic violations. In [46], the authors conducted a study involving 2806 drivers using the driver behavior questionnaire (DBQ) and the driver behavior inventory (DBI). The DBI assesses dimensions of driver stress, whereas the DBQ is concerned with assessing the relative frequencies with which drivers engage in different types of aberrant driving behavior. They found a strong correlation between an aggressive driving style and high levels of stress. They also observed that when the stress is high, drivers make

more mistakes, although in this case, the dislike of driving also seems to play a role. This is consistent with our findings that stressed drivers accelerate and brake more often than non-stressed drivers. Furthermore, harsh accelerations are six times higher than the values obtained by non-stressed drivers. In the case of harsh braking, the values are twice as high as those obtained by non-stressed drivers. Harsh accelerations and harsh braking are indicative of an aggressive driving style. The main difference between our analysis and the previous literature is that we have monitored the driver's state and driving behavior. Most of the proposals are based on self-reports of drivers or traffic accident databases provided by the government [99]. The problem with self-reports is that they depend on the drivers' perception, which could be wrong. In [99], the authors found that drivers with high confinement had a low risk perception and reported driving errors incorrectly.

Regarding the sadness factor, we observe that it is mainly characterized by a very significant increase in the number of traffic accidents. This emotion also contributes significantly to the increase in sudden decelerations. Attentional self-focus and repetitive negative thoughts are two main elements in sadness [100,101]. These elements affect information processing and attention [102]. In [103], the authors observed that sadness-induced drivers made more errors in target location. This could explain why, in our experiment, drivers with sadness suffered more traffic accidents than drivers who do not feel this emotion. On the other hand, we also found in our driving test that drivers with sadness did not manifest more stress than other drivers. In [22], the researchers conducted a simulated driving experiment with two induced affective states to examine how sadness and anger differently influence driving-related risk perception, driving performance and perceived workload. The results they obtained showed that sad drivers make more driving errors, but do not perceive a higher workload than drivers with an emotionally neutral state. This could explain why we have not found significant differences in driving stress.

In the literature, many researchers focus on analyzing how fatigue that increases during driving affects driving performance and road safety [104]. These studies point out that fatigue is a very important factor that causes a lack of hazard perception [105]. This may lead to driving accidents [106]. In this regard, the European Union has a regulation that sets the maximum driving time for professional drivers [107]. The relationship between driver fatigue and hours of service regulations is a challenge [108]. Some authors have found that driving time is a significant predictor of accident risk [109]. In other studies, there is no evidence of a time-on-task effect [110]. This could be due to the repercussion of the driver's initial fatigue level. In our study, we have observed that initial fatigue significantly influences driving behavior and driving stress. We have also observed a non-compliance with traffic regulations that require high attention from the subject such as "yield to a pedestrian at a crosswalk". This demonstrates the need to not only monitor fatigue during driving, but also to do so beforehand in order to ensure driving safety.

Traditionally, the CO₂ concentration inside the vehicle cabin was not considered dangerous because of its low level. However, several recent studies have shown that the concentration of CO₂ can be quite high depending on the number of vehicle occupants, speed and the environment [111]. In addition, cognitive impairment has also been observed with low or moderate CO₂ concentrations with short exposure times [112]. In [113], the authors observed that the mental task required more effort from the subjects when the CO₂ concentration in the air reached 3000 ppm. In [12], the researchers concluded that decision-making performance decreased when participants were exposed to CO₂ concentrations between 1000 and 2500 ppm. This is in line with what was observed in our study. The worsening of decision making when the CO₂ concentration is high causes the number of traffic accidents to increase. A high CO₂ concentration also causes fatigue and drowsiness in drivers, reducing reaction time [114]. As a consequence, we observed in our study an increase in the frequency and intensity of decelerations. Finally, the combination of high initial stress with fast-paced music causes, in our experiment, a significant increase in the number of times the maximum allowed speed is exceeded. There are many marketing studies where fast music is used to encourage customers to purchase [115,116]. In the field of driving, many researchers have observed a similar behavior. In [90],

the authors concluded that listening to fast music in the background affects non-compliance with traffic rules such as speeding.

As a limitation in our study, we did not take into account variables such as personality, gender, socio-educational level or the driver's history (fines and traffic accidents). In [117], the researchers conducted a study with 41 drivers using a driving simulator, where they observed that these variables affect driving behavior, especially when drivers are tired. These factors were not included in the survey in order not to extend our experiment and discourage participants. In most of the papers, the subjects only had to fill out surveys and did not drive. Another limitation is in the evaluation of the music factor. We have only analyzed the tempo. The subject could freely adjust the volume of the music and the playlist was the same for all participants. We have not considered other elements that can influence driving behavior such as gender or music familiarity [118].

6. Conclusions

In this work, we have analyzed how the mental state of the driver and the interior state of the vehicle affects driving and its relation to compliance with traffic regulations and accidents.

Among the factors analyzed, the negative influence of stress stands out. On the one hand, stress is strongly related to an aggressive driving style with sudden accelerations and decelerations. This behavior means that the rest of the road users are not able to predict their actions, increasing the probability of traffic accidents. In the driving tests, these drivers did not often respect the speed limits, they overtook other vehicles in areas where this action should not be performed and did not stop at the crosswalks. On the other hand, the driving style associated with this state increases fuel consumption. As the driver drives at an inappropriate speed, the brakes are used more, and the driver does not take advantage of the energy generated by burning the fuel.

Sadness also influences driving behavior. This emotion in combination with stress and listening to fast music increases the number of harsh accelerations, causing problems for both safety and the environment. Drivers suffering from sadness are frequently involved in traffic accidents because they are thinking about their own problems and do not focus on paying attention to the road.

Tiredness is another analyzed factor that has negative consequences. We have observed that tired drivers suffer more stress while driving than non-tired drivers. Tiredness increases response times, and as a result, drivers accelerate and brake more frequently. This could cause a traffic accident because the driver of the vehicle behind only has a short time to react.

Furthermore, we have observed that drivers who listen to music with a fast tempo drive at high speeds, not respecting the limits indicated on the traffic signs. High-speed driving demands more cognitive ability. If the demand is prolonged, it causes an increase in the level of fatigue.

Regarding the interior state of the vehicle, the results obtained when analyzing the data of drivers who were exposed to high concentrations of CO₂ are very similar to those of drivers who were tired. A high concentration of CO₂ causes fatigue and headache, reducing the concentration of the driver on the road. Finally, we want to highlight that we have observed that some drivers who liked video games and were very stressed or tired improved their initial state when doing the driving test. This result could be very useful for developing driving assistants.

In conclusion, this work shows that the driver's behavior not only depends on the driving conditions, but that it is also influenced by the driver's state. Factors such as stress or fatigue can intensify while driving, but the initial values before driving are also very relevant and strongly related to more erratic and dangerous driving. Researchers working on the design of driving assistants could explore whether issuing lifestyle advice improves driving safety and driving efficiency.

As future work, we would like to evaluate how the personality of the driver impacts driving. This, combined with the results obtained in this work, would allow us to develop an advanced driving assistant (ADAS) that fits with the driver profile. An ADAS could intelligently influence the driver's emotions.

Author Contributions: Conceptualization, V.C.M. and W.D.S.; methodology, V.C.M. and W.D.S.; formal analysis, V.C.M.; investigation, all authors; resources, R.S. and X.G.P.; writing—original draft preparation, V.C.M.; writing—review and editing, all authors; supervision, R.F., N.M.M., X.G.P. and R.G.; funding acquisition, X.G.P. and R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Spanish National Research Program under Project TIN2017-82928-R.

Acknowledgments: This publication received technical support from the Statistical Consulting Unit of the Scientific-Technical Services of the University of Oviedo (Spain).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. DGT Report of Traffic Accidents in Spain. Available online: <http://www.dgt.es/Galerias/prensa/2019/07/INFORME.pdf> (accessed on 2 September 2020).
2. Dewar, R.E. Review of Road user behavior and traffic accidents. *Can. Psychol. Rev. Can.* **1977**, *18*, 365. [[CrossRef](#)]
3. Shi, Y. The research of road traffic accidents in Henan Province based on the Human Factors Engineering. In Proceedings of the 2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management, Changchun, China, 3–5 September 2011; pp. 1424–1427.
4. Bellis, E.A.; Page, J. National Motor Vehicle Crash Causation Survey (NMVCCS) SAS Analytical Users Manual. *Security* **2008**, 1–47.
5. Healey, J.A.; Picard, R.W. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Trans. Intell. Transp. Syst.* **2005**, *6*, 156–166. [[CrossRef](#)]
6. Balasubramanian, V.; Adalarasu, K. EMG-based analysis of change in muscle activity during simulated driving. *J. Bodyw. Mov. Ther.* **2007**, *11*, 151–158. [[CrossRef](#)]
7. Deffenbacher, J.L.; Oetting, E.R.; Lynch, R.S. Development of a driving anger scale. *Psychol. Rep.* **1994**, *74*, 83–91. [[CrossRef](#)] [[PubMed](#)]
8. Hu, T.-Y.; Xie, X.; Li, J. Negative or positive? The effect of emotion and mood on risky driving. *Transp. Res. Part F Traffic Psychol. Behav.* **2013**, *16*, 29–40. [[CrossRef](#)]
9. Jeon, M. Don't Cry While You're Driving: Sad Driving Is as Bad as Angry Driving. *Int. J. Hum.-Comput. Interact.* **2016**, *32*, 777–790. [[CrossRef](#)]
10. Braun, M.; Pflöging, B.; Alt, F. A Survey to Understand Emotional Situations on the Road and What They Mean for Affective Automotive UIs. *Multimodal Technol. Interact.* **2018**, *2*, 75. [[CrossRef](#)]
11. Allen, J.G.; MacNaughton, P.; Satish, U.; Santanam, S.; Vallarino, J.; Spengler, J.D. Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments. *Environ. Health Perspect.* **2016**, *124*, 805–812. [[CrossRef](#)]
12. Satish, U.; Mendell, M.J.; Shekhar, K.; Hotchi, T.; Sullivan, D.; Streufert, S.; Fisk, W.J. Is CO₂ an Indoor Pollutant? Direct Effects of Low-to-Moderate CO₂ Concentrations on Human Decision-Making Performance. *Environ. Health Perspect.* **2012**, *120*, 1671–1677. [[CrossRef](#)]
13. Lutin, J.; Kornhauser, A.L.; Lerner-Lam, E. The Revolutionary Development of Self-Driving Vehicles and Implications for the Transportation Engineering Profession. *Cell* **2013**, *215*, 630–4125.
14. Mladenović, M.N.; Abbas, M.; McPherson, T. Development of socially sustainable traffic-control principles for self-driving vehicles: The ethics of anthropocentric design. In Proceedings of the 2014 IEEE International Symposium on Ethics in Science, Technology and Engineering, Chicago, IL, USA, 23 May 2014; pp. 1–8.
15. Greenblatt, N.A. Self-driving cars and the law. *IEEE Spectr.* **2016**, *53*, 46–51. [[CrossRef](#)]
16. Endsley, M.R. Autonomous Driving Systems: A Preliminary Naturalistic Study of the Tesla Model S. *J. Cogn. Eng. Decis. Mak.* **2017**, *11*, 225–238. [[CrossRef](#)]
17. McMurray, L. Emotional stress and driving performance: The effect of divorce. *Behav. Res. Highw. Saf.* **1970**, *1*, 100–114.
18. Legree, P.J.; Heffner, T.S.; Psotka, J.; Martin, D.E.; Medsker, G.J. Traffic crash involvement: Experiential driving knowledge and stressful contextual antecedents. *J. Appl. Psychol.* **2003**, *88*, 15–26. [[CrossRef](#)] [[PubMed](#)]

19. Norris, F.H.; Matthews, B.A.; Riad, J.K. Characterological, situational, and behavioral risk factors for motor vehicle accidents: A prospective examination. *Accid. Anal. Prev.* **2000**, *32*, 505–515. [[CrossRef](#)]
20. Lu, J.; Xie, X.; Zhang, R. Focusing on appraisals: How and why anger and fear influence driving risk perception. *J. Saf. Res.* **2013**, *45*, 65–73. [[CrossRef](#)]
21. Zimasa, T.; Jamson, S.; Henson, B. The influence of driver's mood on car following and glance behaviour: Using cognitive load as an intervention. *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *66*, 87–100. [[CrossRef](#)]
22. Jeon, M.; Zhang, W. Sadder but Wiser? Effects of Negative Emotions on Risk Perception, Driving Performance, and Perceived Workload. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, San Diego, CA, USA, 30 September 2013; pp. 1849–1853. [[CrossRef](#)]
23. Gjoreski, M.; Gjoreski, H.; Luštrek, M.; Gams, M. Continuous stress detection using a wrist device: In laboratory and real life. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*; Association for Computing Machinery: Heidelberg, Germany, 2016; pp. 1185–1193.
24. Halim, Z.; Rehan, M. On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Inf. Fusion* **2020**, *53*, 66–79. [[CrossRef](#)]
25. Izquierdo-Reyes, J.; Ramirez-Mendoza, R.A.; Bustamante-Bello, M.R.; Pons-Rovira, J.L.; Gonzalez-Vargas, J.E. Emotion recognition for semi-autonomous vehicles framework. *Int. J. Interact. Des. Manuf. IJIDeM* **2018**, *12*, 1447–1454. [[CrossRef](#)]
26. Kaplan, S.; Guvensan, M.A.; Yavuz, A.G.; Karalurt, Y. Driver Behavior Analysis for Safe Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3017–3032. [[CrossRef](#)]
27. Lazarus, R.S.; Folkman, S. *Stress, Appraisal, and Coping*; Springer Publishing Company: New York, NY, USA, 1984; ISBN 978-0-8261-4192-7.
28. Loft, S.; Sanderson, P.; Neal, A.; Mooij, M. Modeling and Predicting Mental Workload in En Route Air Traffic Control: Critical Review and Broader Implications. *Hum. Factors* **2007**, *49*, 376–399. [[CrossRef](#)] [[PubMed](#)]
29. Hansen, J.H.L. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech Commun.* **1996**, *20*, 151–173. [[CrossRef](#)]
30. Marksberry, K. What is Stress? *The American Institute of Stress*. Available online: <https://www.stress.org/what-isstress> (accessed on 11 September 2020).
31. Selye, H. Stress without Distress. In *Psychopathology of Human Adaptation*; Serban, G., Ed.; Springer: Boston, MA, USA, 1976; pp. 137–146.
32. Reimer, B.; Mehler, B.; Coughlin, J. *An Evaluation of Driver Reactions to New Vehicle Parking Assist Technologies Developed to Reduce Driver Stress*; Technical Report; New England University Transportation Center, Massachusetts Institute of Technology: Cambridge, MA, USA, 2010.
33. Mayou, R.; Bryant, B. Consequences of road traffic accidents for different types of road user. *Injury* **2003**, *34*, 197–202. [[CrossRef](#)]
34. Gulian, E.; Matthews, G.; Glendon, A.I.; Davies, D.R.; Debney, L.M. Dimensions of driver stress. *Ergonomics* **1989**, *32*, 585–602. [[CrossRef](#)]
35. Matthews, G.; Desmond, P.A.; Joyner, L.; Carcary, B.; Gilliland, K. A Comprehensive Questionnaire Measure of Driver Stress and Affect. Available online: <http://www.academia.edu/download/48157065/VALPROC.DOC> (accessed on 14 September 2020).
36. Mackay, C.; Cox, T.; Burrows, G.; Lazzarini, T. An inventory for the measurement of self-reported stress and arousal. *Br. J. Soc. Clin. Psychol.* **1978**, *17*, 283–284. [[CrossRef](#)]
37. Matthews, G. Stress states, personality and cognitive functioning: A review of research with the Dundee Stress State Questionnaire. *Personal. Individ. Differ.* **2020**, *110083*. [[CrossRef](#)]
38. Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*; Hancock, P.A., Meshkati, N., Eds.; Human Mental Workload; Elsevier: North-Holland, The Netherlands, 1988; Volume 52, pp. 139–183.
39. Pauzié, A. A method to assess the driver mental workload: The driving activity load index (DALI). *IET Intell. Transp. Syst.* **2008**, *2*, 315–322. [[CrossRef](#)]
40. Funke, G.; Matthews, G.; Warm, J.S.; Emo, A.K. Vehicle automation: A remedy for driver stress? *Ergonomics* **2007**, *50*, 1302–1323. [[CrossRef](#)]

41. Munla, N.; Khalil, M.; Shahin, A.; Mourad, A. Driver stress level detection using HRV analysis. In Proceedings of the 2015 International Conference on Advances in Biomedical Engineering (ICABME), Beirut, Lebanon, 16–18 September 2015; pp. 61–64.
42. Giannakakis, G.; Grigoriadis, D.; Giannakaki, K.; Simantiraki, O.; Roniotis, A.; Tsiknakis, M. Review on psychological stress detection using biosignals. *IEEE Trans. Affect. Comput.* **2019**. [[CrossRef](#)]
43. Can, Y.S.; Chalabianloo, N.; Ekiz, D.; Ersoy, C. Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study. *Sensors* **2019**, *19*, 1849. [[CrossRef](#)] [[PubMed](#)]
44. Nelson, B.W.; Allen, N.B. Accuracy of Consumer Wearable Heart Rate Measurement during an Ecologically Valid 24-Hour Period: Intraindividual Validation Study. *JMIR mHealth uHealth* **2019**, *7*, e10828. [[CrossRef](#)] [[PubMed](#)]
45. Lee, B.-G.; Lee, B.-L.; Chung, W.-Y. Smartwatch-based driver alertness monitoring with wearable motion and physiological sensor. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 6126–6129.
46. Westerman, S.J.; Haigney, D. Individual differences in driver stress, error and violation. *Personal. Individ. Differ.* **2000**, *29*, 981–998. [[CrossRef](#)]
47. Kim, J.; Park, J.; Park, J. Development of a statistical model to classify driving stress levels using galvanic skin responses. *Hum. Factors Ergon. Manuf. Serv. Ind.* **2020**, *30*, 321–328. [[CrossRef](#)]
48. Bitkina, O.V.; Kim, J.; Park, J.; Park, J.; Kim, H.K. Identifying Traffic Context Using Driving Stress: A Longitudinal Preliminary Case Study. *Sensors* **2019**, *19*, 2152. [[CrossRef](#)]
49. Coughlin, J.F.; Reimer, B.; Mehler, B. *Driver Wellness, Safety & the Development of an Awarecar*; Technical Report; New England University Transportation Center, Massachusetts Institute of Technology: Cambridge, MA, USA, 2009.
50. Lockhart, R.A. Interrelations between Amplitude, Latency, Rise Time, and the Edelberg Recovery Measure of the Galvanic Skin Response. *Psychophysiology* **1972**, *9*, 437–442. [[CrossRef](#)]
51. Yamakoshi, T.; Yamakoshi, K.; Tanaka, S.; Nogawa, M.; Park, S.B.; Shibata, M.; Sawada, Y.; Rolfe, P.; Hirose, Y. Feasibility study on driver's stress detection from differential skin temperature measurement. In Proceedings of the 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, BC, Canada, 20–25 August 2008; pp. 1076–1079.
52. Gao, H.; Yüce, A.; Thiran, J.-P. Detecting emotional stress from facial expressions for driving safety. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5961–5965.
53. Automatic cognitive load detection from speech features. In Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces; Available online: <https://dl.acm.org/doi/abs/10.1145/1324892.1324946> (accessed on 26 August 2020).
54. Fernández, A.; Usamentiaga, R.; Carús, J.L.; Casado, R. Driver Distraction Using Visual-Based Sensors and Algorithms. *Sensors* **2016**, *16*, 1805. [[CrossRef](#)]
55. Pruetz, J.; Watson, C.; Tousignant, T.; Govindswamy, K. *Assessment of Automotive Environmental Noise on Mobile Phone Hands-Free Call Quality*; Technical Report; SAE international: Warrendale, PA, USA, 2019. [[CrossRef](#)]
56. Lanatà, A.; Valenza, G.; Greco, A.; Gentili, C.; Bartolozzi, R.; Bucchi, F.; Frenzo, F.; Scilingo, E.P. How the Autonomic Nervous System and Driving Style Change With Incremental Stressing Conditions during Simulated Driving. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1505–1517. [[CrossRef](#)]
57. Meiring, G.A.M.; Myburgh, H.C. A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms. *Sensors* **2015**, *15*, 30653–30682. [[CrossRef](#)]
58. Lee, B.-G.; Chung, W.-Y. Wearable Glove-Type Driver Stress Detection Using a Motion Sensor. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1835–1844. [[CrossRef](#)]
59. Castaldo, R.; Melillo, P.; Bracale, U.; Caserta, M.; Triassi, M.; Pecchia, L. Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis. *Biomed. Signal Process. Control* **2015**, *18*, 370–377. [[CrossRef](#)]
60. Georgiou, K.; Larentzakis, A.V.; Khamis, N.N.; Alsuhaibani, G.I.; Alaska, Y.A.; Giallafos, E.J. Can Wearable Devices Accurately Measure Heart Rate Variability? A Systematic Review. *Folia Med. (Plovdiv)* **2018**, *60*, 7–20. [[CrossRef](#)] [[PubMed](#)]

61. Gilgen-Ammann, R.; Schweizer, T.; Wyss, T. RR interval signal quality of a heart rate monitor and an ECG Holter at rest and during exercise. *Eur. J. Appl. Physiol.* **2019**, *119*, 1525–1532. [CrossRef] [PubMed]
62. Kreibitz, S.D.; Wilhelm, F.H.; Roth, W.T.; Gross, J.J. Cardiovascular, electrodermal, and respiratory response patterns to fear- and sadness-inducing films. *Psychophysiology* **2007**, *44*, 787–806. [CrossRef] [PubMed]
63. Elite HRV—Top Heart Rate Variability App, Monitors, and Training. Available online: <https://elitehrv.com/> (accessed on 26 August 2020).
64. H10 User Manual Technical Specifications. Available online: https://support.polar.com/e_manuals/H10_HR_sensor/Polar_H10_user_manual_English/Content/Technical-Specifications.htm (accessed on 3 September 2020).
65. E4 Wristband Technical Specifications. Available online: <http://support.empatica.com/hc/en-us/articles/202581999> (accessed on 3 September 2020).
66. E4 Wristband—Real-Time Physiological Signals—Wearable PPG, EDA, Temperature, Motion Sensors. Available online: <https://www.empatica.com/research/e4> (accessed on 29 August 2020).
67. McCarthy, C.; Pradhan, N.; Redpath, C.; Adler, A. Validation of the Empatica E4 wristband. In Proceedings of the 2016 IEEE EMBS International Student Conference (ISC), Ottawa, ON, Canada, 29–31 May 2016; pp. 1–4.
68. Milstein, N.; Gordon, I. Validating Measures of Electrodermal Activity and Heart Rate Variability Derived From the Empatica E4 Utilized in Research Settings That Involve Interactive Dyadic States. *Front. Behav. Neurosci.* **2020**, *14*. [CrossRef]
69. Ledalab. Available online: <http://www.ledalab.de/> (accessed on 26 August 2020).
70. Xianglong, S.; Hu, Z.; Shumin, F.; Zhenning, L. Bus drivers' mood states and reaction abilities at high temperatures. *Transp. Res. Part F Traffic Psychol. Behav.* **2018**, *59*, 436–444. [CrossRef]
71. Chan, A.T.; Chung, M.W. Indoor-outdoor air quality relationships in vehicle: Effect of driving environment and ventilation modes. *Atmos. Environ.* **2003**, *37*, 3795–3808. [CrossRef]
72. Hancock, P.A.; Verwey, W.B. Fatigue, workload and adaptive driver systems. *Accid. Anal. Prev.* **1997**, *29*, 495–506. [CrossRef]
73. Zlatoper, T.J. Determinants of motor vehicle deaths in the united states: A cross-sectional analysis. *Accid. Anal. Prev.* **1991**, *23*, 431–436. [CrossRef]
74. Simion, M.; Socaciu, L.; Unguresan, P. Factors which Influence the Thermal Comfort Inside of Vehicles. *Energy Procedia* **2016**, *85*, 472–480. [CrossRef]
75. Daanen, H.A.M.; van de Vliert, E.; Huang, X. Driving performance in cold, warm, and thermoneutral environments. *Appl. Ergon.* **2003**, *34*, 597–602. [CrossRef]
76. Wörner, D.; von Bomhard, T.; Röschlin, M.; Wortmann, F. Look twice: Uncover hidden information in room climate sensor data. In Proceedings of the 2014 International Conference on the Internet of Things (IOT), Cambridge, MA, USA, 6–8 October 2014; pp. 25–30.
77. MacNaughton, P.; Spengler, J.; Vallarino, J.; Santanam, S.; Satish, U.; Allen, J. Environmental perceptions and health before and after relocation to a green building. *Build. Environ.* **2016**, *104*, 138–144. [CrossRef] [PubMed]
78. Petersen, J.; Kristensen, J.; Elarga, H.; Andersen, R.K.; Midtstraum, A. Accuracy and Air Temperature Dependency of Commercial Low-cost NDIR CO₂ Sensors: An Experimental Investigation. In Proceedings of the International Conference on Building Energy and Environment of COBEE2018, Melbourne, Australia, 5–9 February 2018; pp. 203–207.
79. Smart Indoor Air Quality Monitor—How Do I calibrate My Smart Indoor Air Quality Monitor? *Netatmo Helpcenter*. Available online: <https://helpcenter.netatmo.com/en-us/smart-indoor-air-quality-monitor/measures-and-calibrations/how-do-i-calibrate-my-smart-indoor-air-quality-monitor> (accessed on 29 August 2020).
80. Goh, C.C.; Kamarudin, L.M.; Shukri, S.; Abdullah, N.S.; Zakaria, A. Monitoring of carbon dioxide (CO₂) accumulation in vehicle cabin. In Proceedings of the 2016 3rd International Conference on Electronic Design (ICED), Phuket, Thailand, 11–12 August 2016; pp. 427–432.
81. Specifications for the Smart Indoor Air Quality Monitor. Available online: <https://www.netatmo.com/en-eu/aircare/homecoach/specifications> (accessed on 3 September 2020).
82. City Car Driving—Car Driving Simulator, PC Game. Available online: <https://citycardriving.com/> (accessed on 27 July 2020).
83. Logitech G920 & G29 Driving Force Steering Wheels & Pedals. Available online: <https://www.logitechg.com/en-us/products/driving/driving-force-racing-wheel.html> (accessed on 29 March 2020).

84. Specifications of Logitech G920 & G29 Driving Force Steering Wheels & Pedals. Available online: <https://www.logitechg.com/en-us/products/driving/driving-force-racing-wheel.html#product-tech-specs> (accessed on 3 September 2020).
85. SFML. Available online: <https://www.sfml-dev.org/> (accessed on 27 July 2020).
86. Ericsson, E. Independent Driving Pattern Factors and Their Influence on Fuel-Use and Exhaust Emission Factor. *Transp. Res. Part Transp. Environ.* **2001**, *6*, 325–345. [CrossRef]
87. Duncan Herrington, J. Effects of music in service environments: A field study. *J. Serv. Mark.* **1996**, *10*, 26–41. [CrossRef]
88. North, A.C.; Hargreaves, D.J. Can Music Move People? The Effects of Musical Complexity and Silence on Waiting Time. *Environ. Behav.* **2016**. [CrossRef]
89. Mayfield, C.; Moss, S. Effect of Music Tempo on Task Performance. *Psychol. Rep.* **1989**, *65*, 1283–1290. [CrossRef]
90. Brodsky, W. The effects of music tempo on simulated driving performance and vehicular control. *Transp. Res. Part F Traffic Psychol. Behav.* **2001**, *4*, 219–241. [CrossRef]
91. Spotify. Available online: <https://www.spotify.com/es/> (accessed on 27 July 2020).
92. Joshi, A.; Kale, S.; Chandel, S.; Pal, D.K. Likert Scale: Explored and Explained. *Curr. J. Appl. Sci. Technol.* **2015**, 396–403. [CrossRef]
93. Limiyati, Y.; Wahyudianingsih, R.; Maharani, R.D.; Christabella, M.T. Mozart’s Sonata for Two Pianos K448 in D-Major 2nd Movement Improves Short-Term Memory and Concentration. *J. Med. Health* **2019**, *2*. [CrossRef]
94. Rendon-Velez, E.; van Leeuwen, P.M.; Happee, R.; Horváth, I.; van der Vegte, W.F.; de Winter, J.C.F. The effects of time pressure on driver performance and physiological activity: A driving simulator study. *Transp. Res. Part F Traffic Psychol. Behav.* **2016**, *41*, 150–169. [CrossRef]
95. Malik, M. Heart Rate Variability. *Ann. Noninvasive Electrocardiol.* **1996**, *1*, 151–181. [CrossRef]
96. Snow, S.; Boyson, A.; Felipe-King, M.; Malik, O.; Coutts, L.; Noakes, C.J.; Gough, H.; Barlow, J.; Schraefel, M.C. Using EEG to characterise drowsiness during short duration exposure to elevated indoor Carbon Dioxide concentrations. *bioRxiv* **2018**, 483750. [CrossRef]
97. Nævestad, T.-O.; Laiou, A.; Phillips, R.O.; Bjørnskau, T.; Yannis, G. Safety Culture among Private and Professional Drivers in Norway and Greece: Examining the Influence of National Road Safety Culture. *Safety* **2019**, *5*, 20. [CrossRef]
98. Rony, R.J.; Ahmed, N. Monitoring Driving Stress using HRV. In Proceedings of the 2019 11th International Conference on Communication Systems Networks (COMSNETS), Bengaluru, India, 7–11 January 2019; pp. 417–419.
99. Kontogiannis, T. Patterns of driver stress and coping strategies in a Greek sample and their relationship to aberrant behaviors and traffic accidents. *Accid. Anal. Prev.* **2006**, *38*, 913–924. [CrossRef] [PubMed]
100. Lazarus, R.S.; Lazarus, R.S. *Emotion and Adaptation*; Oxford University Press: New York, NY, USA, 1991.
101. Frijda, N.H.; Fridja, N.H.A. *The Emotions*; Cambridge University Press: Cambridge, UK, 1986; ISBN 0521301556.
102. Huffziger, S.; Kuehner, C. Rumination, distraction, and mindful self-focus in depressed patients. *Behav. Res. Ther.* **2009**, *47*, 224–230. [CrossRef]
103. Jallais, C.; Gabaude, C.; Paire-figout, L. When emotions disturb the localization of road elements: Effects of anger and sadness. *Transp. Res. Part F Traffic Psychol. Behav.* **2014**, *23*, 125–132. [CrossRef]
104. Sikander, G.; Anwar, S. Driver Fatigue Detection Systems: A Review. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2339–2352. [CrossRef]
105. Alkinani, M.H.; Khan, W.Z.; Arshad, Q. Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges. *IEEE Access* **2020**, *8*, 105008–105030. [CrossRef]
106. Hartley, L.R. *Fatigue and Driving: Driver Impairment, Driver Fatigue, and Driving Simulation*; Routledge: London, UK, 2019; ISBN 978-1-351-44885-7.
107. Driving Time and Rest Periods. Available online: https://ec.europa.eu/transport/modes/road/social_provisions/driving_time_en (accessed on 25 August 2020).
108. Stern, H.S.; Blower, D.; Cohen, M.L.; Czeisler, C.A.; Dinges, D.F.; Greenhouse, J.B.; Guo, F.; Hanowski, R.J.; Hartenbaum, N.P.; Krueger, G.P.; et al. Data and methods for studying commercial motor vehicle driver fatigue, highway safety and long-term driver health. *Accid. Anal. Prev.* **2019**, *126*, 37–42. [CrossRef]

109. Jovanis, P.P.; Wu, K.-F.; Chen, C. Hours of Service and Driver Fatigue: Driver Characteristics Research. Available online: <https://rosap.ntl.bts.gov/view/dot/70> (accessed on 11 September 2020).
110. Hanowski, R.J.; Olson, R.L.; Bocanegra, J.; Hickman, J.S. Analysis of Risk as a Function of Driving-Hour: Assessment of Driving Hours 1 through 11. Available online: https://rosap.ntl.bts.gov/view/dot/69/dot_69_DS1.pdf? (accessed on 14 September 2020).
111. Jung, H.S.; Grady, M.L.; Victoroff, T.; Miller, A.L. Simultaneously reducing CO₂ and particulate exposures via fractional recirculation of vehicle cabin air. *Atmos. Environ.* **2017**, *160*, 77–88. [CrossRef]
112. Kajtár, L.; Herczeg, L.; Lang, E. Examination of influence of CO₂ concentration by scientific methods in the laboratory. *Proc. Healthy Build.* **2003**, *3*, 176–181.
113. Kajtár, L.; Herczeg, L. Influence of carbon-dioxide concentration on human well-being and intensity of mental work. *Időjárás* **2012**, *116*, 145–169.
114. Rödjegård, H.; Franchy, M.; Ehde, S.; Zoubir, Y.; Al-Khaldy, S.; Olsson, P.; Bengtsson, C.; Nowak, T.; O'Brien, D. *Drowsy Driver & Child Left Behind-Prevention via in Cabin CO₂ Sensing*; SAE International: Warrendale, PA, USA, 2020.
115. Kuribayashi, R.; Nittono, H. Speeding up the tempo of background sounds accelerates the pace of behavior. *Psychol. Music* **2014**. [CrossRef]
116. Oakes, S. The influence of the musicscape within service environments. *J. Serv. Mark.* **2000**, *14*, 539–556. [CrossRef]
117. Fountas, G.; Pantangi, S.S.; Hulme, K.F.; Anastasopoulos, P.C. The effects of driver fatigue, gender, and distracted driving on perceived and observed aggressive driving behavior: A correlated grouped random parameters bivariate probit approach. *Anal. Methods Accid. Res.* **2019**, *22*, 100091. [CrossRef]
118. Millet, B.; Ahn, S.; Chattah, J. The impact of music on vehicular performance: A meta-analysis. *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *60*, 743–760. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Assessment of the Speed Management Impact on Road Traffic Safety on the Sections of Motorways and Expressways Using Simulation Methods

Jacek Oskarbski ^{1,*}, Tomasz Kamiński ², Kyandoghere Kyamakya ³, Jean Chamberlain Chedjou ³, Karol Żarski ¹ and Małgorzata Pędzierska ²

¹ Faculty of Civil and Environmental Engineering, Gdansk University of Technology, 80-233 Gdańsk, Poland; karol.zarski@pg.edu.pl

² Motor Transport Institute, 03-301 Warszawa, Poland; tomasz.kaminski@its.waw.pl (T.K.); malgorzata.pedzierska@its.waw.pl (M.P.)

³ Institute for Smart Systems Technologies, University Klagenfurt, A9020 Klagenfurt, Austria; kyandoghere.kyamakya@aau.at (K.K.); jean.chedjou@aau.at (J.C.C.)

* Correspondence: jacek.oskarbski@pg.edu.pl; Tel.: +48-604-475-876

Received: 8 July 2020; Accepted: 3 September 2020; Published: 5 September 2020

Abstract: Methods used to evaluate the impact of Intelligent Transport System (ITS) services on road safety are usually based on expert assessments or statistical studies. However, commonly used methods are challenging to apply in the planning process of ITS services. This paper presents the methodology of research using surrogate safety measures calculated and calibrated with the use of simulation techniques and a driving simulator. This approach supports the choice of the type of ITS services that are beneficial for traffic efficiency and road safety. This paper presents results of research on the influence of selected scenarios of variable speed limits on the efficiency and safety of traffic on the sections of motorways and expressways in various traffic conditions. The driving simulator was used to estimate the efficiency of lane-keeping by the driver. The simulation traffic models were calibrated using driving simulator data and roadside sensor data. The traffic models made it possible to determine surrogate safety measures (number of conflicts and their severity) in selected scenarios of using ITS services. The presented studies confirmed the positive impact of Variable Speed Limits (VSLs) on the level of road safety and traffic efficiency. This paper also presents recommendations and plans for further research in this area.

Keywords: variable speed limits; intelligent transportation systems; ITS services; driving simulator studies; traffic modelling; surrogate safety measures

1. Introduction

Variable Speed Limit (VSL) systems have been implemented in many countries as a method of improving traffic flow and road safety. Upon completion of the data analysis process, the recommended speed limits are dynamically updated, with new messages being displayed on Variable Message Signs (VMSs) to influence driver behaviour. The VSL algorithms are usually based on speed, occupancy and volume variables. The desired speed is reduced upstream to limit the spread of shock waves at critical values of the variables defined in the algorithm. Properly designed VSL systems reduce the number and severity of accidents, travel time and emissions by harmonising traffic flow speed [1–3]. The benefits of VSL were also presented by Papageorgiou et al. [4] and by Abdel-Aty et al. [5], where the development of a crash model was described. Li et al. [6] presented the impact of VSL on reducing the number of secondary collisions in poor visibility conditions. For this purpose, Li used a modified car-following model. VSL systems contribute to improving traffic safety by reducing the speed

difference between vehicles and minimising the speed variation, resulting in less frequent lane changes and sudden braking.

In 2017–2019, 2078 accidents occurred on Polish motorways and expressways. As a result, 2965 people were injured (including 773 seriously injured) and 295 people died [7]. Further expansion of the high-speed road network in Poland, and thus shifting the majority of the traffic to expressways, may consequently lead to an increase in the number of collisions and accidents, as well as the number of fatalities and injuries on these roads.

In 2011–2015, 2339 accidents (1.3% of the total number of accidents) occurred on motorways and expressways within a length of 3194.55 km (17% of the length of national roads, 1.1% of all roads in Poland). They caused 402 deaths (2.3% of the total number of fatalities) and 3443 injuries (1.6% of the total). The analysis of trends in 2011–2015 indicated the growing risk of being involved in an accident on motorways and expressways. Although these changes are caused by the development of the road network, the increase in the number of accidents and victims is more significant than the increase in the length of these roads. As the data show, motorways exceed expressways in terms of increased collision risk. The situation is even more worrying when compared to other national roads, where the risk has decreased significantly in the same period. The accident (number of accidents per 100 km) and victim (number of victims per 100 km) densities for motorways and expressways are much lower than on other national roads. Nevertheless, these rates on expressways and motorways are increasing, while on other roads a reduction is observed. Analysis of the data showed that there was a decrease in the severity of accidents (except for motorways). However, accidents on motorways (16 fatalities per 100 accidents) and expressways (15 fatalities per 100 accidents) were much more severe than on other roads (nine fatalities per 100 accidents). Among the accidents that occurred on motorways and expressways, rear-end collisions of vehicles were the most frequent (34.2 % of all accidents). Besides, drivers were found to exceed speed limits (19%), not maintain a safe distance (8%) and change lanes incorrectly (4%), leading to a collision. On motorways and expressways (similarly to other roads), road accidents most often occur during the day (64%) and up to 22% at night on unlit roads. Up to 64% of accidents occur in good weather conditions and, less frequently than on other national roads, accidents occur in adverse weather conditions: cloudy weather (18%), rainfall (11%), fog (2%) and snowfall (2%). It should be noted that on motorways and expressways, 72% of vehicles involved in accidents are passenger cars, while more often than on other national roads, Heavy Goods Vehicles (HGVs) are involved in accidents (23%) [8].

The authors of this paper identified the coverage of Polish motorways and expressways with different Intelligent Transport System (ITS) services based on data received from the National Road Administration (GDDKiA) under the project “The impact of the usage of Intelligent Transport System services on the level of road safety” (RID-4D) [9]. The studies of road traffic safety for motorway sections was carried out based on accidents and traffic volume data gathered in 2013–2015. For the assessment of road traffic safety, motorway sections without ITS services and sections with the implemented ITS services (primarily the provision of information to drivers via VMSs about weather conditions, adverse surface condition and related speed limits) were selected. Due to dispersed ITS services and the lack of a testing field and data for the case of using a series of variable message signs, it was not possible to apply statistical studies identifying the impact of an ITS service such as VSLs on the road safety level. The level of traffic safety was assessed using a risk estimation of individual involvement in an accident or becoming an accident victim. The risks were represented by the number of accidents and victims of these accidents per million vehicle kilometres travelled (VKT). The highest number of accidents per VKT was recorded on sections of motorways without ITS services. On sections equipped with ITS devices near urban areas, the risk of being a participant or a victim of an accident was higher than on motorway sections distant from highly urbanised areas. Similar relationships were observed for serious injuries and fatalities. The individual risk of being involved in an accident was 37% lower on motorways with ITS services than on motorways without such services. The number of fatalities

per VKT was also 49% lower and the number of injuries was 26% lower. These results may suggest a positive influence of selected ITS services on the level of road traffic safety.

The results of the studies presented above [8,9] indicate that speed management measures (including Variable Speed Limits—VSLs) that can contribute to reducing collision risks are advisable to improve the level of road traffic safety. A significant proportion of accidents involving over-speed drivers and those who do not maintain a safe distance between vehicles require measures to harmonise traffic flow. The lack of sufficient data, and often, as in the case of Poland, the lack of test sites with fully functioning VSL systems, is the reason for using simulation methods to assess road traffic safety.

Statistical studies of accidents and their victims are a direct and widely used way to evaluate road safety issues. The condition for carrying out reliable statistical analyses is the use of data available over a longer period of several years and when the characteristics of the road and its surroundings do not change during this period. Insufficient statistical information creates a barrier in evaluating the safety level for the newly constructed roads or the roads in the designing or planning stages. The available data samples are small or evidence does not exist (including those for the accident rates). The above statement also applies to the planned road safety improvement measures to be implemented, including ITS services if they were not previously used and tested on the road [10].

The possibility to reliably assess the impact of planned ITS services on the functioning of the transport system is crucial considering the many services implemented on the roads which have successfully improved road safety and traffic efficiency. The accident data directly indicate both the structure and causes of the safety level. For each specific time and element of the transport system, the expected number of accidents or their victims can be estimated based on the known risk factors, consequences and exposure [11]. Many tools have been developed to select, process, analyse and visualise traffic accident data. Unfortunately, such tools do not take into account future changes that may affect the level of safety (including the development of new ITS services for vehicles or road infrastructure). The above statements demonstrate that the statistical analyses have little potential to assess ITS services, especially those recently introduced or planned for introduction. Besides, statistics are less useful for determining the causes of accidents, as they are rarely recorded in sufficient detail to conclude the complex chain of incidents preceding the accident [12]. Other safety assessment measures may be used when planning or introducing innovative road improvements.

One possibility is to use models to predict the number of accidents taking into account road characteristics (e.g., class of road) and traffic volume forecasts [13–16]. In 2010, the American Association of State Highway Transportation Officials (AASHTO) published the results of more than 10 years of research work carried out by many scientific centres and experts in the form of the Highway Safety Manual (HSM) [17]. The HSM presents a method of forecasting the number of accidents, victims and their costs. In 2012, the HSM method was extended [18]. The procedure of calculating the projected average number of accidents for individual elements of the motorway was presented for sections between interchanges, ramps, weaving and merging sections and also junctions within an interchange. Moreover, research was conducted using Bayesian methods [19–23] and advanced statistical techniques (e.g., classification and regression trees) to verify the results of analyses made using observations (traffic conflicts technique) [24–26]. Other ways of combining measures were also proposed to estimate the “level of service safety”, analogous to the traffic level of service (LOS) [27] or indicators for particular types of incidents (e.g., material loss incidents) [28]. Macroscopic measures of traffic flow were used in the proposed method.

Road safety researchers also use Surrogate Safety Measures (SSMs), derived from the theory of traffic conflicts. These measures are based on indirect indicators such as differences in speeds or estimated time to a collision of two interacting vehicles, to identify traffic conflicts and calculate their number. SSMs can be estimated based on the recording, analysis and comparison of trajectories and changes in movement dynamics of vehicles or other traffic users. Vehicle trajectories can be estimated by analysing data obtained from a real road section or junction (e.g., using video or RADAR techniques), data from a driving simulator or data obtained from simulations developed using traffic models.

This paper presents a novel methodology of research and assessment of road traffic safety using surrogate safety measures calculated based on simulation models and supported by driving simulator studies. In the presented studies, the Surrogate Safety Assessment Model (SSAM) was used for the first time in assessing the safety of ITS services, such as VSLs, with the use of surrogate safety measures. The added value is also the study and results of the location of variable message signs in the VSL system on a road section developed using the SSAM. The SSAM has so far been used to assess road traffic safety mainly due to the geometric parameters of the road and fixed traffic organisation measures [29], parameters of intersections (signalised or unsignalised) [30–33] or ramp metering ITS services [34]. The location and the number of variable message signs along road sections, depending on the traffic volume or accompanying road incidents, can influence the places where dangerous spots occur.

For this reason, it is important to identify such places in advance and limit their number to reduce risks. The surrogate safety measures are widely used in road safety analyses, however, to date the research has not been focused on such a broad approach as presented in this paper. This publication takes into account not only the impact of measures on individual elements of the road network (including types of intersections) but also on the entire network of co-existing roads (major road corridor). The method of calibrating the simulation model was also an added value. In the calibration process, data from sensors (inductive loops) of traffic measurement stations located on road cross sections were used. The data allowed us to develop cumulative distribution functions of key traffic flow variables in various traffic conditions and for different types of vehicles. The data from traffic measurement stations allowed us to develop functions in road cross sections (at selected points on each lane). To take into account the impact of the information displayed on the VMS, it was necessary to use data from the driving simulator, which made it possible to reproduce drivers' behaviour (speed changes) along the road section upstream and downstream of the VMS. The dynamic model was used for traffic assignment in the road network.

The applied techniques can be used to assess road safety on planned roads and to determine changes in the safety level in case of planned road modernisation or the implementation of ITS services. Solutions in the field of ITS services on Polish motorways and expressways are currently being implemented on a large scale within the National Traffic Management System (KSZR). Poland lacks detailed guidelines for determining the structure of VSL systems (taking into account the location of VMSs and also sensors collecting data for traffic control) and the resulting distribution of VMSs, as well as the implementation of such a service on different roads. The presented research methodology, which takes into account different levels of traffic intensity and the occurrence of road incidents, may support the development of such guidelines.

2. Simulation Methods of Road Safety Assessment

Several methodologies have been proposed and applied in the scientific literature to collect and analyse data on SSMs and road users' behaviour, including:

- naturalistic driving studies [35–38],
- site-based observation studies [39–41],
- microsimulation modelling studies [30,42,43],
- driving simulator studies [44–47].

The first two methodologies reflect the behaviour of road users in a real road environment, while the latter two can be considered as a controlled form of data collection in which researchers can manipulate and control traffic events [48]. This paper presents the application of methods based on simulations using microsimulation traffic models and a driving simulator.

2.1. Microsimulation Modelling Studies

The calculation of surrogate safety measures in microsimulation modelling studies is supported by traffic simulation models [29,49,50]. The simulation of traffic users' behaviour when driving through a

defined virtual road network requires the use of microsimulation models, which are computerised analytical tools [30]. The stochastic parameters adopted in the simulation models allow each traffic user to be treated as an individual unit and for the interaction between these individual units to be defined. [42]. These parameters make it possible to define individual preferences and trends in traffic users' behaviour at a reasonable level of approximation. The Surrogate Safety Assessment Model (SSAM) is a tool for the collection and preliminary analysis of SSMs from microsimulation models. The SSAM is a post-processing tool using vehicle trajectories generated in microsimulation packages [51]. The use of SSMs and methods of assessing the road safety level on motorways and expressways require the consideration of three main types of events: rear-end collisions, side collisions (these two types of events require the participation of at least two vehicles) and single-vehicle accidents [52]. The use of commonly available simulation techniques is hampered by the need to take into account interactions with infrastructure elements in addition to the trajectory analysis in the case of incidents involving single vehicles. For the incidents involving one vehicle, the location of the accident (within the lane, off the road, within the sidewalk/emergency lane) and the severity of the accident were used to determine the accident topology [53]. The main benefit of research using microsimulation models is the possibility to evaluate the impact of the road infrastructure and ITS services on traffic safety proactively and without significant financial resources. The usefulness of most of the algorithms used in microsimulation models for safety assessments is limited due to focus on typical driver's behaviour and the inability to take into account vehicle collision occurrences [43]. Some simplifications are inevitably necessary, even in the most advanced models. Therefore, the importance of results as a representation of actual road users' behaviour could be discussed.

Simulation techniques at the microscopic level are used for the assessment of road safety based on surrogate measures. Golob et al. [53] indicated that the mean traffic volume, median speed and instantaneous deviations in the values of volume and speed significantly affect the possibility of an incident occurring. Xin et al. and Evans and Wasilewski showed that the most common cause of accidents is too small headways between vehicles [54,55]. Surrogate (indirect) safety measures might be speed and its variations, distances between vehicles in the traffic flow, traffic-related measures (including occupancy, traffic density, etc.) and lane change manoeuvres [30]. The most common measures used in simulation models are those used in traffic conflict theory [56]. The observation of sudden braking and avoidance manoeuvres makes it possible to identify traffic conflicts. The research conducted with the use of surrogate measures allows us to find a connection between the conflict and the real-life accident [57–60]. A traffic conflict is an observed situation in which two or more road users approach each other in space and time to such an extent that there is a risk of a collision if their movements (speed and direction) remain unchanged. In the face of the possibility of a collision, a fast, decisive manoeuvre of the vehicle, pedestrian or cyclist is required to avoid it. The underlying assumption of this method is that the greater the number of traffic conflicts, the more likely it is for an accident to occur. The method involves observing the elements of the road system and noting the conflict situations, i.e., those that could lead to the occurrence of an accident in particular areas of the road network (on a road section, in different parts of a junction).

Conflict is defined as an often-repeated behaviour of road users that can lead to an accident (e.g., braking too late, selection of an incorrect driving trajectory). Several basic measures, which are characteristic of traffic conflicts, have been proposed, e.g., Time To Collision (TTC), Deceleration Rate (DR) and the time interval between collision vehicles—leaving the collision point and arriving at the collision point (Post-Encroachment Time—PET). The most commonly used SSMs of traffic safety in the research was TTC followed by PET and their derivatives [30]. Surrogate measures may be used to determine the degree of significance of the conflict, which translates into the probability of accident severity [61]. The conflict technique allows us to obtain more data for analysis, but the parameters used to describe the manoeuvres/behaviour of drivers are indirect indicators of accident risk and the reduction of its severity.

Despite attempts to use the aggregated data to calculate the risk of incident occurrence, there are still uncertainties about the effectiveness of using the above-mentioned measures as risk measures [23]. However, it was estimated that the ratio between the frequency of conflicts calculated with the use of surrogate measures and the frequency of accidents is 20,000 to 1 [51]. SSMs can replace statistical measures for accidents and their victims. Observations of SSMs may be supplemented by behavioural observations and/or data from other fields, such as driving simulator tests. Traffic conflict theory is a proactive safety research method that can be used without waiting for accidents to happen as well as to simulate planned solutions. Surrogate measures are most often used in research conducted using microsimulation traffic models and driving simulators.

2.2. Driving Simulator Studies

Another method of data collection that enables the simulation of the real road environment is the use of a driving simulator. Driving simulators aim to reproduce the real road environment in a virtual world by placing participants in a mock-up of the vehicle interior and displaying the moving road and its surroundings on the screens. The most advanced simulators are high-level ones, which use virtual projection on screens around the vehicle and a mobile base platform on which the vehicle is placed [62]. The advantages of driving simulator research over field research are as follows [45]:

- possibility of using proactive research methods,
- generally unlimited possibility of defining the road environment according to the criteria assumed by the researcher,
- high level of detail and scope of collected data,
- ensuring the safety of test participants even for tests that would be dangerous in the real environment.

The main disadvantage of driving simulator tests is the limitation of visual realism that can be offered [44]. Due to limited realism, which can contribute to abnormal driver behaviour, the validity of driving simulator tests is quite often questioned [44,62]. However, many studies proved that driving simulators tend to achieve a high level of relative validity [45–47] and can be an important tool for comparing safety aspects between different controlled experimental scenarios.

Similar or SSM-related measures were considered for research using driving simulators. For example, one of the safety measures is Time to Line Crossing (TLC). This is a measure used to determine the remaining time to collision in a conflicting situation before the vehicle crosses the lane border. One of the measures of driver distraction may be the so-called information on staying within a lane (lane-keeping), i.e., the distance between the longitudinal axis of the vehicle and the lane axis [63,64]. It is possible, then, to analyse the driver's effectiveness of staying within a lane (lateral control capability), which in real conditions is usually assessed by measuring lateral acceleration and the Standard Deviation of Lateral Position (SDLP) [65]. SDLP is a measure similar to TLC, reflecting the degree of control the driver has over the vehicle in each particular driving situation and is related to the probability of going off the road. It should be emphasised that inadequate lane-keeping is one of the basic factors contributing to road collisions [66].

Blaschke et al. [67] stated that drivers who were given additional information via In-Vehicle Information Systems (IVISs) increased, in most cases, the distance between the vehicle's axis and the lane axis (so-called lateral deviation). The tests were conducted in real conditions. The authors defined distraction as "any activity that diverts the driver's attention away from the task of driving." They also referred to the research presented by Klauer et al. [68] regarding the scope of additional activities performed by drivers (not directly related to driving a vehicle) and the likelihood of a collision in the case of these activities. In almost 80% of collisions and 65% of incidents (situations close to collision), the driver did not pay appropriate attention.

Peng et al. [65] studied the driving paths of 24 vehicle drivers, driving a vehicle under real conditions. Based on this, they were divided into two groups, i.e., drivers who watched the road

in front of the vehicle and a group of drivers who directed their eyes off the road, performing an additional task. A comparison of the results of lateral deviation measurements between groups showed an increase in the standard deviation of the lateral deviation in the case of the group of drivers taking their eyes off the road. An increase in cognitive load can, in some special situations, result in increased effectiveness in lane-keeping. Such a phenomenon can be encountered, for example, in the case of icy roads, when drivers focus their efforts on the activities related to observing the surroundings and the vehicle's behaviour to maintain control, ensuring the maintenance of the driving path and vehicle's speed. A similar principle is described by He et al. [69]. The tests were performed using a high-level driving simulator. Drivers drove the vehicle in a crosswind, being tasked with maintaining a stable lane position while the speed was stabilised using cruise control. This enabled drivers to focus on keeping the vehicle within the lane. During this engaging task, drivers heard audio recordings of numbers spoken by a sound synthesising program. It was a so-called n-back task. Drivers performed an additional task of repeating the four numbers heard in the order in which they heard them. Then the difficulty level of this task increased and they had to repeat the numbers they heard in ascending order, becoming more involved in the task. The authors of the article concluded that in the case under study, the increase in cognitive load, which, although it disrupts the activities performed by the driver, increases the effectiveness of lane-keeping. The measures described above are useful for comparative driving simulator studies but are also difficult to determine in field measurements without the use of advanced vehicle equipment [30].

2.3. Application of Sensors to Improve Road Traffic Safety and SSM-Related Microsimulation Studies

The traffic management tasks, including road traffic safety management, cover three main areas, which are:

- estimation of the traffic state in which data from different traffic sensors and traffic flow models fed by them are used to reproduce the traffic state picture of the whole road network (e.g., in terms of traffic density, speed and current dynamics of changes in traffic parameter values),
- prediction of the traffic state in which traffic projection in the future is calculated (short-term predictions are used to address traffic control issues),
- optimisation of traffic control measures (e.g., algorithms such as route guidance, VSLs, ramp metering, incident detection, etc.), the results of which are transmitted to the traffic control systems using actuators (traffic signals, VMSs, other roadside or in-vehicle information panels, etc.), including emergency events when traffic incident management is activated.

The use of the appropriate type of sensors enables us to collect the necessary and adequate data for a given traffic management process, but also data for modelling traffic control systems using simulation methods to improve traffic management strategies and tasks. Over recent decades, sensor technology has been developing more and more rapidly and has become ubiquitous. This has opened up new opportunities for the establishment and development of ITS services and the use of data for traffic modelling, including road safety studies. Sensors improving traffic safety can be installed both in vehicles and in the road environment. In the case of in-vehicle systems, speed sensors, RADAR and laser beams, micro-mechanical oscillators, cameras, inertial sensors, proximity sensors, ultrasonic sensors and haptic and night vision sensors are most often used to improve traffic safety. They are part of safety systems that focus on near real-time recognition of accident hazards and events [70–73]. The behaviour of drivers changes while warnings from sensors occur. As the number of such vehicles increases, this will need to be taken into account in the modelling of drivers' behaviour, including the modelling and estimating of surrogate safety measures. Sensors that can be used to determine the distance between a vehicle and another one can be, for example, ultrasonic or electromagnetic sensors. Ultrasonic sensors allow for the identification of the distance between a vehicle and an object, warning the driver when he or she is approaching another vehicle above a defined distance threshold. Electromagnetic sensors are used to warn the driver when another vehicle is in an electromagnetic field

generated around the bumpers. These types of sensors can be used to calculate the number of traffic conflicts not only between vehicles but also between vehicles and other objects in the road environment. The disadvantage of this type of sensor is a reduction in measuring accuracy due to humidity and temperature. Speed sensors and RADAR sensors are used to warn the driver of potential danger when changing lanes or detecting movement out of the lane [74]. Information from these sensors can also be used as a basis for analysing surrogate safety measures. Accelerometric and gyroscope sensors are used in conjunction with Global Positioning Systems (GPS) to improve the accuracy of navigation systems for determining vehicle parameters such as position and speed. The accuracy of the data obtained from such solutions does not appear to be sufficient to model traffic at the microscopic level and thus to estimate SSMS. Light Detection And Ranging (LIDAR) enables vehicles (especially autonomous vehicles for which it is one of the key elements) to observe the road environment through 360° continuous visibility and very accurate depth information. LIDAR was applied for the collection of surrogate safety measures [75]. Sensor data (timestamps, the precise location of other vehicles and objects) can be used to estimate and validate SSMS. Camera-based image processing methods are used in in-vehicle systems to monitor the position of the driver's head and eye activity. It enables the detection of fatigue, unusual vehicle behaviour (lane departure) [76], the appearance of an object within the road (e.g., a sudden pedestrian or an animal crossing the road, the appearance of another object on the road) and is also used as a basis for night vision applications. Object appearance around the vehicle recorded by the camera system can be useful in estimating SSMS. Other sensors that can be used in SSM estimation are Radio Detection And Ranging (RADAR) and laser sensors. They constantly scan the road in the vicinity of the vehicle to detect dangerous proximity to other vehicles or objects. Dangerous proximity detection allows safety applications to adjust the throttle and apply the brakes to prevent potential collisions. The RADAR sensors use radio waves to determine the distance to an obstacle. Applications notify the driver when a hazard is detected and can automatically apply the brakes to avoid a collision [70].

Mobile system data [77] (connected vehicles, Internet of Things, smartphones, new ways of information flow and cloud computing) enable the low-cost determination of vehicle speeds [78], vehicle travel time [79], vehicle tracking profiles (instantaneous speed, acceleration, deceleration) [80] and road safety performance assessment [81]. Data from mobile systems (especially vehicle tracking data) can be a valuable source of data for calibrating traffic models and estimating SSMS (if they are collected continuously) [80,82]. Further limitations on the use of individual sensors and the possibilities to use them for estimating SSMS are discussed below in the section on sensors installed in the road environment.

The automotive industry has invested a lot of funds in increasing safety, performance and comfort in vehicles by using sensors. However, the collection of traffic data with sensors along the roadside remains one of the main challenges for the development of ITS services. The deployment of sensors in the transport network provides drivers with many services, such as traffic management in road networks (traffic control, speed control, accessibility management, detection of incidents and objects on the road). Appropriate control strategies aim at improving the safety, reliability and resilience of the road network. Sensors can be divided into two categories (invasive and non-invasive), depending on their location in the road environment [83]. The invasive sensors are installed on the roadway surface. They are characterised by high accuracy, but also by relatively moderate installation and maintenance costs (installation and maintenance often require the temporary closure of road lanes and can contribute to shortening the life cycle of the pavement). Two groups of the invasive sensors are most commonly used: passive magnetic sensors and inductive loops, which send data to processing units. Inductive loops are most often used on Polish roads. The main advantage of invasive road sensors (especially inductive loops) is their technological maturity and large experience base. They have been widely implemented and are characterised by high accuracy in the detection of basic traffic parameters (volume, presence, occupancy, speed, headway, gap). These sensors are also insensitive to inclement weather (rain, fog, snow). The accuracy of the data collected by these sensors

enables their use in traffic modelling and SSM estimation. The disadvantage that is described later in this paper is the possibility to collect data only in road cross sections where the sensors are installed. Alternatives to invasive sensors are non-invasive technologies [84,85].

The most promising non-invasive sensors that can be used in studies based on SSMs are RADAR sensors and a Video Image Processor (VIP). RADAR sensors emit low-energy microwave radiation, which is reflected by all objects in the detection area. We can distinguish between different types of RADAR sensor systems. The first type is Doppler systems, which allow for counting the number of vehicles and their speed. There are also continuous-wave RADAR with frequency modulation, which are used to measure the traffic volume, speed and presence of vehicles. RADAR sensors are very accurate and easy to install. They support multiple lane operation and can operate in the dark or adverse weather conditions. Their main disadvantage is their susceptibility to electromagnetic interference. In a Video Image Processor (VIP), video cameras placed on the roadside collect and analyse the video images from the road section or intersection using advanced software to determine changes between successive image frames. This technology enables the measurement of traffic parameters such as traffic volumes, speed, presence of vehicles and classification of vehicles. The main disadvantage of VIP systems is that they are prone to performance degradation due to adverse weather conditions (rain, fog, snow, wind) or vehicle shadows, occlusion and vehicle/road contrast [86,87]. A VIP was applied for the collection of surrogate safety measures [88]. The use of sophisticated algorithms in the software enables the analysis of the trajectory of individual vehicles on a given road section, which makes RADAR and VIP sensors the most suitable sensors for traffic safety analysis with SSMs.

Other types of sensors that are mainly used in road cross sections can also be mentioned: infrared, acoustic array and ultrasonic sensors. These sensors allow us to measure traffic parameters and support multiple lane operation. Sensors (e.g., piezoelectric, quartz, tensometric, fibre optic or capacitive) are also used to weigh vehicles in motion. These sensors can also be used to count the number of vehicles, their speed and to determine their classification, but due to high installation and maintenance costs, their main goal is to weigh vehicles as a part of Weight-In-Motion (WIM) systems. The use of WIM systems improves traffic safety. The risk of an overloaded truck driver being involved in an accident is higher than with a legally loaded truck. Moreover, the involvement of overweight vehicles in road accidents increases the severity of accidents [89]. Video image processing techniques and RADAR sensors (or sets of different sensors including image capture connected to a traffic signal controller) are often used to detect or predict traffic violations. For instance, when a vehicle exceeds the speed limit, as well as when a vehicle crosses the stop line at a junction or at a pedestrian crossing when red signals are displayed [90,91]. Information on the scale of the violations may be taken into account when assessing the safety of selected elements of the road system. The data collected by the sensors described above may complement the standard traffic modelling data. However, these are most often data collected at specific points on the road without recording the trajectory of vehicles and changes in the dynamics of their movement on the road section. The usefulness of such data in traffic modelling for SSM analysis along road sections is limited.

Roadside sensors are used in traffic safety management to collect data on the travel time of vehicles on a given road section. Automatic Number Plate Recognition (ANPR) cameras or Bluetooth and Wi-Fi scanners located at the beginning and the end of a road section enable the identification of an individual vehicle (in the case of ANPR cameras by the vehicle registration number, in the case of Bluetooth/Wi-Fi scanners by the Media Access Control (MAC) address—MAC number of the electronic device) and calculation of its travel time based on recorded time stamps. The main element of the data processing module is algorithms analysing the collected data to detect incidents on the road section (different techniques are applied to deal with erroneous or missing values, e.g., time series analysis, Kalman and particle filtering, neural networks, fuzzy logic). The incident detection algorithm searches for changes in the length of travel time between measuring points. In case of a sudden and unjustified change, a notification is sent to the traffic management system. The immediate detection of an incident results in a reduction in the time needed for emergency services to help the victim and the early

activation of traffic control strategies (warnings displayed on the VMSs about the incident occurrence, detours, variable speed limits or road closures) to reduce traffic disruptions. A study based on the simulation of reactive VSL systems shows that the accuracy of information from the sensor stations, prediction of traffic conditions, estimation of time and place of the incident and the extent of the impact of the incident on traffic conditions are essential for the performance of a VSL system [92]. The data collected by the ANPR cameras and Bluetooth/Wi-Fi sensors can be used to calibrate and validate macroscopic, mesoscopic and microscopic traffic models in terms of travel time on the road sections and also for the calibration of time-dependent Origin–Destination (OD) travel matrices [93]. Incident detection is also possible by using video image processing methods or monitoring volume, speed and occupancy variations. It can be done by sensors located upstream and downstream of the incident (traffic measurement stations including inductive loops are most often used for this purpose) [94]. Data collected from sensors may also contain information about other road network disturbances (weather and state of pavement conditions, unexpected demands). Such data, if available, may be used for traffic modelling in the circumstances of a traffic incident occurrence.

Nowadays, an intensified development of Cooperative Intelligent Transportation System (C-ITS) services is observed. C-ITS services enable information exchange between vehicles (Vehicle To Vehicle—V2V) or vehicle and infrastructure (Vehicle To Infrastructure—V2I). Operating transport management systems are mostly not prepared to use Floating Car Data (FCD) or exchange data with vehicles. It is necessary to indicate the direction of development of these systems and to verify their architecture. Technological developments are therefore giving rise to integrated data sources that should be able to be used in research. The challenge is to process big data and merge them to use them in traffic modelling and safety assessment using SSMs. Nowadays, sensor data in vehicles and VIP and RADAR data are the best solution for traffic modelling and vehicle trajectory studies in the SSAM. Further development of methods based on the fusion of data from sensors located in the road environment and FCD data is required [95,96]. ANPR and Bluetooth/Wi-Fi sensor data are useful for travel distribution modelling and traffic model calibration or validation and can be complemented or replaced by mobile phones or electronic devices in vehicle location data.

Driving simulators are useful research tools in case of difficulties in obtaining data from mobile sensors or sensors located in the road environment. High-end simulators enable the recording of about 60 parameters related to the simulation, the location of the vehicle and its control mechanisms, and the quantities characterising the vehicle's interaction with the environment. The simulator is a realistic simulation environment allowing for driver behaviour assessment in terms of road safety. Therefore, it is a laboratory research tool constituting a multisensory stand-in for a drive [97]. It enables directly recording parameters such as the steering angle and the degree of brake and acceleration pedal pressing. It also enables the recording of values necessary to calculate parameters such as the distance to the vehicle in front and the vehicle position vector in three dimensions with a timestamp. Another calculated parameter used to measure drivers distraction is vehicle position (lane-keeping). Simulators are useful in determining the dynamics of speed changes along the road and can support traffic modelling and SSM estimation.

Sensors play a key role in the collection of data for the improvement of services related to road traffic safety as well as data necessary for scientific research supporting the development of ITS services. It is important to make use of the fusion of data from many available sources (including sensors located in the vehicle and the road environment as well as mobile devices or systems) [98–102].

It is, therefore, reasonable to ask how the driver will react when further information is displayed on variable message signs, and the driver is involved in the analysis of the presented content. Then, as in the experiment presented in [67], will the lateral deviation increase? Will the level of traffic safety measured with SSMs decrease because of this? What will be the impact of speed harmonisation and speed reduction recommendations on traffic safety? This paper presents the possibilities of using data from various sensors to develop simulation models, which allowed for an attempt to answer the above research questions.

3. Methodology and Selected Results of Research

The methodology and results of research presented in Section 3 are based on the use of research tools such as traffic models (macroscopic, mesoscopic and microscopic) and driving simulators. Section 3.1 presents the process of developing test road network models and travel models that are developed based on real data. Moreover, the characteristics of the mesoscopic dynamic traffic model and the process of calibration of the microscopic model (including the calibration of the car-following model) using data from the traffic measurement stations are presented. Scenarios studied in the RID-4D project are also described, as well as scenarios that are used as a basis for the development of VSL models. Section 3.2 presents the process of tests using a driving simulator. The results of the research are used to further calibrate the microscopic traffic model in terms of drivers' behaviour (changes in speed and dynamics of these changes) along the road section where VMSs are located. In the studies conducted with the use of a driving simulator, the influence of information displayed on the VMSs on lane-keeping by the driver is additionally identified. Section 3.3.1 presents the results of studies on the impact of VSL application (taking into account the location of VMSs) on road safety and traffic conditions. The research was conducted using a microscopic traffic model and the SSAM. In Section 3.3.2 the studies of additional scenarios of VSL impact are presented. Scenarios assumed the occurrence of different types of incidents in the road section (taking into account different incident duration and the scale of capacity limitation of the major road).

3.1. Development and Calibration of the Microscopic Test Models

The studies using microscopic models were part of research in which a multilevel approach was applied [9,103]. In the multilevel approach, macroscopic models (PTV VISUM software) [104] were used at the first stage of studies to obtain typical traffic distribution data in the road network with the use of the National Traffic Model. Detailed research on the influence of selected ITS service implementation on the road safety and traffic efficiency was conducted using mesoscopic (SATURN software) [105] and microscopic models (PTV VISSIM) [106]. Moreover, in the first stage of the study, available raw data from traffic measurement stations on motorways and expressways were collected. Inductive loops are the main elements of traffic measurement stations. Loops are located in selected road cross sections on each lane (in Poland, in the case of rural roads, mainly on motorways and expressways). The layout of two loops one after another on each lane enables measuring instantaneous speed and the classification of vehicles based on their length. Data from the traffic measurement stations provided an essential basis for the calibration and validation of macroscopic (traffic volumes, vehicle classification, average speeds), mesoscopic (traffic volumes, average speeds, vehicle classification) and microscopic (traffic volumes, speed values, vehicle classification, time headways between vehicles) models. The large number of available data allowed for the selection of a data set for model calibration and a control data set for model validation (including test network models and control models of the real road network). Models of the real road network were used to validate the adopted methodology in terms of traffic conditions on the road network in the case of typical traffic conditions and the circumstances in which the traffic incident occurred.

ViaToll system data were also collected to determine the routes of vehicles. The primary task of the ViaToll system is electronic toll collection on national roads. Vehicles with a maximum permissible weight of more than 3.5 tonnes and buses regardless of the maximum permissible weight are subject to payment. Other vehicles can voluntarily join the system and pay the charge on toll motorways. The basic elements of the ViaToll system are devices equipped with Dedicated Short-Range Communication (DSRC) readers, which recognise the passing vehicles. Some of the gantries are equipped with laser sensors or ANPR cameras and they collect data on all passing vehicles. Other gantries allow data collection only on vehicles with an on-board unit (ViaBox for heavy vehicles and ViaAuto for other vehicles). The system classifies vehicles into eight categories. The system does not measure vehicle speed. Data from these measurement points are aggregated to an hourly interval, for each lane. There are currently more than 700 gantries with DSRC readers on national roads. DSRC operates based

on separate short-range radio communication in the 5.8 GHz band. Data from 30 stations equipped with laser readers were used to calibrate the National Traffic Model (macroscopic model). ViaToll data were also used to determine the percentage of vehicles that were leaving the motorway or expressway after the incident occurrence (depending on the length of the incident). In the dynamic mesoscopic model, the process of route selection by drivers in subsequent periods of the incident was reproduced, taking into account the traffic volume.

In the next stage, the topology of the road network in the corridors of Polish motorways or expressways was analysed. The analysis took into account the structure and traffic alignment within road interchanges, distances between nodes and characteristics of alternative routes located in the major road corridor. Based on the collected data and the research carried out, test models of the road network were developed for selected road classes (expressways: S 2/2 and motorways: A 2/2, A 2/3) [34]. According to the Polish road classification, both motorways and expressways are designed and built for international or national motor traffic over longer distances. These roads are intended exclusively for motor vehicle traffic. The main difference between the discussed road classes is the maximum permitted speed (140 km/h on motorways and 120 km/h on dual carriageway expressways). Different permitted speeds and related design speeds determine the differences in the geometric parameters of the roads. These include, among others, the width of lanes, horizontal and vertical curves, roadway inclinations, width and presence of the emergency stopping lane, geometric solutions and parameters within road interchanges. The second important difference between motorways and expressways is their accessibility. The minimum permitted distance between interchanges on motorways is 15 km (within large cities, 5 km) and on expressways, 5 km (within cities, 3 km). In the test network models, the calculated average distance between interchanges along major roads based on the actual road network topology for the A 2/2 motorway was 15 km and for the S 2/2 expressway, 10 km. The length of roads in the test network for the A 2/2 motorway corridor was about 120 km, including 45 km of the major road, while in the S 2/2 expressway corridor, it was 74 km, including 30 km of the major road.

Traffic simulations with the use of the test network made it possible to conduct studies on the impact of the location of VMSs with speed limits displayed on them on the level of road traffic efficiency and safety. An example of a test network model for the S 2/2 expressway is shown in Figure 1 [34].

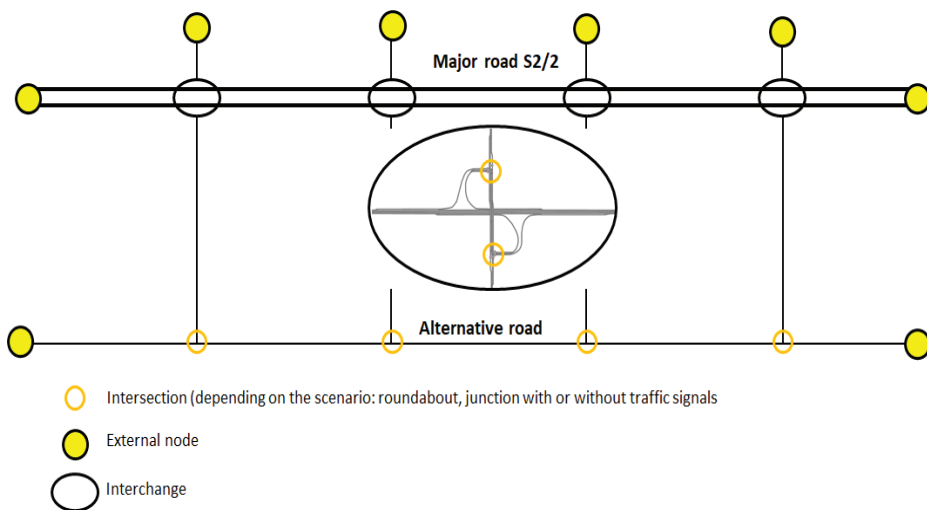


Figure 1. Test network for S2/2 expressway corridor.

Selected scenarios of the road network and traffic intensity were analysed, taking into account the occurrence of incidents on the major road to determine the impact of the use of ITS services on road safety and traffic efficiency. In addition to the road network topology, different types of interchanges and junctions along the major road and alternative routes were considered and defined in the simulated scenarios. The occurrence of an incident on the road resulting in blocking one or two lanes on the main road was assumed during the development of test models for selected scenarios. The analyses also took into account cases where the incident did not cause lane blocking. A reduction in road capacity during a simulated incident was adopted in mesoscopic models and, in the next step, in microscopic models. In addition to capacity changes due to traffic distribution, the rubbernecking phenomenon was taken into account in capacity limitation based on research [107,108]. The results of traffic assignment in mesoscopic test road networks were used to develop microscopic models.

Mesoscopic models of the test road network allowed us to proceed with traffic assignment, taking into account variable traffic conditions and capacity limitation resulting from queues, delays and the stopping of vehicles in traffic flow at junctions, road interchanges and individual road sections. A quasi-dynamic model was applied during the stochastic traffic assignment process to obtain more reliable results. Over-capacity queues were moved to subsequent defined periods by using the model (the simulations were divided into 30-min periods). The application of a quasi-dynamic model allowed us to take into account the dynamics of changes in traffic conditions in the face of the temporary blocking of the road by incident occurrence [9]. The mesoscopic model was used mainly to analyse the impact of incident management, while the microscopic model (fed by the traffic distribution in the road network and traffic volume data from the mesoscopic model) was used for the studies of Intelligent Transport System (ITS) services. This included scenarios involving providing the drivers with information through content displayed on VMSs. The selected scenarios were defined based on Regional Fire Departments' databases with data on the duration, location and type of incidents. The share of vehicles choosing an alternative route in the periods when incidents occurred was estimated based on the data from the ViaToll system. The data from the ViaToll system enabled determining the routes of the vehicles in selected areas of the road network under conditions of different incident duration compared to the traffic distribution in the network under non-incident conditions. It was done based on the identification of the vehicles and the time stamp of their appearance at subsequent ViaToll measurement points. The collected data were used to calibrate and validate the test network models. Independently, mesoscopic and microscopic models of selected corridors of the real road network (sections of A1 motorway and S6 expressway corridors) were developed. The comparison of real and model results allowed for a positive validation of the adopted models and the methodology of their development (traffic distribution in the network under the conditions of incident occurrence, traffic volume values and speed on selected sections of the network were compared). The data on the duration of incidents, their location and the scale of disturbances (closing of the entire roadway, the closing of one lane, etc.) were obtained from the reports of the Regional Fire Departments, which complemented the data collected by the ViaToll system and traffic measurement stations.

The representative hourly traffic volume in the different scenarios was classified into cohorts (Table 1) [9]. The classification of cohorts was made based on traffic data collected by the traffic measurement stations. The large amount of data collected by the sensors and data variations in terms of traffic volume values and circumstances (typical states, conditions with the occurrence of an incident) allowed us to develop traffic models for various conditions in the road network. The purpose of adopting cohort sets was to determine the frequency of occurrence of selected types of incidents causing different traffic limitations (number of lanes blocked) on particular road classes during the total road operation.

Table 1. Cohorts defined for traffic intensity scenarios.

| Cohort | Traffic Volumes in the Cohort q (veh/h/lane) | Volume-to-Capacity Ratio q/C | The Intensity of Traffic Assumed for the Load of the Major Road Lane in Test Models | Representative Volume-to-Capacity Ratio q/C |
|--------|---|------------------------------|---|---|
| 0 | q > 2100 | 0.95–1 | 2150 | 0.98 |
| 1 | 1300–2099 | 0.59–0.95 | 1700 | 0.77 |
| 2 | 720–1299 | 0.33–0.59 | 1010 | 0.46 |
| 3 | 0–719 | 0–0.33 | 360 | 0.16 |

The calibration of the microscopic models was carried out, taking into account time headways between vehicles and vehicle speed distribution. The data from traffic measurement stations (separately from selected expressways and motorways) and data obtained from tests with a driving simulator were used for calibration. Raw data (vehicle after vehicle) obtained from the traffic measurement station were used to determine the probability distribution (empirical cumulative distribution functions) of the choice of speed and time headways by drivers of different classes of vehicles (passenger cars and delivery vans as well as HGVs and buses).

The speed at which the driver is not influenced by other road users is defined as the desired speed. Desired speed was a variable used to calibrate the microscopic model. Traffic conflicts may occur when a vehicle interacts with slower-moving or queued vehicles during sudden braking or lane changing. The determination of speed and time headway distribution was developed based on data from traffic measurement stations for randomly selected days from different seasons. A comparison of the empirical desired speed distribution with the default VISSIM distribution for heavy vehicles and buses on the expressway is shown in Figure 2. The basic assumption is that for vehicles travelling slower than the displayed limit, the speed limit on VMSs will not affect these vehicles. The actual impact of VMS devices will be visible for vehicles travelling faster than the permitted speed. Information on the variable sign for what reason the speed is limited makes drivers more willing to adapt to it. Desired speed distribution functions were modified in such a way that the percentage below the speed on the VMS is identical to the initial one. On the other hand, the modification above the value indicated by the VMS included achieving speed values similar to those observed in reality, e.g., quantile 85 for a speed 20 km/h higher than allowed. Based on this, the distribution functions for the 80 km/h and 100 km/h limits were developed and implemented in the model. The modelling of the speed management service was performed using COMInterface. It is one of the VISSIM modules that allows the user to develop scripts that execute commands during the simulation that affect model elements and driver behaviour. An algorithm was developed to modify speed limit values on VMSs on the major road. During the simulation, the algorithm collected traffic volume data from the virtual sensors located in VMS areas and decided to change the speed limit every 5 min.

Virtual sensors correspond to inductive loops or other sensors detecting vehicle appearance and collecting basic traffic data. Virtual sensors performed two tasks in simulation models. The first task was to start the traffic control algorithm (displaying the corresponding speed limit on the VMS) in case of exceeding the defined traffic volume threshold. The second task of the virtual sensors was to monitor indicators of model calibration and validation (traffic volumes, vehicle speed values, time headways). An important element was the location of the virtual sensors in the test network model. Placing the virtual sensor too close to the VMS could cause the algorithm to indicate the conditions for changing the displayed speed with a delay. Such a situation may affect drivers' behaviour in terms of speed and distance to the next vehicle. The location of sensors is one of the key issues and should be thoroughly studied in future research work.

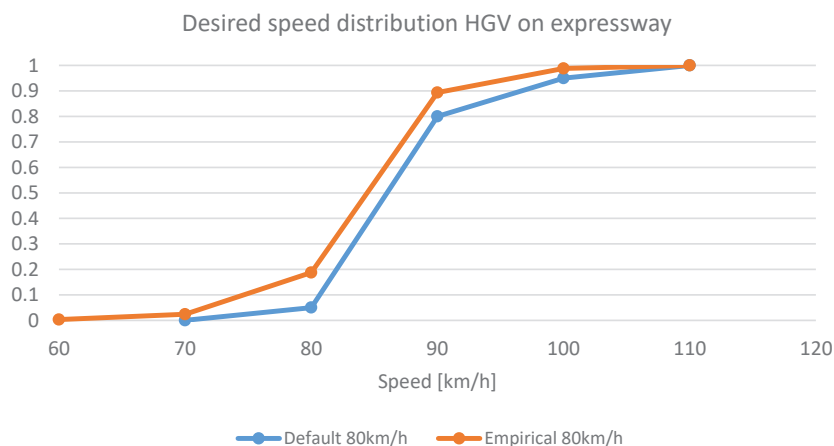


Figure 2. Comparison of the default VISSIM desired speed distribution and the empirical distribution for heavy goods vehicles (HGVs) and buses on the expressway.

For the study, the following boundary conditions were adopted to change the speed limit value on VMSs on the expressway:

- 120 km/h \leq 1000 veh/h/lane,
- 100 km/h $>$ 1000 veh/h/lane,
- 80 km/h $>$ 1550 veh/h/lane (if the VMS series was used, a speed limit of 100 km/h was displayed on the first VMS after the interchange and a limit of 80 km/h on the subsequent VMS. If one sign was located on the road section between the interchanges, a speed limit of 100 km/h was displayed on it, which followed the applicable regulations).

Another variable that was used for the model calibration was following time headway. This variable largely reflects driver behaviour, affecting road safety and capacity. The presence of a too short time headway between vehicles increases the risk of an incident in case of specific driver behaviour. Time headway is one of the variables in the Wiedemann 99 car-following model [106]. Researchers usually calibrate the models by setting a single variable value, resulting in a less realistic simulation of drivers' behaviour [109]. Time headway empirical data with a duration of less than 10 s (the limit above which free-flow traffic conditions occur) were used to develop cumulative distribution functions and to calibrate test models for representative traffic volumes in each scenario. Selected time headway cumulative distribution functions for different traffic volumes are shown in Figure 3 [34].

In the process of the calibration and validation of the model, data sets were used which were extracted for particular road classes and traffic volumes that were representative of the cohorts presented in Table 1. Data on the speed of the vehicles and the time headways between them were selected for each representative value of the traffic intensity (Table 1). The validation process used control data sets from different periods or other traffic measurement stations than the data used in the calibration process. The model validation related only to the solutions presented in the baseline scenario (without a VSL system) due to difficulties in finding a test field on Polish roads (no solutions with operating VSL systems or no data in the vicinity of the operating of a single VMS).

In the first step, the functions of the time headway distribution for individual road classes and representative traffic volumes in ten 1-s duration intervals (ranging from 0 to 10 s) were defined. In the validation process, the distributions of the sample frequencies over the intervals for the data set to develop the traffic model and control data set (percentage of the number of vehicles in each interval) were compared. No differences between the distributions of more than 12% (in most cases up to 8%) were noted for the individual intervals, which was considered a satisfactory result. The resulting

cumulative distribution functions contributed to the simulation traffic models in the calibration process and were not subject to further changes.

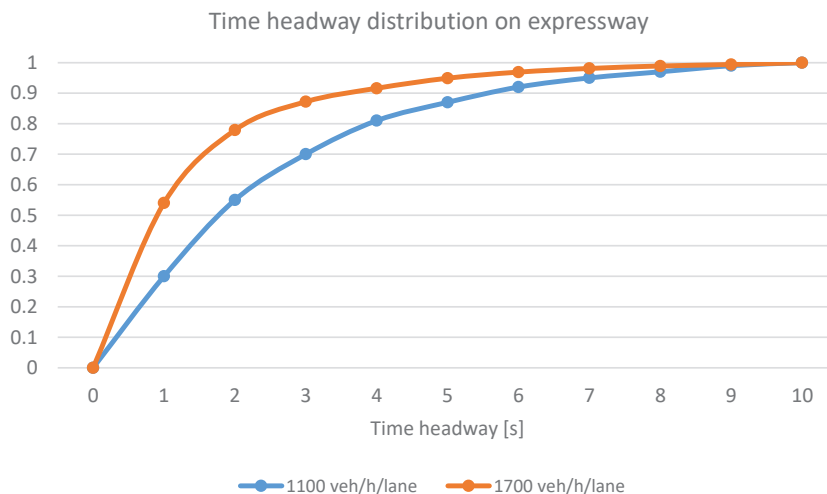


Figure 3. Time headway distribution on an expressway for 1010 veh/h/lane and 1700 veh/h/lane.

Traffic models have been developed for representative traffic intensity values (Table 1). During the calibration process, it was assumed that the traffic intensity values from the models would not differ by more than 6% when concerning the representative values. The above requirement was fulfilled.

Before starting to calibrate the model, the speed distribution functions for the different road classes, representative traffic volumes and vehicle categories were defined at 10 km/h intervals (in the range of up to 150 km/h). During the calibration process, the distributions of sample frequencies in individual intervals for the data set to develop the traffic model and the results from the model (percentage of the number of vehicles in each interval) were compared. No differences between the distributions of more than 16% (most often up to 13%) were observed for individual intervals. The most significant differences were observed in the speed intervals above 90 km/h, and in the other intervals for higher traffic volumes. The Mean Absolute Percentage Error (MAPE) for mean speeds in road cross sections calculated from observed data and the modelling results did not exceed 6% for passenger vehicles and 4% for heavy goods vehicles (values from the upper error range were observed for higher traffic volumes). The results were found to be satisfactory.

The validation process compared the distributions of sample frequencies in individual speed intervals for the model data set and the control observed data set. No differences in distributions exceeding 27% (most often up to 18%) were observed for individual intervals. The most significant differences were observed in the speed intervals above 80 km/h. The MAPE for the mean speed in road cross sections calculated from the observed data control group and the model results did not exceed 12% for passenger vehicles and 6% for heavy goods vehicles (values from the upper error range were observed for higher traffic volumes).

The results of driving simulator tests were used to calibrate models in VSL scenarios. Data enabled the identification of areas of speed change by drivers in response to a message on VMSs [97]. Due to the lack of field test sites, it was not possible to validate them. The conclusions of the paper indicate plans to carry out verification using data from field measurements. The capability of the driving simulator is limited in terms of triggering linear and angular accelerations and also due to imperfect movement patterns of the vehicle and its components (e.g., tyre/wheel model). The screen is located at a short distance from the driver's eyes and focuses his or her eyes on a different point than when driving.

Despite these disadvantages, driving simulators are commonly used as a testing tool, as they enable simulation of driving in repeatable, safe conditions. The screen is placed approximately 3 m from the driver's eyes in the AS 1200-6 simulator to minimise its drawbacks. It was also necessary to calibrate it according to the perception of the driver. The AS 1200-6 driving simulator was calibrated according to the individual perceptions of professional drivers with extensive driving experience. Although it was not possible to validate the results from the driving simulator, the data used enabled the development of comparable simulation conditions in the simulation scenarios.

3.2. Driving Simulator Study Results

The results of studies carried out using a driving simulator were also used to calibrate the microscopic test model. The data obtained from the studies, with the use of the high-end AS 1200-6 driving simulator, made it possible to take into account the behaviour of road users and to calibrate the microscopic model in terms of changes in the speed of vehicles before and behind the VMS location [97].

In studies using the driving simulator, the research group consisted of 60 people. The conditions to participate in the studies were to have a valid driving licence and to drive at least 2000 km per year. Participants were divided into three age groups: 18–24, 25–49 and over 50. There were 20 persons in each age group. The set of research scenarios involved conducting a simulation of driving past a sign or a VMS.

As part of the selected scenarios, six gate support structures (so-called gantries) were placed, on which graphic symbols corresponding to the signs and the VMS, were presented (Figure 4). Speed limits, information on traffic incidents and information on the weather conditions and alternative routes, as well as instructions for the drivers (e.g., about the need for lane changes), were presented.



Figure 4. Simulated road environment (a) and high-level driving simulator (b).

The purpose of the subsequent series of research experiments was to assess the impact of services providing information to the drivers on their behaviour. During the experiments, several dozen parameters related to the simulation and the vehicle itself moving in a virtual environment were recorded. One of these parameters was SDLP, mentioned earlier, as well as outputs on change of speed in the section before and behind the location of the VMS and the range of speed changes, taking into account the drivers' different reactions to the information about the speed limit that was displayed (outputs were used in the process of microscopic traffic model calibration).

The data were recorded for two weather conditions (precipitation and no precipitation), reflecting actual road conditions in Poland (the number of days with precipitation is approximately equal to the number of days without precipitation). Based on the deviation of the longitudinal axis of the vehicle from the lane axis (lateral deviation), one can draw conclusions about the additional load on the driver and the impact of the content presented on the value of this deviation. The lateral deviation is an objective measure of the driver's efficiency and makes it possible to make conclusions

about the distraction of the driver's attention as a result of additional tasks [65]. Such a task may be observing signposting in the form of VMSs. Although they can be regarded as an element of the road infrastructure, and their observation as a driving activity, in the case analysed in this article, this marking was considered as an additional element of the road signposting, which may or may not have to be used as a tool to inform drivers about the traffic situation. Thanks to this, it is possible to assess their impact on lane-keeping, and thus on the road safety itself. To evaluate the impact of the information provided through ITS services, lane-keeping was compared in pairs over a distance of 400 m, i.e., 200 m before the gantry support structure and 200 m behind it. The comparison was made for the baseline scenarios (without the impact of ITS services) and with such services introduced. The comparison of pairs of lane-keeping data under individual research scenarios is shown in Figure 5. The figure shows the number of samples (the deviation of the longitudinal axis of the vehicle with a frequency of 50 Hz was recorded) in individual deviation intervals. Negative values in the given intervals indicate an increased deviation from the right edge of the lane. Scenarios with the introduction of a speed limit with the use of a road sign (S0/G5), VMSs (S1/G2) and VMSs with additional information about the cause of the limit, which was a slippery road surface (S2/G5), were considered.

For the cases presented in Figure 5, the average value of the distance between the longitudinal axis of the vehicle from the lane axis and standard deviation of this value was calculated (Table 2).

Comparing the effect on the driver of a static (S0/G5) and dynamic (S1/G2) speed limit (Figure 5a), an increase in the average lateral deviation was noted from 6.3 cm, in the case of a static sign, to 13.4 cm in the case of a VMS. The standard deviation also increased from 30.7 cm to 33.2 cm. A possible reason for this is the recent legal regulation of respecting the indications of variable road signs compared to traditional solutions, well known to drivers. This indicates that they are more responsive to traditional road signs and, in this case, the considered impact of ITS on lane-keeping did not bring the expected effect. However, the differences were negligible. Comparing the static speed limit (S0/G5) with the VMS (S2/G5) (Figure 5b) containing the reason for the speed limit displayed, it had a high impact on drivers' speed. In the case of lane-keeping, the position deviation from the lane axis increased insignificantly from 6.3 cm to 9 cm and the standard deviation also slightly increased from 30.7 cm to 31.8 cm. Comparing the speed limit displayed on the VMS (S1/G2) with the same limit supplemented by the reason for the limit (S2/G5) (Figure 5c), an increase in lane-keeping was observed from 13.4 cm to 9 cm. The standard deviation decreased, from 33.2 to 31.8. This is an indication to apply solutions for imposing the speed limit, supplemented with information about the reason for the limit.

Table 2. Average value and the standard deviation of the distance between vehicle axis and lane axis.

| No. | Scenario | Average Value of the Position Deviation (m) | Standard Deviation of Lateral Position (SDLP) (m) |
|-----|---|---|---|
| 1. | Static limit (S0/G5) | −0.063 | 0.307 |
| 2. | Speed limit on the Variable Message Sign (VMS) (S1/G2) | −0.134 | 0.332 |
| 3. | Speed limit on VMS with the reason for limitation (S2/G5) | −0.090 | 0.318 |

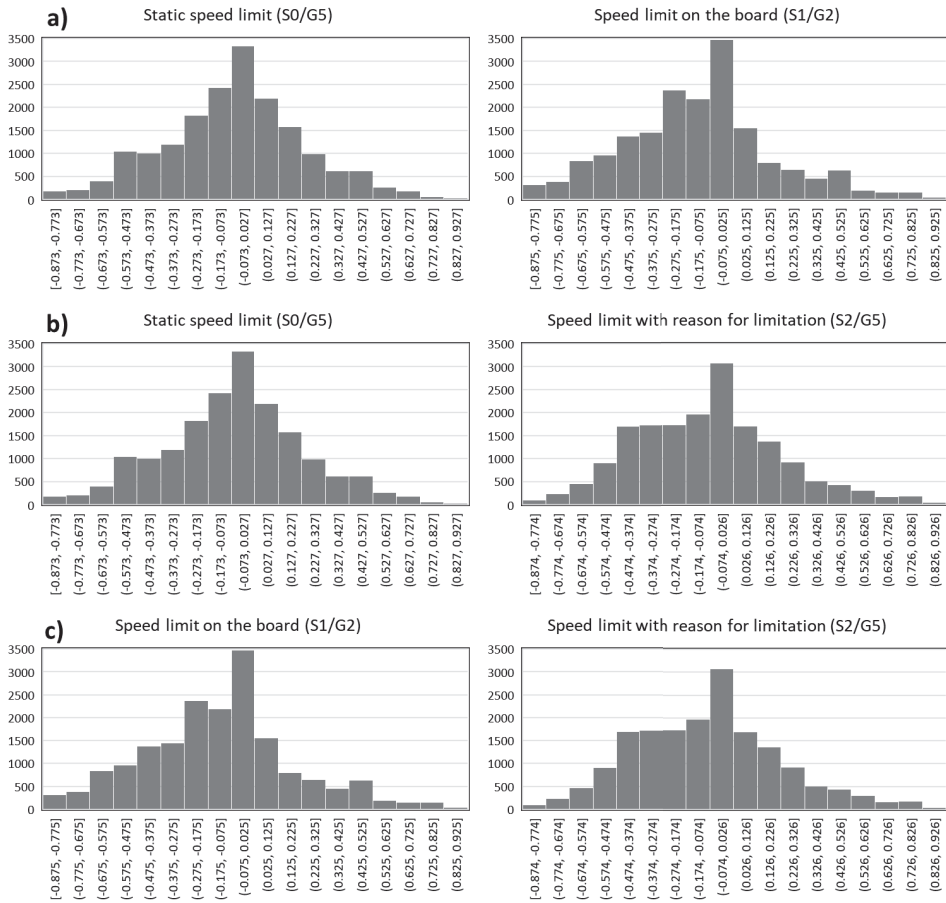


Figure 5. Comparison of pairs of the lane-keeping data under individual research scenarios.

3.3. Microscopic Modelling Results

3.3.1. Location of Variable Message Signs

The analysis of the state of the existing location of VMS devices on Polish motorways and expressways showed that the main arrangement is that with a single device between interchanges. Only on a few motorways (the A8 road is an example) are the devices located in series between interchanges. As part of the research, scenarios of VMS locations between interchanges were analysed. This paper presents the simulation results of three scenarios of VMS location on the expressway (S 2/2) with a traffic volume of 1700 vehicles per lane (the presented scenarios assume the use of junctions with traffic signals within interchanges on the major road and within the alternative route, as shown in Figure 1):

- W0—a baseline scenario—no service (VMS),
- W1—one VMS between interchanges,
- W2—VMSs between interchanges placed every two kilometres.

The impact of implementing speed control on road safety as measured by the number of traffic conflicts is presented in Table 3. The impact of introducing speed control on traffic efficiency measured

by traffic conditions results from simulation scenarios on the test road network, including the major road and the alternative routes (expressway corridor), is presented in Table 4.

The Surrogate Safety Assessment Model (SSAM) was used to calculate the number and severity of traffic conflicts [51]. Two surrogate safety measures were used to assess the level of road safety in selected simulation scenarios: Time To Collision (TTC) and MaxDeltaV, which is determined by the speed difference between vehicles involved in a traffic conflict. Movement trajectories for every interacting vehicle were extracted from the simulation environment of microscopic models and compared to identify the occurrence of conflicts and their characteristics.

A traffic conflict should be defined as the occurrence of an interaction between two vehicles which would lead to a collision if one of the vehicles did not change speed or direction. TTC is defined as the period during which a collision would occur if neither of the two vehicles changed speed or the direction of driving [110]. The simulation results were used to compare different types of conflicts (lane change and rear-end conflicts on the sections of the major road between interchanges were taken into account). Equation (1) can be used to calculate TTC. A minimum assumed TTC value (TTCmin between 0.1 s and 1.5 s) was adopted to define the traffic conflict occurrence.

$$TTC_i = \frac{X_{i-1}(t) - X_i(t) - l_i}{\dot{X}_i(t) - \dot{X}_{i-1}(t)} \quad \forall \dot{X}_i(t) > \dot{X}_{i-1}(t) \quad (1)$$

where:

X —vehicle position,

\dot{X} —vehicle speed,

i —vehicle following the leader,

$i-1$ —lead vehicle,

t —moment in time,

l —vehicle length.

The MaxDeltaV measure can be used indirectly to evaluate the severity of traffic conflict, which occurred when TTC was less than the minimum assumed value, i.e., when the conflict was identified. The relative severity of the traffic conflict can be determined by comparing the trajectories of two vehicles in terms of direction and speed at the moment when TTCmin occurs. The severity of the conflict is higher in the case of greater speed differences between vehicles involved in traffic conflict and an unfavourable angle of a potential collision. The angle at which the trajectories of vehicles involved in traffic conflict cross is a criterion for assigning a conflict type. The SSAM defines three types of traffic conflicts: crossing, lane change and rear-end. It is assumed that the conflict is identified as severe in the case of a difference in speed of vehicles exceeding 20 km/h [47].

It should be noted that in the analysed scenarios, saturated traffic conditions (1700 veh/h/lane) on the major road and in the remaining road network were assumed, which led to lower traffic speed. The results for all scenarios (Table 3) showed differences in the distribution of specific types of conflicts (2–3% crossing, 82–84% rear-end and 13–16% lane change) on specific parts of the test road network (39–53% on the major road between interchange areas, 39–50% at interchanges along the major road and 8–11% on the rest of test road network) depending on the scenario. In terms of the severity of conflicts assessed based on MaxDeltaV in the individual scenarios, the highest percentage of conflicts was observed on the major road and in the area of road interchanges. Due to speed limits on alternative routes, the percentage share of severe conflicts was marginal. Conflicts of the lane change type prevailed (50–61% depending on the variant) and rear-end (25–29%). In saturated traffic conditions, greater differences in speed between vehicles in the case of conflict were more frequent.

Table 3. Results of simulations for traffic safety assessment in the individual scenarios.

| Part of Road Network | Type of Conflict | Measure | Scenario | | | Difference | |
|---|------------------|---------------------|----------|--------|------|------------|-------|
| | | | W0 | W1 | W2 | W1/W0 | W2/W0 |
| Main road sections without merging and weaving sections | crossing | Number of conflicts | 0 | 0 | 0 | 0% | 0% |
| | | MaxDeltaV > 20 km/h | 760 | 562 | 271 | -26% | -64% |
| | lane change | Number of conflicts | 1183 | 886 | 489 | -25% | -59% |
| | | MaxDeltaV > 20 km/h | 4899 | 4228 | 2969 | -12% | -39% |
| | rear end | Number of conflicts | 210 | 184 | 100 | -12% | -52% |
| | | MaxDeltaV > 20 km/h | 226 | 254 | 205 | 12% | -9% |
| Interchanges along major roads | crossing | Number of conflicts | 226 | 254 | 205 | 12% | -9% |
| | | MaxDeltaV > 20 km/h | 223 | 252 | 200 | 13% | -10% |
| | lane change | Number of conflicts | 699 | 675 | 657 | -3% | -6% |
| | | MaxDeltaV > 20 km/h | 230 | 194 | 234 | -16% | 2% |
| | rear end | Number of conflicts | 3604 | 3724 | 3500 | 3% | -3% |
| | | MaxDeltaV > 20 km/h | 192 | 188 | 191 | -2% | -1% |
| Other parts of the road network | crossing | Number of conflicts | 11 | 14 | 16 | 27% | 45% |
| | | MaxDeltaV > 20 km/h | 1 | 2 | 9 | 100% | 800% |
| | lane change | Number of conflicts | 3 | 12 | 15 | 300% | 400% |
| | | MaxDeltaV > 20 km/h | 0 | 1 | 0 | 100% | 0% |
| | rear end | Number of conflicts | 919 | 889 | 934 | -3% | 2% |
| | | MaxDeltaV > 20 km/h | 0 | 0 | 1 | 0% | 100% |
| Number of conflicts | | | 11,544 | 10,742 | 8785 | -7% | -24% |
| MaxDeltaV > 20 km/h | | | 1616 | 1383 | 1006 | -14% | -38% |

Table 4. Results of simulations for traffic efficiency assessment in individual scenarios.

| Measure | W0 | W1 | W2 | W1/W0 | W2/W0 |
|--|--------|--------|--------|-------|--------|
| Average delays (s/veh) | 85.35 | 82.98 | 73.46 | -2.8% | -13.9% |
| Average number of stops (stops/veh) | 0.42 | 0.43 | 0.40 | 3.0% | -4.8% |
| Mean speed (km/h) | 64.52 | 64.30 | 63.60 | -0.3% | -1.4% |
| Total delays in the entire network (h) | 38,140 | 37,180 | 33,010 | -2.5% | -13.5% |
| Total number of stops | 65,888 | 68,313 | 63,279 | 3.7% | -4.0% |

The effect of implementing variable message signs was most evident on the sections of expressways and motorways (this paper shows an example of research for an expressway with two lanes in each direction). The road sections studied did not include the area of interchanges with merging and weaving sections. As shown in Table 3, the most significant reduction in the number of conflicts that occurred in the W2 scenario (implementation of a VMS series along road sections). The reduction concerned both lane change conflicts (a decrease by 59% compared to the baseline scenario) and the most common rear-end conflicts (39%). The introduction of a service providing information about the recommended speed also resulted in a more significant reduction of conflict severity than in the case of their number. In the case of lane change conflicts, there was a reduction by 64%, for rear-end conflicts by 52%. The impact of speed reduction on the expressway sections was also noticeable within road interchanges. If a VMS was located closer to the interchange (in the case of scenario W2 with a series of variable message signs along the section), there was a reduction in the number of conflicts and their severity compared to the baseline scenario. Within the rest of the road network (alternative routes and routes connecting with the major road), there was a slight deterioration in the level of traffic safety measured in absolute values of the number of conflicts. In the whole road network under study, the introduction of the speed control service contributed to the reduction of the number of conflicts (by 24%) and their severity (by 38%).

The frequency of the TTC value occurrence in the scenarios of the localisation of variable message signs is presented in Figure 6. The figure shows an example of a TTC distribution for rear-end conflicts on sections of the major road. In scenarios involving providing the information on speed limits to the drivers, decreases in TTC were noted in all the considered intervals. The results showed a positive impact of the measures on the level of road safety. Conflicts for which TTC took values above 1 s were the most frequent. Similar distributions were obtained for lane change conflicts.

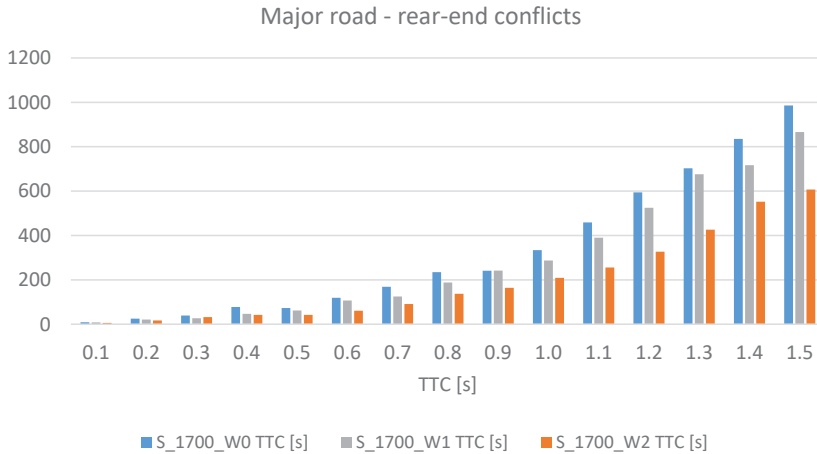


Figure 6. Distribution of occurrence frequencies of Time to Collision (TTC) values in individual scenarios.

A useful tool in the study of the conflict occurrence process is the analysis of their concentration. In the case of the considered scenarios of variable message sign location, different patterns of conflict concentration areas were observed (Figures 7 and 8). Figure 7 shows the areas of conflict concentration (conflict density represented by the size and the colour of the area, where a darker colour means a higher density). The distribution of individual conflicts along the major road section (rear-end type—yellow colour, lane change type—blue colour) is shown in Figure 8. For all scenarios considered, the first place where the conflicts accumulate is the first section of the major road, along the merging section at the place where traffic flows at different speeds: the point where entry traffic flow from the ramp and the expressway traffic flow meet. This contributes to the occurrence of lane change conflicts but also contributes to the formation of a shock wave, which is accompanied by rear-end conflicts and, in a smaller number, by lane change conflicts. In the absence of dynamic information about the speed limit (W0), conflicts arise evenly over the entire section between interchanges (Figure 8). The places where conflicts accumulate are irregular and, under saturated traffic conditions, are associated with the formation of short-term shock waves due to greater differences in vehicle speeds along the road section. The phenomena described above result in the highest number of conflicts and their highest severity (Table 3) among the scenarios under consideration. In a scenario where the use of a single VMS (W1) is taken into account, areas with an accumulation of conflicts appear at the place of impact of the VMS, in its vicinity (Figure 7), due to speed reduction by some drivers, which also contributes to the formation of shock waves. On the longer sections without VMSs, some drivers increase their speed, resulting in irregular areas of conflict accumulation similar to the W0 scenario (Figure 8). In the case of using variable speed limits with the use of a VMS series (W2 scenario), the concentration of conflicts was observed in the area of the first VMS on the road section directly behind the interchange (effect of merging traffic flows of different speeds) and before the second VMS in the series (Figure 7). In the remaining section of the road, a lower frequency of conflicts and fewer concentrated areas can be observed (Figures 7 and 8), which proves the harmonisation of speed thanks to the use of a VMS.

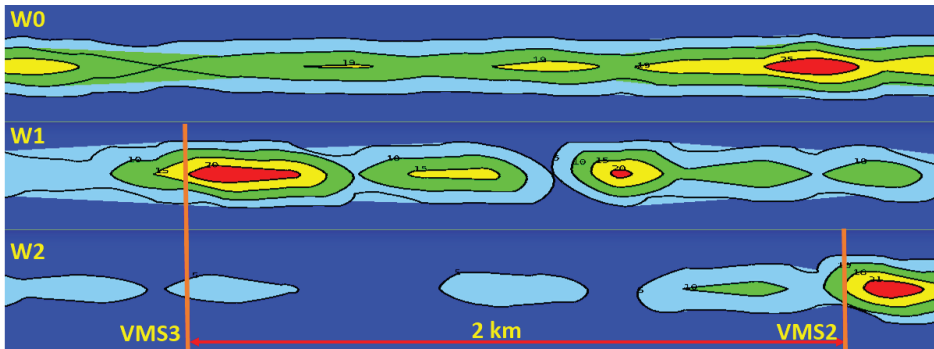


Figure 7. Areas of conflict concentration along the major road (the direction of traffic from left to right, conflict density represented by the size and the colour of the area, where a darker colour means a higher density).

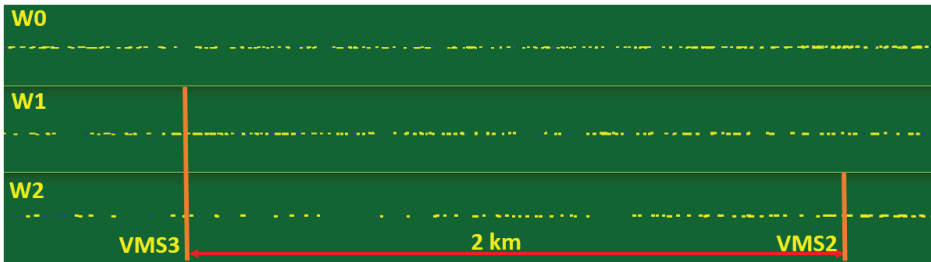


Figure 8. Distribution of individual conflicts along the major road section (yellow colour—rear-end type, blue colour—lane change type).

The simulation results presented in Table 4 show an improvement of the traffic conditions in the whole test network due to the introduction of speed limit information and the resulting traffic harmonisation by placing more VMSs on a road section.

3.3.2. Comparison of Scenarios Taking into Account the Occurrence of Incidents

Further comparisons took into account the lower level of traffic volume in the test road network and considered situations in which traffic incidents occurred. The research was carried out for the baseline scenario and scenario W2 assuming the location of a series of variable message signs in the section between interchanges (scenario W2 proved to be the most effective due to improved traffic conditions and road traffic safety). The traffic microsimulation for each scenario took 2.5 h.

This paper presents and compares the simulation results for two traffic intensity levels representing cohorts 1 (saturated traffic flow conditions) and 2 (free-flow traffic conditions) from Table 1. The results of simulations for the test road networks in the expressway corridor (S 2/2) are presented. Selected scenarios assume the occurrence of signal-controlled junctions within interchanges along the major road and on alternative routes.

The results of simulation scenarios assuming an incident in the major road middle section (scenarios W0b and W2b) are also presented. The simulated incident lasted 30 min and caused one lane to be blocked. The following are the characteristics of the selected scenarios in terms of incident occurrence and application of the ITS service, which informed drivers about the recommended speed via VMSs:

- W0—a baseline scenario—no ITS service (VMS),

- W2—VMSs between interchanges located every two kilometres,
- W0b—a baseline scenario—no ITS service (VMS), the occurrence of an incident,
- W2b—VMSs between interchanges located every two kilometres, the occurrence of an incident.

The impact of introducing speed control on traffic efficiency measured by traffic conditions from simulation scenarios on the test road network (Figure 1), including the major road and the alternative routes, is presented in Table 5.

Table 5. Traffic efficiency measures for the entire test network for the model with an expressway (S 2/2).

| Scenario | | Traffic Volume (veh/h/lane) | Average Delays (s/veh) | Average Number of Stops | Average Speed (km/h) | Total Delay (h) | Total Number of Stops |
|----------|------------------|-----------------------------|------------------------|-------------------------|----------------------|-----------------|-----------------------|
| W0 | Without incident | 1010 | 58.51 | 0.31 | 74.83 | 1563 | 28,988 |
| W2 | | | 55.31 | 0.31 | 73.17 | 1498 | 28,984 |
| W0 | | 1700 | 85.35 | 0.42 | 64.52 | 3814 | 65,888 |
| W2 | | | 73.46 | 0.40 | 63.60 | 3301 | 63,279 |
| W0 | With incident | 1010 | 67.84 | 0.42 | 70.86 | 1904 | 40,899 |
| W2 | | | 64.66 | 0.39 | 69.52 | 1838 | 38,512 |
| W0 | | 1700 | 91.90 | 0.68 | 62.07 | 4360 | 117,601 |
| W2 | | | 85.89 | 0.67 | 60.04 | 4168 | 117,738 |

The introduction of variable speed limits contributed to the improvement of traffic conditions, taking into account the whole road network located in the expressway corridor. The implementation of variable speed limits resulted in a decrease in total delays and the number of stops for both lower and higher traffic volumes. The improvement was more evident in the saturated flow conditions. Moreover, the occurrence of the incident resulted in a reduction of capacity on the major road. For this particular reason, the traffic assignment changed and congestion along alternative routes increased.

The introduction of ITS services related to speed control aims at improving the level of road safety by harmonising the speed of vehicles and reducing over-speeding both in normal traffic conditions and when an incident occurs. Traffic conditions deteriorated significantly both along the major road and alternative routes after an incident occurred in the road network. One of the main objectives of the implementation of ITS services is to reduce congestion and help to restore normal traffic conditions as quickly as possible after clearing a road lane blocked during an incident. Changes in average speed during the simulation in the selected scenarios of test models are shown in Figure 9 (scenario W1 with a single VMS on the road section between interchanges was included).

The occurrence of an incident between 60 and 90 min of simulation had a significant impact on the decrease in average speed in both the baseline scenario and the scenario with variable speed limit implementation. The average speed was lower in the scenario with the incident (W2b) compared to the baseline scenario (W0b). Still, the implementation of variable speed limits contributed to an increase in traffic volume along the major road to a value in the range near to its capacity, as well as improved the level of road safety (Table 6).

The results of the studies on the level of road safety using the SSAM are presented in Table 6. The presented results include the road network of the expressway corridor together with alternative routes. The positive impact of the variable speed limit on road safety is expressed in a reduction in the number of conflicts and their severity. It should be noted that there is a significant deterioration in the level of road safety in the case of increased traffic volumes in the studied road network. The implementation of the service providing information about speed limits on sections of expressways (excluding interchange areas) resulted in a reduction in the number of conflicts to 7% in the case of a lower traffic volume and to 24% in the case of saturated traffic conditions. The number of severe

conflicts was also significantly reduced, to 16% and 38%, respectively. In the case of scenarios where an incident was assumed to occur, there was also a reduction in the number of conflicts and severe conflicts, but to a lesser extent than in non-incident conditions. The results indicate a significant potential for introducing measures to harmonise vehicle speeds, resulting in an improved level of road safety.

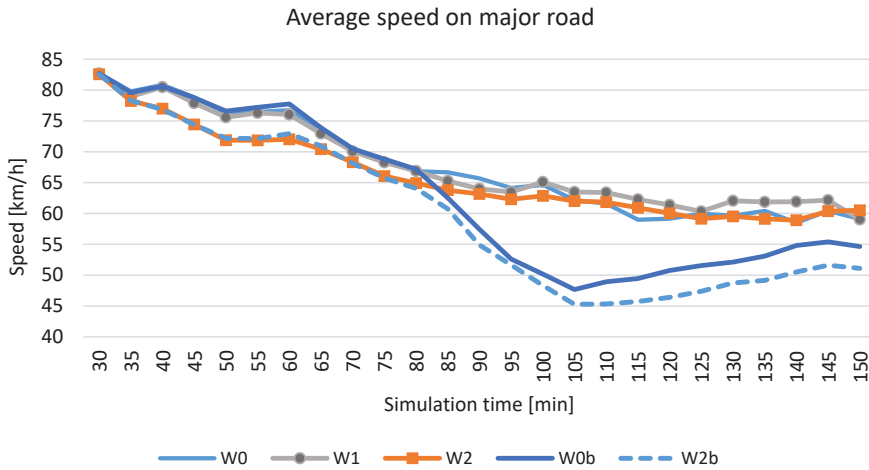


Figure 9. Average speed on the major road with traffic volume of 1700 veh/h/lane.

Table 6. Traffic safety measures for the entire test network for the test model with an expressway (S2/2).

| Scenario | | Traffic Volume (veh/h/lane) | Number of Conflicts | MaxDeltaV >20 km/h | Number of Conflicts | MaxDeltaV >20 km/h |
|----------|------------------|-----------------------------|---------------------|--------------------|---------------------|--------------------|
| | | | Entire Test Network | | Major Road | |
| W0 | Without incident | 1010 | 2079 | 446 | 984 | 303 |
| W2 | | | 1940 | 374 | 816 | 233 |
| W0 | | 1700 | 11,544 | 1616 | 6082 | 970 |
| W2 | | | 8785 | 1006 | 3458 | 371 |
| W0 | With incident | 1010 | 3208 | 507 | 2012 | 361 |
| W2 | | | 2978 | 428 | 1791 | 277 |
| W0 | | 1700 | 13,520 | 1511 | 8889 | 876 |
| W2 | | | 13,007 | 1082 | 8252 | 447 |

4. Discussing the Results and Conclusions

One of the main objectives of intelligent transportation systems solutions is to improve road safety. The implemented solutions should be used to provide drivers with information and influence their behaviour in such a way that the above postulate can be realised. The relevant questions are, therefore, how to study the impact of ITS solutions on road safety and how sensors can be used within the presented research methodology. This paper presents measures that can be useful in road traffic safety research and can complement or replace traditional statistical studies, especially in the case of innovative implementations for which field or statistical studies are difficult to conduct due to their rarity. The key basis for the research is the sensors and data collected by the sensors, which made it possible to develop, calibrate and validate traffic models and then estimate surrogate safety

measures. Surrogate safety measures enable us to study the effects of planned modifications to the existing ITS services and the search for optimal solutions to increase their effectiveness. Moreover, the research methodology presented in this paper allows for collecting data as a starting point for developing comprehensive models of ITS services' impact on road efficiency and safety. It allowed us to obtain data to power the model, enabling the estimation of ITS services' efficiency based on the Analytic Hierarchy Process (AHP) method and to estimate indicators for functional criteria for the AHP method [9,34]. There are no guidelines for introducing and designing VSL systems in Poland. There is also a lack of a method for assessing the effectiveness of the implemented solutions. The presented research methodology, which takes into account different levels of traffic intensity and occurrence of road incidents, may support the development of such guidelines. Field tests of the VSL service are also hampered due to the lack of testing sites in Poland. VSLs with a series of variable message signs have been used on some sections of motorways and expressways (e.g., on the A8 motorway or the S8 expressway). Usually, VMSs providing information about the recommended speed are installed on road sections between interchanges either singly (after passing an interchange) or in pairs (after passing the interchange and before the next). This practice makes it impossible to introduce an effective VSL system with a series of signs on road sections where this may be appropriate. The possibility of obtaining data on the periods when individual information is displayed on VMSs is very limited (such information is not collected in databases). Databases usually collect data from traffic measurement stations that are used to calibrate microscopic models. Traffic measurement stations, which include inductive loops, are most often used on Polish roads. Traffic measurement stations give the possibility to collect data only in selected points of the road system where the sensors are installed. Dynamic traffic safety management and traffic safety studies usually require the collection of traffic flow data from road sections. One of the ways to strengthen the detection systems and improve the quality of the collected data sets is to install more traffic measurement stations on road sections (increasing the possibility to estimate the traffic condition on road sections between the stations, based on the variability of traffic flow parameters). Another way to improve the quality of collected data is to use technologies that enable monitoring of the traffic flow condition and movement of individual vehicles along road sections. The most promising sensors that can be used in SSM-based research and current traffic safety management (e.g., for detecting incidents or dangerous driver behaviour) are RADAR sensors and video image processors (VIPs). The use of advanced algorithms in processing data from these sensors allows us to analyse the trajectory of individual vehicles on a given road section (vehicle position, dynamics of speed changes) and to detect interactions between vehicles. Such data were not available for the research presented in this paper. However, the impact of the information displayed on the VMS on driver behaviour required a detailed analysis of data on changes in driver behaviour (speed changes and the dynamics of these changes) along the road. Such analysis is currently possible only with the use of a driving simulator [97]. This approach made it possible to determine the areas where drivers decide to change speed as a result of seeing information on the VMS and further calibration of microscopic models. In future research works, we plan to take in-depth field measurements of vehicle trajectory and speed change dynamics in the vicinity of the VMS using RADAR sensors and video image processing techniques.

In the case of tests using a driving simulator, one measure of driver distraction may be lane-keeping information. The efficiency of lane-keeping by the driver (lateral control capability) can then be analysed. The literature reports point to two aspects of additional driver tasks, namely distraction, but also a possible increase in driver performance during a difficult task. The realisation of an additional task may be difficult in this case. The authors of the paper [69] concluded that the increase in cognitive load, which, although disruptive to the activities performed by the driver, causes an increase in lane-keeping performance. It seems that both cases described above can be observed as a result of experiments carried out under the RID-4D project. The histograms of lateral deviation values and histograms of standard deviation of this variable presented in this article may be used for preliminary evaluation of the analysed phenomena. The histograms present distributions of values close to the normal

distribution, indicating the existence of value intervals with the number of values much not greater than the number of adjacent intervals. For the analysed cases, the mean value of the lane deviation and the variance of this variable were also calculated. In the analysed cases, a positive impact of ITS solutions on vehicle lane-keeping was noted. It varies from case to case, but in combination with other aspects of the application of ITS solutions, such actions aimed at improving road safety should be positively evaluated. The research also allowed us to define the drivers' behaviours with the variable message signs, including the information provided by the signs. The results of the research conducted with the driving simulator were used to calibrate the microscopic models, examples of which are presented in this paper and are a promising source of data to develop models of driver behaviour.

This paper presents the methodology used for the preliminary assessment of the impact of a variable speed limit on major roads on the efficiency and safety of traffic on sections of expressways, taking into account the remaining elements of the test network model. The methodology is recommended to be used in planned implementations both to determine the effectiveness of planned solutions, their optimal location and the type of information provided to drivers, taking into account the traffic volume. The traffic safety assessment method is based on the analysis of surrogate traffic safety measures. In the presented case, TTC measures, representing the probability of accidents, and MaxDeltaV, determining their severity, were used. The above-mentioned measures are widely used in road safety analyses, however, to date the research has not been focused on such a broad approach as presented in this article, which takes into account not only the impact of measures on individual elements of the road network but also on the entire network of roads (major road corridor). In the presented studies, the Surrogate Safety Assessment Model (SSAM) was used for the first time to assess the safety of an ITS service, such as VSLs, with the use of surrogate safety measures. Added value is also found in the study and results of the location of variable message signs in the VSL system on a road section developed using the SSAM. Besides, an approach that takes into account the areas of conflict accumulation and conflict density in SSM modelling concerning traffic intensity seems promising.

Previous research on VSL systems [111,112], or co-operative VSL systems (C-VSLs) [113–115] used simulation traffic models but were mostly conducted to assess traffic conditions without considering road safety. Yu and Abdel-Aty [116] found a level of safety that increased with VSL techniques on steep mountain bottlenecks using VISSIM microscopic simulation software. They estimated and used a real-time crash risk assessment model to quantify the crash risk. The safety impact of the VSL system was quantified as decreasing the risk of a crash and improving speed homogeneity. The conclusions showed that the proposed VSL system could improve road traffic safety. The results of our research also confirm the positive impact of VSLs on the level of traffic safety. The decrease in the number of conflicts and their severity indicates homogenisation of vehicle speed (TTC and MaxDeltaV measures reflect changes in speed differences between vehicles). This is particularly noticeable when using a series of VMSs on the road (39% reduction in the number of conflicts on major road sections, 52% reduction in the severity of conflicts) for rear-end conflicts compared to the baseline scenario without using VSLs. An even greater safety improvement was observed in the case of lane change conflicts (59% and 64%, respectively). The reduction in the number of lane change conflicts and their severity also shows the positive impact of VSLs on the homogenisation of traffic in terms of speed (less tendency for drivers to change lanes after speed harmonisation). Moreover, in our research, simulation models were calibrated using the driving simulator data and also real data on speed and time headway distribution from the traffic measurement stations. Therefore, the tendency of drivers to exceed the allowed speed on Polish roads was taken into account. This is important in terms of the effectiveness of VSL systems, as the results of research [116,117] indicated that the VSL system fails to significantly enhance traffic safety under low driver compliance.

Many of the issues mentioned in this paper indicate the possibility of developing in-depth research. In future research on VSLs, it would make sense to compare SSMs with the actual occurrence of incidents and to analyse real data (vehicle trajectories) obtained through RADAR sensors or image processing techniques. It will be important to take into account the provision of information to drivers

by ways other than VMSs, e.g., in-vehicle information. Data from connected vehicles can be a new source of information for VSLs. Data on vehicle speed, distance to VMSs and position on the road can be used to improve VSL performance. If the number of connected vehicles is high enough, traffic management systems will not depend on detection stations and VMSs, which are expensive to maintain. Instead, traffic management systems could use V2V and V2I communication to provide drivers with information about individualised speed limits. The use of more data and the timely provision of information to the driver will improve traffic safety. It is important to make use of the fusion of data from many available sources (including sensors located in the vehicle and the road environment, as well as mobile devices or systems). It would also make sense to develop density analyses of different types of conflicts depending on the VSL solutions and to conduct studies on the impact of different speed control strategies and algorithms on road traffic safety on different types of roads. The research methodology presented in this paper and the developed research tools allow us to deepen the research in the areas mentioned above. The location of sensors collecting data for traffic control is also one of the key issues and should be thoroughly studied in future research work. The rational placement of sensors in the road network allows for continuous monitoring of traffic conditions and providing information to the driver at the right time. Providing information to the driver at an early stage reduces traffic disturbances and warns the driver about traffic hazards.

The problems presented in this article may provide advice on the development of sensor research to use the data for research on traffic safety management. Knowledge of what data are needed to manage and analyse traffic safety is essential to decide on the directions of sensor technology development. The presented studies have confirmed the positive impact of variable speed limits on the level of road safety and the efficiency of traffic. The impact may vary depending on the volume of traffic, the continuity of informing drivers about limits and the location of VMSs.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: J.O.; data collection/investigation: J.O. and K.Ž. (traffic modelling and simulations) and T.K. and M.P. (driving simulator studies); methodology/analysis and interpretation of results: J.O. and K.Ž. (traffic modelling and simulations) and T.K. and M.P. (driving simulator studies); draft manuscript preparation: J.O. and K.Ž.; critical review/commentary: J.O., K.K. and J.C.C. All authors reviewed the results and approved the final version of the manuscript.

Funding: The authors developed in this paper research that was carried out as part of the project called “The impact of the usage of Intelligent Transport System services on the level of road safety” (RID-4D) funded by the National Research and Development Centre and General Directorate for National Roads and Highways in Poland (agreement no. DZP/RID-I-41/7/NCBR/2016 from 26.02.2016).

Acknowledgments: The results will be presented at the 13th International Road Safety Conference GAMBIT 2020.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Randolph, L. Texas Variable Speed Limit Pilot Project. In Proceedings of the National Rural Intelligent Transport Systems Conference, San Antonio, TX, USA, 2015.
2. Fudala, N.J.; Fontaine, M.D. *Work Zone Variable Speed Limit Systems: Effectiveness and System Design Issues*; FHWA/VTRC 10-R20; Virginia Department of Transportation: Fairfax, VA, USA, 2010.
3. Garcia-Castro, A.; Monzon, A. Homogenization effects of variable speed limits, *transp. Telecommun. J.* **2014**, *15*, 130–143. [[CrossRef](#)]
4. Papageorgiou, M.; Kosmatopoulos, E.; Papamichail, I. Effects of variable speed limits on motorway traffic flow. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2047*, 37–48. [[CrossRef](#)]
5. Abdel-Aty, M.; Dilmore, J.; Dhindsa, A. Evaluation of variable speed limits for real-time freeway safety improvement. *Accid. Anal. Prev.* **2006**, *38*, 335–345. [[CrossRef](#)] [[PubMed](#)]
6. Li, Z.; Li, Y.; Liu, P.; Wang, W.; Xu, C. Development of a variable speed limit strategy to reduce secondary collision risks during inclement weathers. *Accid. Anal. Prev.* **2014**, *72*, 134–145. [[CrossRef](#)]

7. Polish Road Safety Observatory. Available online: <http://www.obserwatoriumbrd.pl> (accessed on 23 July 2020).
8. Zielińska, A. *Report on the Implementation of Task 2.1 within the Framework of the Project Entitled "Impact of the Usage of Intelligent Transport Systems Services on the Level of Road Safety" (RID-4D)*; Motor Transport Institute: Warsaw, Poland, 2018.
9. Oskarbski, J.; Gumińska, L.; Marcinkowski, T.; Mowiński, K.; Oskarbska, I.; Oskarbski, G.; Zawisza, M.; Żarski, K. Methodology of research on the impact of ITS services on the safety and efficiency of road traffic using transport models. *MATEC Web Conf.* **2018**, *231*, 02008. [[CrossRef](#)]
10. Dijkstra, E.; Bald, A.; Benz, S.; Gaitanidou, T. Overview of resulting tools, guidelines, and instruments. IN-SAFETY workpackage 3: New models, tools and guidelines for road safety assessment, deliverable 3.4. *SWOV* **2008**, *35*, 24.
11. Nilsson, G. *Traffic Safety Dimensions and the Power Model to Describe the Effect of Speed on Safety*. Ph.D. Thesis, Lund University, Lund, Sweden, 2004.
12. Elvik, R. Assessing the validity of road safety evaluation studies by analysing causal chains. *Accid. Anal. Prev.* **2003**, *35*, 741–748. [[CrossRef](#)]
13. Washington, S.; Persaud, B.; Lyon, C.; Oh, J. *Validation of Accident Models for Intersections*; FHWA-RD-03-037; Federal Highway Administration: Washington, DC, USA, 2005.
14. Roshandel, S.; Zheng, Z.; Washington, S. Impact of real-time traffic characteristics on freeway crash occurrence: Systematic review and meta-analysis. *Accid. Anal. Prev.* **2015**, *79*, 198–211. [[CrossRef](#)]
15. Reurings, M.; Janssen, T.; Eenink, R.; Elvik, R.; Cardoso, J.; Stefan, C. First deliverable of WP2 of the Ripcord-Iserest project. In *Accident Prediction Models and Road Safety Impact Assessment: A State of the Art*; Institute for Road Safety Research: The Hague, The Netherlands, 2006.
16. Kustra, W.; Jamroz, K.; Budzynski, M. Safety PL—A support tool for road safety impact assessment. *Transp. Res. Procedia* **2016**, *14*, 3456–3465. [[CrossRef](#)]
17. Part, D. *Highway Safety Manual*; American Association of State Highway and Transportation Officials: Washington, DC, USA, 2010.
18. Bonneson, J.; Geedipally, S.; Pratt, M.P.; Lord, D. *Safety Prediction Methodology and Analysis Tool for Freeways and Interchanges*; Project No. 17-45; National Cooperative Highway Research Program Transportation Research Board of The National Academies: Washington, DC, USA, 2012.
19. Hauer, E. *Observational before/after Studies in Road Safety*; Pergamon: Oxford, UK, 1997.
20. Hauer, E.; Harwood, D.W.; Council, F.M.; Griffith, M.S. Estimating safety by the empirical bayes method: A tutorial. *Transp. Res. Rec.* **2002**, *126*–131. [[CrossRef](#)]
21. Council, F.M.; Harwood, D.W.; Hauer, E.; Hughes, W.E.; Vogt, A. *Prediction of the Expected Safety Performance of Rural Two-Lane Highways*; Publication No. FHWA-RD-99-207; Federal Highway Administration: Washington, DC, USA, 2000.
22. Elvik, R. The predictive validity of empirical Bayes estimates of road safety. *Accid. Anal. Prev.* **2008**, *40*, 1964–1969. [[CrossRef](#)] [[PubMed](#)]
23. Elvik, R.; Hoye, A.; Vaa, T.; Sorensen, M. *The Handbook of Road Safety Measures*; Emerald Group Publishing: Bingley, UK, 2009.
24. Hauer, E. Identification of sites with promise. *Transp. Res. Rec. J. Transp. Res. Board* **1996**, *1542*, 54–60. [[CrossRef](#)]
25. Hagle, J.; Hecht, M. A comparison of techniques for the identification of hazardous locations. *Transp. Res. Rec.* **1989**, *1238*, 10–19.
26. Hagle, J.L.; Witkowski, J.M. Bayesian identification of hazardous locations. *Transp. Res. Rec.* **1988**, *1185*, 24–36.
27. Sayed, T.; Brown, G.; Navin, F. Simulation of traffic conflicts at unsignalized intersections with TSC-Sim. *Accid. Anal. Prev.* **1994**, *26*, 593–607. [[CrossRef](#)]
28. Kaub, A. Highway corridor safety levels of service based on annual risk of injury. In Proceedings of the 79th Transportation Research Board Annual Meeting, Washington, DC, USA, 9–13 January 2000.
29. Cafiso, S.; D'Agostino, C.; Bağ, R.; Kieć, M. The assessment of road safety for passing relief lanes using microsimulation and traffic conflict analysis. *Adv. Transp. Stud. Int. J.* **2016**, *2*, 55–64.
30. Archer, J. *Indicators for Traffic Safety Assessment and Prediction and Their Application in Micro-Simulation Modelling: A Study of Urban and Suburban Intersections*; Royal Institute of Technology: Stockholm, Sweden, 2005.

31. Ghanim, M.S.; Shaaban, K. A case study for surrogate safety assessment model in predicting real-life conflicts. *Arab. J. Sci. Eng.* **2019**, *44*, 4225–4231. [[CrossRef](#)]
32. Wu, J.; Radwan, E.; Abou-Senna, H. Determination if VISSIM and SSAM could estimate pedestrian-vehicle conflicts at signalized intersections. *J. Transp. Saf. Secur.* **2018**, *10*, 572–585. [[CrossRef](#)]
33. Vasconcelos, L.; Neto, L.; Seco, Á.M.; Silva, A.B. Validation of the surrogate safety assessment model for assessment of intersection safety. *Transp. Res. Rec. J. Transp. Res. Board* **2014**, *2432*, 1–9. [[CrossRef](#)]
34. Oskarbski, J.; Zarski, K. Methodology of research on the impact of ramp metering on the safety and efficiency of road traffic using transport models. In Proceedings of the 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Cracow, Poland, 5–7 June 2019.
35. Dingus, T.A.; Neale, V.L.; Klauer, S.G.; Petersen, A.D.; Carroll, R.J. The development of a naturalistic data collection system to perform critical incident analysis: An investigation of safety and fatigue issues in long-haul trucking. *Accid. Anal. Prev.* **2006**, *38*, 1127–1136. [[CrossRef](#)]
36. Guo, F.; Klauer, S.G.; Hankey, J.M.; Dingus, T.A. Near crashes as crash surrogate for naturalistic driving studies. *Transp. Res. Rec. J. Transp. Res. Board* **2010**, *2147*, 66–74. [[CrossRef](#)]
37. Wu, K.-F.; Jovanis, P.P. Crashes and crash-surrogate events: Exploratory modeling with naturalistic driving data. *Accid. Anal. Prev.* **2012**, *45*, 507–516. [[CrossRef](#)] [[PubMed](#)]
38. Bagdadi, O. Assessing safety critical braking events in naturalistic driving studies. *Transp. Res. Part F Traffic Psychol. Behav.* **2013**, *16*, 117–126. [[CrossRef](#)]
39. Van der Horst, A.R.A. *Video-Recorded Accidents, Conflicts and Road User Behaviour: A Step Forward in Traffic Safety Research*; Palmero Ediciones: Valencia, Spain, 2007.
40. Saunier, N.; Mourji, N.; Agard, B. Mining microscopic data of vehicle conflicts and collisions to investigate collision factors. *Transp. Res. Rec. J. Transp. Res. Board* **2011**, *2237*, 41–50. [[CrossRef](#)]
41. Van Nes, N.; Christoph, M.; Hoedemaeker, M.; van der Horst, R. The value of site-based observations complementary to naturalistic driving observations: A pilot study on the right turn manoeuvre. *Accid. Anal. Prev.* **2013**, *58*, 318–329. [[CrossRef](#)]
42. Gettman, D.; Head, L. Surrogate safety measures from traffic simulation models. *Transp. Res. Rec. J. Transp. Res. Board* **2003**, *1840*, 104–115. [[CrossRef](#)]
43. Xin, W.; Hourdos, J.; Michalopoulos, P. *Enhanced Micro-Simulation Models for Accurate Safety Assessment of Traffic Management ITS Solutions*; CTS 08-17; University of Minnesota Center for Transportation Studies: Minneapolis, MN, USA, 2008.
44. De Ceunynck, T.; Ariën, C.; Brijs, K.; Brijs, T.; van Vlierden, K.; Kuppens, J.; van der Linden, M.; Wets, G. Proactive evaluation of traffic signs using a traffic sign simulator. *Eur. J. Transp. Infrastruct. Res.* **2015**, *15*, 184–204. [[CrossRef](#)]
45. Godley, S.T.; Triggs, T.J.; Fildes, B.N. Driving simulator validation for speed research. *Accid. Anal. Prev.* **2002**, *34*, 589–600. [[CrossRef](#)]
46. Bella, F. Can driving simulators contribute to solving critical issues in geometric design? *Transp. Res. Rec.* **2009**, *2138*, 120–126. [[CrossRef](#)]
47. Yan, X.; Abdel-Aty, M.; Radwan, E.; Wang, X.; Chilakapati, P. Validating a driving simulator using surrogate safety measures. *Accid. Anal. Prev.* **2008**, *40*, 274–288. [[CrossRef](#)]
48. Van Haperen, W. *Review of Current Study Methods for VRU Safety. Appendix 5—Systematic Literature Review: Behavioural Observations*; No. 635895; InDeV: Lund, Sweden, 2016.
49. Bonsall, P.; Liu, R.; Young, W. Modelling safety-related driving behaviour—Impact of parameter values. *Transp. Res. Part A Policy Pract.* **2005**, *39*, 425–444. [[CrossRef](#)]
50. Minderhoud, M.M.; Bovy, P.H.L. Extended time-to-collision measures for road traffic safety assessment. *Accid. Anal. Prev.* **2001**, *33*, 89–97. [[CrossRef](#)]
51. Gettman, D.; Sayed, T.; Pu, L.; Shelby, S. *Surrogate Safety Assessment Model and Validation*; Publication No. Fhwa-Hrt-08-051; Federal Highway Administration: Washington, DC, USA, 2008. [[CrossRef](#)]
52. Bevrani, K.; Chung, E. An examination of the microscopic simulation models to identify traffic safety indicators. *Int. J. Intell. Transp. Syst. Res.* **2012**, *10*, 66–81. [[CrossRef](#)]
53. Golob, T.F.; Recker, W.W.; Alvarez, V.M. Freeway safety as a function of traffic flow. *Accid. Anal. Prev.* **2004**, *36*, 933–946. [[CrossRef](#)]
54. Evans, L.; Wasielewski, P. Risky driving related to driver and vehicle characteristics. *Accid. Anal. Prev.* **1983**, *15*, 121–136. [[CrossRef](#)]

55. Davis, G.A.; Swenson, T. Collective responsibility for freeway rear-ending accidents? An application of probabilistic causal models. *Accid. Anal. Prev.* **2006**, *38*, 728–736. [[CrossRef](#)]
56. Essa, M.; Sayed, T. Traffic conflict models to evaluate the safety of signalized intersections at the cycle level. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 289–302. [[CrossRef](#)]
57. Perkins, S.R.; Harris, J.L. Traffic conflict characteristics-accident potential at intersections (Paper sponsored by Committee on Traffic Safety and presented at the 47th Annual Meeting). In *Traffic Safety Accident Research Highway Research. Recovery*; Highway Research Board: Washington, DC, USA, 1968; pp. 35–43.
58. Hydén, C. The development of a method for traffic safety evaluation: The Swedish traffic conflicts technique. In *Bulletin Lund Institute of Technology, Department; Trafikteknik Tekniska Hoegskdan i Lund: Lund, Sweden, 1987*; p. 57.
59. Svensson, Å. *A Method for Analysing the Traffic Process in a Safety Perspective*; Lund University: Lund, Sweden, 1998.
60. Lord, D.; Washington, S. *Safe Mobility: Challenges, Methodology and Solutions (Transport and Sustainability)*; Emerald Publishing: Bingley, UK, 2018; Volume 11.
61. PIARC. *Road Safety Manual*; World Road Association PIARC: Paris, France, 2004.
62. Fisher, D.L.; Rizzo, M.; Caird, J.K.; Lee, J.D. *Handbook of Driving Simulation for Engineering, Medicine, and Psychology*; CRC Press: Boca Raton, FL, USA, 2011.
63. Lee, J.D.; Young, K.L.; Regan, M.A. Defining driver distraction. In *Driver Distraction: Theory, Effects, and Mitigation*; CRC Press: Boca Raton, FL, USA, 2009.
64. Pettitt, M.; Burnett, G.; Stevens, A. Defining driver distraction. In Proceedings of the 12th World Congress on Intelligent Transport Systems, San Francisco, CA, USA, 6–10 November 2005.
65. Peng, Y.; Boyle, L.N.; Hallmark, S.L. Driver's lane keeping ability with eyes off road: Insights from a naturalistic study. *Accid. Anal. Prev.* **2013**, *50*, 628–634. [[CrossRef](#)]
66. Horberry, T.; Anderson, J.; Regan, M.A.; Triggs, T.J.; Brown, J. Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance. *Accid. Anal. Prev.* **2006**, *38*, 185–191. [[CrossRef](#)]
67. Blaschke, C.; Breyer, F.; Färber, B.; Freyer, J.; Limbacher, R. Driver distraction based lane-keeping assistance. *Transp. Res. Part F Traffic Psychol. Behav.* **2009**, *12*, 288–299. [[CrossRef](#)]
68. Klauer, S.G.; Dingus, T.A.; Neale, V.L.; Sudweeks, J.D.; Ramsey, D.J. *The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data*; Publication No. DOT HS 810 594; National Highway Traffic Safety Administration: Washington, DC, USA, 2006.
69. He, J.; McCarley, J.S.; Kramer, A.F. Lane keeping under cognitive load: Performance changes and mechanisms. *Hum. Factors* **2014**, *56*, 414–426. [[CrossRef](#)] [[PubMed](#)]
70. Guerrero-Ibañez, J.; Zeadally, S.; Contreras-Castillo, J. Sensor technologies for intelligent transportation systems. *Sensors* **2018**, *18*, 1212. [[CrossRef](#)] [[PubMed](#)]
71. Shi, J.; Wu, J. Research on adaptive cruise control based on curve radius prediction. In Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017. [[CrossRef](#)]
72. Bulumulle, G.; Bölöni, L. A study of the automobile blind-spots' spatial dimensions and angle of orientation on side-sweep accidents. In Proceedings of the 2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS), Pasadena, CA, USA, 3–6 April 2016. [[CrossRef](#)]
73. Kim, S.G.; Kim, J.E.; Yi, K.; Jung, K.H. Detection and tracking of overtaking vehicle in blind spot area at night time. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017. [[CrossRef](#)]
74. Lin, Y.-C.; Nguyen, H.-L.T.; Wang, C.-H. Adaptive neuro-fuzzy predictive control for design of adaptive cruise control system. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017. [[CrossRef](#)]
75. Tarko, A.; Romero, M.A.; Bandura, V.K.; Ariyur, K.B.; Lizarazo, C.G. Feasibility of Tracking Vehicles at Intersections with a Low-end LiDAR. In Proceedings of the Transportation Research Board 96th Annual Meeting, Washington, DC, USA, 8–12 January 2017.
76. Katzourakis, D.I.; Lazic, N.; Olsson, C.; Lidberg, M.R. Driver steering override for lane-keeping aid using computer-aided engineering. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 1543–1552. [[CrossRef](#)]

77. Astarita, V.; Festa, D.C.; Giofrè, V.P. Mobile systems applied to traffic management and safety: A state of the art. *Procedia Comput. Sci.* **2018**, *134*, 407–414. [[CrossRef](#)]
78. Guido, G.; Vitale, A.; Saccomanno, F.F.; Festa, D.C.; Astarita, V.; Rogano, D.; Gallelli, V. Using smartphones as a tool to capture road traffic attributes. *Appl. Mech. Mater.* **2013**, *432*, 513–519. [[CrossRef](#)]
79. Bar-Gera, H. Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel. *Transp. Res. Part C Emerg. Technol.* **2007**, *15*, 380–391. [[CrossRef](#)]
80. Guido, G.; Vitale, A.; Saccomanno, F.; Gallelli, V. Sensitivity of simulated vehicle tracking profiles for input into safety performance analysis. *Adv. Transp. Stud.* **2016**, *2*, 65–74.
81. Guido, G.; Vitale, A.; Astarita, V.; Saccomanno, F.; Giofrè, V.P.; Gallelli, V. Estimation of safety performance measures from smartphone sensors. *Procedia Soc. Behav. Sci.* **2012**, *54*, 1095–1103. [[CrossRef](#)]
82. Bierlaire, M.; Chen, J.; Newman, J. *Modeling Route Choice Behavior from Smartphone GPS Data*; TRANSP-OR 101016; Ecole Polytechnique Fédérale de Lausanne: Lausanne, Switzerland, 2010.
83. Barbagli, B.; Manes, G.; Facchini, R.; Marta, S.; Manes, A. Acoustic sensor network for vehicle traffic monitoring. In Proceedings of the the 1st International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2012), Venice, Italy, 24–29 June 2012; pp. 1–6.
84. Zhou, Y.; Dey, K.C.; Chowdhury, M.; Wang, K.-C. Process for evaluating the data transfer performance of wireless traffic sensors for real-time intelligent transportation systems applications. *IET Intell. Transp. Syst.* **2017**, *11*, 18–27. [[CrossRef](#)]
85. Geetha, S.; Cicilia, D. IoT enabled intelligent bus transportation system. In Proceedings of the 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 19–20 October 2017. [[CrossRef](#)]
86. Ki, Y.K. Accident detection system using image processing and MDR. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 35–39.
87. Desai, G.; Ambre, V.; Jakharia, S.; Sherkhane, S. Smart road surveillance using image processing. In Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 5 January 2018. [[CrossRef](#)]
88. Saunier, N.; Sayed, T.; Ismail, K. Large-scale automated analysis of vehicle interactions and collisions. *Transp. Res. Rec. J. Transp. Res. Board* **2010**, *2147*, 42–50. [[CrossRef](#)]
89. Haugen, T.; Levy, J.R.; Aakre, E.; Tello, M.E.P. Weigh-in-motion equipment—Experiences and challenges. *Transp. Res. Procedia* **2016**, *14*, 1423–1432. [[CrossRef](#)]
90. Fu, C.; Liu, H. Investigating influence factors of traffic violations at signalized intersections using data gathered from traffic enforcement camera. *PLoS ONE* **2020**, *15*, e0229653. [[CrossRef](#)]
91. Jang, J.A.; Kim, H.S.; Cho, H.B. Smart roadside system for driver assistance and safety warnings: Framework and applications. *Sensors* **2011**, *11*, 7420–7436. [[CrossRef](#)] [[PubMed](#)]
92. Grumert, E.; Tapani, A.; Ma, X. Evaluation of four control algorithms used in variable speed limit systems. In Proceedings of the Transportation Research Board 95th Annual Meeting, Washinton, DC, USA, 20–29 June 2016; pp. 16–2880.
93. Barceló, J.; Montero, L.; Bullejos, M.; Serch, O.; Carmona, C. A kalman filter approach for exploiting bluetooth traffic data when estimating time-dependent od matrices. *J. Intell. Transp. Syst. Technol. Plan. Oper.* **2013**, *17*, 123–141. [[CrossRef](#)]
94. Oskarbski, J.; Zawisza, M.; Żarski, K. Automatic incident detection at intersections with use of telematics. *Transp. Res. Procedia* **2016**, *14*, 3466–3475. [[CrossRef](#)]
95. Chaturvedi, M.; Srivastava, S. Multi-modal design of an intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2017–2027. [[CrossRef](#)]
96. Axer, S.; Friedrich, B. Level of service estimation based on low-frequency floating car data. *Transp. Res. Procedia* **2014**, *3*, 1051–1058. [[CrossRef](#)]
97. Pedzierska, M.; Kamiński, T. The use of simulator studies to assess the impact of ITS services on road users behaviour. *MATEC Web Conf.* **2018**, *231*, 1–8. [[CrossRef](#)]
98. El Faouzi, N.E.; Klein, L.A. Data fusion for ITS: Techniques and research needs. *Transp. Res. Procedia* **2016**, *15*, 495–512. [[CrossRef](#)]
99. Castanedo, F. A review of data fusion techniques. *Sci. World J.* **2013**, *2013*, 1–19. [[CrossRef](#)] [[PubMed](#)]
100. Luo, R.C.; Yih, C.-C.; Su, K.L. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sens. J.* **2002**, *2*, 107–119. [[CrossRef](#)]

101. Ang, L.-M.; Seng, K.P. Big sensor data applications in urban environments. *Big Data Res.* **2016**, *4*, 1–12. [[CrossRef](#)]
102. Kyriakou, C.; Christodoulou, S.E.; Dimitriou, L. Roadway pavement anomaly classification utilizing smartphones and artificial intelligence. In Proceedings of the 2016 18th Mediterranean Electrotechnical Conference (MELECON), Lemesos, Cyprus, 18–20 April 2016. [[CrossRef](#)]
103. Oskarbski, J. *Perspectives of Telematics Implementation in Tri-City Transport Systems Management and planning*; Springer: Berlin/Heidelberg, Germany, 2011. [[CrossRef](#)]
104. PTV Group. *VISUM Fundamentals*; PTV Group: Karlsruhe, Germany, 2012.
105. ATKINS. *SATURN User Manual*, 11.3th ed.; ATKINS: Epsom, UK, 2015.
106. PTV Group. *PTV VISSIM 10 Manual*; PTV Group: Karlsruhe, Germany, 2017.
107. Transportation Research Board. *HCM2010 Highway Capacity Manual*, 5th ed.; Transportation Research Board: Washington, DC, USA, 2010; Volume 2.
108. Knoop, V.; Hoogendoorn, S.; Adams, K. Capacity reductions at incidents sites on Motorways. *Eur. J. Transp. Infrastruct. Res.* **2009**, *9*, 363–379. [[CrossRef](#)]
109. Abdel-Aty, M.; Wang, L. Reducing real-time crash risk for congested expressway weaving segments using ramp metering. In Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, Italy, 26–28 June 2017. [[CrossRef](#)]
110. Hayward, J.C. Near-miss determination through use of a scale of danger. *Highw. Res. Rec.* **1972**, *384*, 22–34.
111. Allaby, P.; Hellinga, B.; Bullock, M. Variable speed limits: Safety and operational impacts of a candidate control strategy for freeway applications. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 671–680. [[CrossRef](#)]
112. Sadat, M.; Celikoglu, H.B. Simulation-based variable speed limit systems modelling: An overview and a case study on istanbul freeways. *Transp. Res. Procedia* **2017**, *22*, 607–614. [[CrossRef](#)]
113. Grumert, E.; Ma, X.; Tapani, A. Analysis of a cooperative variable speed limit system using microscopic traffic simulation. *Transp. Res. Part C Emerg. Technol.* **2015**, *52*, 173–186. [[CrossRef](#)]
114. Cao, J.; Hu, D.; Luo, Y.; Qiu, T.Z.; Ma, Z. Exploring the impact of a coordinated variable speed limit control on congestion distribution in freeway. *J. Traffic Transp. Eng.* **2015**, *2*, 167–178. [[CrossRef](#)]
115. Khondaker, B.; Kattan, L. Variable speed limit: A microscopic analysis in a connected vehicle environment. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 146–159. [[CrossRef](#)]
116. Yu, R.; Abdel-Aty, M. An optimal variable speed limits system to ameliorate traffic safety risk. *Transp. Res. Part C Emerg. Technol.* **2014**, *46*, 235–246. [[CrossRef](#)]
117. Hadiuzzaman, M.; Fang, J.; Karim, M.A.; Luo, Y.; Qiu, T.Z. Modeling driver compliance to vsl and quantifying impacts of compliance levels and control strategy on mobility and safety. *J. Transp. Eng.* **2015**, *141*, 12. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Human-Like Obstacle Avoidance Trajectory Planning and Tracking Model for Autonomous Vehicles That Considers the Driver's Operation Characteristics

Qinyu Sun, Yingshi Guo *, Rui Fu, Chang Wang and Wei Yuan

School of Automobile, Chang'an University, Xi'an 710064, China; sunqinyu@chd.edu.cn (Q.S.); furui@chd.edu.cn (R.F.); wangchang@chd.edu.cn (C.W.); yuanwei@chd.edu.cn (W.Y.)

* Correspondence: guoys@chd.edu.cn

Received: 22 July 2020; Accepted: 25 August 2020; Published: 26 August 2020

Abstract: Developing a human-like autonomous driving system has gained increasing amounts of attention from both technology companies and academic institutions, as it can improve the interpretability and acceptance of the autonomous system. Planning a safe and human-like obstacle avoidance trajectory is one of the critical issues for the development of autonomous vehicles (AVs). However, when designing automatic obstacle avoidance systems, few studies have focused on the obstacle avoidance characteristics of human drivers. This paper aims to develop an obstacle avoidance trajectory planning and trajectory tracking model for AVs that is consistent with the characteristics of human drivers' obstacle avoidance trajectory. Therefore, a modified artificial potential field (APF) model was established by adding a road boundary repulsive potential field and ameliorating the obstacle repulsive potential field based on the traditional APF model. The model predictive control (MPC) algorithm was combined with the APF model to make the planning model satisfy the kinematic constraints of the vehicle. In addition, a human driver's obstacle avoidance experiment was implemented based on a six-degree-of-freedom driving simulator equipped with multiple sensors to obtain the drivers' operation characteristics and provide a basis for parameter confirmation of the planning model. Then, a linear time-varying MPC algorithm was employed to construct the trajectory tracking model. Finally, a co-simulation model based on CarSim/Simulink was established for off-line simulation testing, and the results indicated that the proposed trajectory planning controller and the trajectory tracking controller were more human-like under the premise of ensuring the safety and comfort of the obstacle avoidance operation, providing a foundation for the development of AVs.

Keywords: autonomous vehicle; obstacle avoidance; artificial potential field; model predictive control; human-like

1. Introduction

The vehicle active obstacle avoidance system is one of the core issues in the research of autonomous vehicle (AV) control [1,2]. A safe and reasonable obstacle avoidance trajectory planning in real time based on accurate obstacle information perception through multiple sensors can promote trajectory tracking technology, which can effectively improve the intelligent level of the autonomous system and reduce the frequency of traffic accidents [3–5]. As one of the key technologies of an active obstacle avoidance system for vehicles, the local trajectory replanning refers to designing a safe trajectory that enables AVs to promptly and accurately bypass obstacles based on global path planning [6]. Under the premise of satisfying multiple constraints, the designed trajectory should also comply with human drivers' driving characteristics of obstacle avoidance. Therefore, active obstacle avoidance trajectory planning and control have become a difficulty in vehicle lateral control. Determining how to ameliorate

the human-like degree of trajectory planning and tracking is the basis for achieving obstacle avoidance control, and is also an effective way for improving the safety and acceptability of AVs [7].

At present, the methods of local trajectory planning mainly include the fuzzy logic (FL) method, genetic algorithm (GA), neural network (NN) method, A* algorithm, rapidly exploring random tree (RRT), state-space trajectory-generation (ST), and artificial potential field (APF) method. The FL method is combined with the perception action of fuzzy control and replaces mathematical variables with linguistic variables [8]. Fuzzy control conditional statements describe the complex relationships between variables. Although the accuracy of the designed trajectory is improved, the FL algorithm itself lacks flexibility. The fuzzy rules and membership degree cannot change with the environment. The GA seeks the optimal solution by imitating the mechanism of selection and inheritance in nature, and then plans the trajectory through coding, crossover, mutation, and the construction of fitness function [9]. However, the programming implementation is relatively complex, and it is difficult to ensure real-time performance. The NN method uses a biologically-inspired NN method to establish an obstacle avoidance trajectory planning model. This method treats each grid as a neuron and the motion space of the vehicle is regarded as a topological NN [10]. Through training the grid, the parameters of the NN model can be adjusted to adapt to different scenarios. Therefore, the algorithm has good learning ability and stability, though the interpretability of the model is insufficient.

The heuristic search method represented by the A* algorithm has the characteristics of fast calculation speed and good flexibility [11]. Weight A* and ARA* accelerate the search direction to the target position by introducing and adjusting the weight value of the heuristic function [12]. A*-Connect adopts a bidirectional search strategy to obtain higher search efficiency [13]. Hybrid A* uses the variant of the A* search to plan trajectories that conform to kinematic constraints based on the three-dimensional motion state space, achieving local optimization through numerical nonlinear optimization [14]. By considering the vehicle motion direction information and the forward and backward movement patterns, a four-dimensional search space is constructed through this algorithm. The RRT method is an efficient planning method in multi-dimensional space [15]. RRT* adopts the tree node in the neighborhood with the lowest total generation value as the parent node, and reconnects the tree nodes in the neighborhood in each iteration, allowing the path on the random tree to always be progressively optimal. Informed-RRT* establishes an ellipsoid with the initial point and the target point as the focus [16]. This ellipsoid is employed as the search domain to gradually optimize the path, and the area of the ellipsoid decreases during the search process. Finally, the ellipsoid converges into the optimal path. Reachability-guided RRT reduces the sensitivity of systems with different constraints to random sampling for measurements in the extended tree, and it is only possible to select the node when the distance from the sampling point to the node is greater than the distance from the sampling point to the accessible set of the node [17]. Continuous-curvature RRT adopts a target-biased sampling strategy, a node connection mechanism based on a reasonable metric function, and a post-processing method that satisfies vehicle motion constraints to improve the speed and quality of planning [18].

The APF method was first proposed by Khatib in the study regarding the obstacle avoidance trajectory planning of robots [19]. It has the characteristics of a small amount of calculation, short planning time, and high execution efficiency. In recent years, it has been gradually applied to the local trajectory planning of intelligent vehicles [20]. The basic theory is to establish an obstacle repulsive potential field and a target point gravitational potential field through the sensor's perception of the environment, as well as to find the descending path of the total potential field as the obstacle avoidance path in the compound potential field [21]. However, the traditional APF method has problems such as local minimum points, and the obstacle avoidance trajectory planning for the robot does not consider the robot's own size range and boundary environment. The intelligent vehicle's obstacle avoidance is also constrained by the size of the vehicle and obstacles and the road boundary [22]. In addition, the planning trajectory should also meet the actual dynamic and kinematic constraints. Therefore, the traditional APF method needs to be improved to satisfy the need for trajectory planning for AVs. Tokson et al. [23] used the gradient descent algorithm of the potential field to find an effective

path in the potential energy field, and added repulsive potential energy when the vehicle failed into a minimum point, to construct a modified potential field to make the controlled object continue to move towards the target point. Bounini et al. [24] proposed the concept of the steering potential field. The steering potential field is established by issuing steering commands to intelligent vehicles through remote control stations or vehicle-mounted navigation systems. Then, the obstacle repulsion potential field is established according to obstacle data. Raja et al. [25] introduced a gradient function in the traditional APF method, which was composed of gravity, repulsion, tangential force, and gradient force according to a certain weight to form a modified potential field function. The gradient force is the function of vehicle yaw and pitch angle at a specific position, which ensures that the vehicle does not drive in the direction of high gradient force, thus obtaining the expected obstacle avoidance trajectory. Zhang et al. [26] employed the elliptic distance to replace the actual distance in the traditional repulsion potential field, and comprehensively considered the influence of lane lines, obstacles, and the road boundary potential energy field on vehicles to obtain a smoother local obstacle avoidance trajectory. Kenealy et al. [27] proposed an enhanced space-based potential field model to realize through the exploration of complex environments for autonomous robots.

Trajectory tracking has been the subject of numerous empirical studies that have investigated different control algorithms to establish an autonomous control model. The commonly used tracking control algorithms mainly include the proportion integration differentiation (PID) control algorithm, optimal preview control (OPC) algorithm, fuzzy control algorithm, sliding mode control (SMC) algorithm, linear quadratic regulator (LQR) control algorithm, and model predictive control (MPC) algorithm. The traditional PID control mainly relies on adjusting the gains of the three parts of the proportional unit, integral unit, and differential unit to set its characteristics [28]. This algorithm is simple and easy to operate, but has poor adaptability in trajectory tracking under complex conditions. The OPC algorithm sets a preview point on the forward road of the vehicle, and realizes the tracking control of the expected trajectory by reducing the lateral deviation between the preview point and the expected road center line [29]. Liu et al. [30] improved the preview distance based on longitudinal speed and steering angle feedback, and designed a trajectory tracking controller according to the preview error model to make the change of steering angle more stable. Park et al. [31] applied the proportion integration control in the lateral offset to reduce tracking error and alleviate the impact of preview distance with velocity change. The fuzzy control algorithm is mainly divided into three parts: input fuzzy, fuzzy reasoning, and defuzzification [32]. The trajectory tracking is realized by designing different membership functions. The increment of the front wheel angle was taken as the output of the controller for the controller designed by Trabia et al. [33], which included a steering fuzzy module and an obstacle avoidance fuzzy module. This algorithm reduced the computation of the controller. The SMC algorithm, also known as variable structure control, can dynamically change based on the current state of the system to achieve the goal of gradually stabilizing to the equilibrium point according to the predetermined state trajectory [34]. The MPC algorithm has advantages in dealing with linear and nonlinear systems with constraints [35]. Kong et al. [36] designed MPC controllers based on vehicle kinematics and dynamics, and implemented tests to compare the prediction errors of the two controllers. The results demonstrated that the MPC controller based on kinematics had better performance and less computation under the low speed condition. Zanon et al. [37] combined the nonlinear MPC with the moving vision estimation method to solve the problem of poor trajectory tracking accuracy on the road with a low friction coefficient.

How to improve the real-time performance and human-like degree is the difficulty and kernel of trajectory planning research. The trajectory planning model based on machine learning and deep learning algorithm can take into account the above two key points, but the inexplicability of the model still cannot be effectively solved. The heuristic search method represented by the A* algorithm has the characteristics of fast calculation speed. However, there are some differences between the planned trajectory and the actual driving trajectory derived from human drivers, since this type of algorithm relies more on the data processing method of computer and lacks a mechanism model similar to driver

behavior. The traditional APF model could simulate the obstacle avoidance behavior of drivers, but how to prompt the human-like degree and some defects of the algorithm still need further study. In the actual driving process, the driver will control the vehicle in advance through preview behavior, and MPC algorithm can simulate the preview behavior of the driver by adjusting the prediction time domain. Existing research combines the APF trajectory planning model with the MPC algorithm to achieve obstacle avoidance. Due to the complexity of the vehicle dynamic model and considering the real-time requirements, the prediction time domain in the MPC algorithm cannot set too large. In addition, the vehicle kinematic model is frequently ignored in the control models. On the one hand, the human-like degree of the obstacle avoidance control would be weakened, and on the other hand, the comfort and smoothness of the planned trajectory would be influenced.

To address the deficiencies in the obstacle avoidance trajectory planning model based on the APF algorithm and the trajectory tracking model based on the MPC algorithm, a modified APF algorithm was proposed in the present research by establishing a road boundary repulsion potential field and an obstacle repulsion potential field with variable parameter. To make the planned obstacle avoidance trajectory meet the vehicle kinematics constraints and ameliorate the human-like degree, the APF algorithm was combined with the MPC algorithm to construct the obstacle avoidance trajectory replanning controller. Considering that there are many kinds of constraints during vehicle lateral control and for the sake of guaranteeing the real-time capability, accuracy, and robustness of the trajectory tracking control algorithm at different speeds, a linear time-varying model predictive trajectory tracking controller was established based on linearizing the vehicle monorail dynamic model. The controller on the basis of MPC determined the vehicle front wheel angle as the control variable, and multiple constraints for the vehicle dynamics and kinematics were combined to design the objective function that can achieve the requirements of fast and accurate tracking of the desired trajectory. In addition, this work implemented driver obstacle avoidance experiments under different speeds based on a driving simulator with six degrees of freedom to ensure that the established trajectory planning model was consistent with a human driver's obstacle avoidance characteristics; that is, the planning trajectory was similar to the driver operation trajectory. Two pivotal parameters in the APF algorithm were determined to enhance the human-like degree of planned trajectory and the trajectory characteristics derived from human drivers were extracted to provide a basis for the parameters design of the proposed trajectory planning model for AVs. Finally, the co-simulation model based on CarSim/Simulink was established for the off-line simulation testing of the obstacle avoidance trajectory planning controller and the trajectory tracking controller designed in this study.

The remainder of the paper is organized as follows. Section 2 details the obstacle avoidance trajectory planning model based on the APF algorithm and the MPC algorithm. Section 3 provides detailed information on the trajectory tracking model based on the linear time-varying MPC algorithm. Section 4 presents the experimental design, process, equipment, and feature analysis of the human driver's obstacle avoidance trajectory. The co-simulation results of the proposed trajectory planning controller and trajectory tracking controller are introduced in Section 5. Finally, conclusions are presented in Section 6. The main framework of this study is presented in Figure 1.

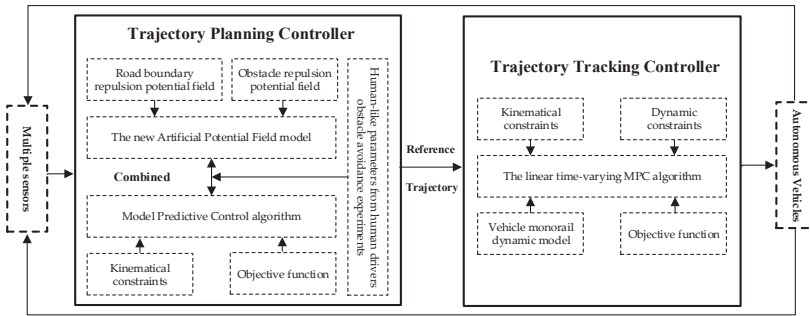


Figure 1. Human-like obstacle avoidance system framework for AVs.

2. Obstacle Avoidance Trajectory Planning Model

2.1. Traditional Artificial Potential Field Model

Khatib first proposed the APF algorithm in 1986. The basic idea of this algorithm is to virtualize the motion of the controlled object in the environment as a forced motion of particles in the artificial virtual force field [38]. The obstacle exerted a repulsive force on the controlled object, and the target point exerted a gravitational force on the controlled object. The controlled object moved toward the combined force of the repulsive force and the gravitational force, as shown in Figure 2. In the figure, F_{rep} is the repulsive force generated by the obstacle, F_{att} is the gravitational force generated by the target point, and F_{sum} is the resultant force. The distance between the controlled object and the obstacle and the target point mainly determines the magnitude of the repulsive force and gravitational force. The smaller the distance between the controlled object and the obstacle, the greater the repulsive force. Further, the greater the distance between the controlled object and the target point, the greater the gravity.

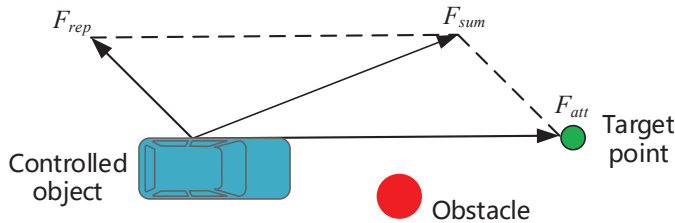


Figure 2. Model of the traditional APF algorithm.

In the traditional APF algorithm, the controlled object is reduced to a particle, and its motion space is regarded as a two-dimensional Euclidean space. Assuming that the coordinate of the controlled object X in space is (x, y) and the target point X_{goal} coordinate is (x_{goal}, y_{goal}) , the gravitational field function of the controlled object in space is defined as a quadratic function related to the position of the controlled object and the target point:

$$U_{att}(X) = \frac{1}{2}k_g\rho_g^2, \tag{1}$$

where k_g is the gain coefficient of the gravitational potential field, ρ_g is the relative distance between the controlled vehicle and the target point, the value is the vector, and the direction is the controlled vehicle points to the target point.

The gravitational force on the controlled object can be obtained by calculating the negative gradient of the gravitational potential field:

$$F_{att}(X) = -\nabla U_{att}(X) = -k_g \vec{u}_g, \quad (2)$$

where \vec{u}_g is the unit vector where the controlled object points to the target point.

Assuming that the coordinates of the obstacle X_{obs} in the space is (x_{obs}, y_{obs}) , the repulsive force field function on the controlled object is defined as:

$$U_{rep}(X) = \begin{cases} \frac{1}{2}k_o \left(\frac{1}{\rho_{ob}} - \frac{1}{\rho_o}\right)^2, \rho_{ob} \leq \rho_o \\ 0, \rho_{ob} > \rho_o \end{cases} \quad (3)$$

where k_o is the repulsive potential field coefficient, ρ_{ob} is the distance constant between the controlled object and the obstacle, and ρ_o is the influence range of the repulsive potential field of the obstacle. When $\rho_{ob} > \rho_o$, the controlled object is not affected by the repulsive force of the obstacle.

The repulsive force on the controlled object can be obtained by calculating the negative gradient of the repulsive potential field:

$$F_{rep}(X) = -\nabla U_{rep}(X) = \begin{cases} k_o \left(\frac{1}{\rho_{ob}} - \frac{1}{\rho_o}\right)^2 \vec{u}_{ob}, \rho_{ob} \leq \rho_o \\ 0, \rho_{ob} > \rho_o \end{cases} \quad (4)$$

where \vec{u}_{ob} is the unit vector where the obstacle points to the controlled object.

Therefore, the combined force of the controlled object when moving in the force field space is:

$$F_{sum}(X) = -\nabla U_{sum}(X) = F_{att}(X) + \sum_{i=1}^n F_{rep,i}(X), \quad (5)$$

where n is the number of obstacles.

The traditional APF algorithm has the following problems when it is used to plan the local obstacle avoidance trajectory of vehicles [39].

(1) Lack of road boundary constraints. The algorithm only considers the passability of obstacle avoidance trajectories, and does not consider the road boundary constraints during vehicle driving.

(2) The goal may be unreachable. When there is an obstacle near the target point, the repulsive force of the vehicle when approaching the target point is greater than the gravitational force, so that the controlled object cannot reach the target point.

(3) The controlled object may come to a deadlock. There may be a situation where the controlled object receives the same repulsive force and gravity at a certain point, resulting in the controlled object being unable to continue to advance.

2.2. Modified Artificial Potential Field Model

In order to solve the above deficiencies in the traditional APF algorithm, a modified APF model is proposed through establishing the road boundary repulsive potential field, ameliorating the obstacle potential field, and combining with the MPC algorithm.

2.2.1. Road Boundary Repulsive Potential Field

Road boundary repulsive potential field is established on the basis of the lane boundary, which is used to limit the driving area of vehicles to ensure that vehicles continue to drive along the center line of the lane after obstacle avoidance, and the vehicle body would not exceed the road boundary during the process of turning and avoiding obstacles. The established road boundary repulsive potential field is presented in Figure 3. The repulsive potential field generates a force based on the road boundary in

the direction of the vehicle, and the repulsive force only takes the lateral force component of the earth coordinate system. The value of the road boundary repulsion is inversely proportional to the relative distance between the vehicle and boundary. The smaller the relative distance, the greater the repulsion, and the larger the relative distance, the smaller the repulsion.

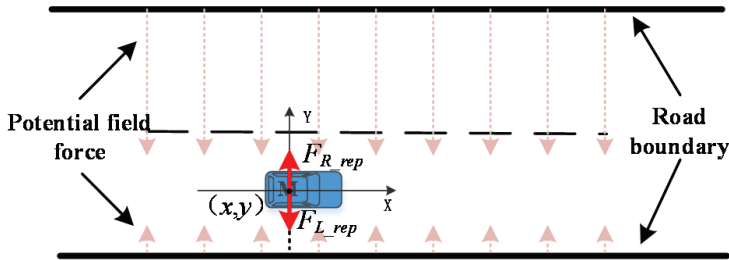


Figure 3. Schematic diagram of the road boundary repulsive potential field force.

When there is no obstacle in the lane, the vehicle travels along the center line of the right lane under the action of the repulsion potential field of the road boundary. Considering the size of the vehicle, the road boundary repulsive potential field model is established as follows:

$$\begin{cases} U_{L_rep}(X) = \frac{1}{2}k_{L_rep}\left(\frac{1}{\rho_{L_rep}-\frac{w_v}{2}}\right)^2 \\ U_{R_rep}(X) = \frac{1}{2}k_{R_rep}\left(\frac{1}{\rho_{R_rep}-\frac{w_v}{2}}\right)^2 \end{cases} \quad (6)$$

where k_{L_rep} and k_{R_rep} are the repulsive potential field coefficients of the left and right road boundaries, respectively; w_v is the lateral width of the vehicle, and ρ_{L_rep} and ρ_{R_rep} are the shortest distances between the center of mass of the vehicle and the boundary of the left and right lanes, respectively.

The repulsive force on the controlled object can be obtained by calculating the negative gradient of the road boundary repulsive potential field:

$$\begin{cases} F_{L_rep}(X) = k_{L_rep}\left(\frac{1}{\rho_{L_rep}-\frac{w_v}{2}}\right)\left(\frac{1}{\rho_{L_rep}}\right)^3 \vec{a}_{Lv} \\ F_{R_rep}(X) = k_{R_rep}\left(\frac{1}{\rho_{R_rep}-\frac{w_v}{2}}\right)\left(\frac{1}{\rho_{R_rep}}\right)^3 \vec{a}_{Rv} \end{cases} \quad (7)$$

where \vec{a}_{Lv} and \vec{a}_{Rv} are the unit vector where the road boundaries point to the controlled object.

2.2.2. Obstacle Repulsive Potential Field

The circular repulsion field of the traditional APF does not satisfy the requirements of the actual vehicle obstacle avoidance trajectory according to the human driver experience, and it is difficult to meet the requirements of steering smoothness in the trajectory planning, resulting in a decrease of ride comfort. Therefore, the scope of action of the potential field was modified in this work, and the longitudinal action distance of the obstacle repulsion potential field was increased, so that the vehicle can correct the direction in advance to avoid obstacles; the lateral action distance was reduced to prevent the vehicle from driving out of the lane during obstacle avoidance. Figure 4 illustrates the schematic diagram of the obstacle repulsive potential field. The longitudinal and lateral

acting distances of the repulsive potential field of an obstacle were defined as A and B, respectively. The scope of action of the repulsive field ρ_o can be rewritten as:

$$\rho_o \in \frac{(x - x_{obs})^2}{A^2} + \frac{(y - y_{obs})^2}{B^2} \quad (8)$$

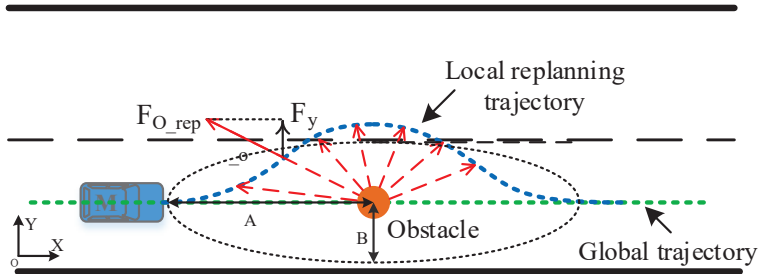


Figure 4. Schematic diagram of the road boundary repulsive potential field force.

Considering that the obstacle avoidance process is similar to the lane change process, the force exerted by the obstacle repulsive potential field on the vehicle can only be retained in the lateral component under the geodetic coordinate system to avoid the vehicle coming to a deadlock. The repulsive direction of the obstacle to the vehicle is upward when the vehicle enters the obstacle repulsive potential field. At this time, the vehicle will turn to the left for avoidance. During this process, the repulsive potential energy increases with the decrease of the relative distance between the vehicle and the obstacle, thus forcing the vehicle to drive away from the obstacle.

The obstacle repulsive potential field is established with an obstacle as the center of the potential energy. Within the scope of action of the repulsive potential field of an obstacle, it exerts a repulsive force on the vehicle to keep the vehicle away from the obstacle. In the traditional APF model, the gravitational force is less than the repulsive force when the vehicle reaches the target point, which will lead to the problem of unreachable target. Therefore, an adjustment factor R_d^m is added to the obstacle repulsive potential field. In this way, the relative distance between the vehicle and the target point R_d is supplemented in the modified obstacle repulsive potential field. Hence the repulsive force and the gravitation force are reduced to zero at the same time only when the vehicle reaches the target point, so that the problem of unreachable target is solved. The modified obstacle repulsion potential field function is shown as follows:

$$U_{o_rep}(X) = \begin{cases} \eta_{rep} \left\{ \exp \left[-\frac{1}{2} \left(\frac{(x - x_{obs})^2}{A^2} + \frac{(y - y_{obs})^2}{B^2} \right) \right] \right\} R_d^m, & p_{o_rep}^n \leq \rho_o \\ 0, & p_{o_rep}^n > \rho_o \end{cases} \quad (9)$$

where R_d is the relative distance between the vehicle and the target point, m is constant, η_{rep} is the repulsive potential field coefficient of the obstacle, $p_{o_rep}^n$ is the distance between the vehicle and the n th obstacle, and ρ_o is the range of action of the repulsive field.

In addition, the vehicle may come to a deadlock when the repulsive force from other surrounding vehicles is equal to the gravitational force. In this case, the value of m in the adjustment factor will gradually increase from 0 until the force balance is broken, so that the vehicle could jump out of the local minimum and then the value of m would return to the original value. Within the scope of

the obstacle, the repulsive force on the controlled object can be obtained by calculating the negative gradient of the obstacle repulsive potential field:

$$F_{o_rep}(X) = \begin{cases} -\eta_{rep} \left(\frac{(x-x_{obs})^2}{A^2} + \frac{(y-y_{obs})^2}{B^2} \right)^2 R_d^m \vec{a}_{ov}, & p_{o_rep} \leq \rho_o \\ 0, & p_{o_rep} > \rho_o \end{cases} \quad (10)$$

where \vec{a}_{ov} is the unit vector where the obstacle points to the controlled object.

2.3. Model Prediction Algorithm With Trajectory Planning

To ensure that the planning trajectory of the modified APF model is practical and can satisfy the kinematic constraints of the vehicle, the MPC algorithm was combined with the modified APF model and a reasonable objective function was constructed to minimize the deviation between the planning trajectory of the modified APF model and the predicted trajectory of the MPC algorithm. Due to the low real-time requirement of the planning layer, the adoption of a relatively simple point mass model can fully meet the requirements of re-planning. Therefore, as shown in Figure 5, the steering motion model was established with XOY as the geodetic coordinate system and xoy as the vehicle coordinate system.

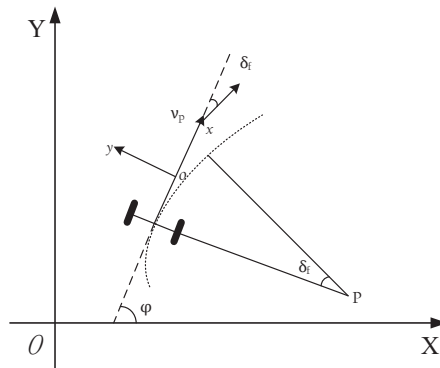


Figure 5. The vehicle kinematics model.

The vehicle kinematics model can be expressed as follows:

$$\begin{cases} \dot{X} = \dot{x}\cos(\varphi) - \dot{y}\sin(\varphi) \\ \dot{Y} = \dot{x}\sin(\varphi) + \dot{y}\cos(\varphi) \\ \dot{\varphi} = \frac{\dot{y}}{\dot{x}} \\ \ddot{x} = 0 \end{cases} \quad (11)$$

where \dot{x} and \dot{y} represent the longitudinal and lateral speeds in the vehicle coordinate system, respectively; \ddot{x} and \ddot{y} correspondingly represent the longitudinal and lateral accelerations in the vehicle coordinate system; φ and $\dot{\varphi}$ represent the yaw angle and yaw rate of the vehicle, respectively; \dot{X} and \dot{Y} correspondingly represent the longitudinal and lateral speeds in the geodetic coordinate system.

This article only considers the obstacle avoidance strategy of the vehicle at constant speed, so the longitudinal acceleration is set to zero. Five discrete state variables were determined as $X = [\dot{x}, \dot{y}, \varphi, X, Y]$, and the lateral acceleration was selected as the control variable $v = [\ddot{y}]$. Then the state equation can be expressed as:

$$\dot{X}(t) = f(X(t), v(t)) \quad (12)$$

Using Taylor expansion and first-order difference quotient to linearize and discretize Equation (13), the linear time-varying model can be obtained as follows:

$$\tilde{X}(k+1) = \tilde{A}_p(k)\tilde{X}(k) + \tilde{B}_p(k)\tilde{v}(k) \tag{13}$$

where $\tilde{A}_p(k) = \begin{bmatrix} 1 & 0 & -\dot{x}\sin(\varphi)T \\ 0 & 1 & \dot{x}\cos(\varphi)T \\ 0 & 0 & 1 \end{bmatrix}$, $\tilde{B}_p(k) = \begin{bmatrix} \cos(\varphi)T & 0 \\ \sin(\varphi)T & 0 \\ T \tan(\delta_f)T/l & \dot{x}T/l\cos^2(\delta_f) \end{bmatrix}$, T is the sampling time, l is the wheel base, and δ_f is the front wheel angle.

The control objective in the trajectory planning layer is to minimize the deviation between the planning trajectory of the modified APF model and the predicted trajectory of the MPC algorithm under the premise of ensuring the smooth and comfortable driving of vehicles. Therefore, the objective function of trajectory planning is defined as follows:

$$J_p(k) = \sum_{j=1}^{N_{pp}} \|\eta_p(k+j|t) - \eta_{ref}(k+j|t)\|_{Q_p}^2 + \sum_{j=1}^{N_{pc}-1} \|\Delta\tilde{v}(k+j|t)\|_{R_p}^2, \tag{14}$$

where Q_p and R_p are the weight matrixes, η_{ref} is the planning trajectory of the modified APF model, η_p is the predicted trajectory of the MPC algorithm, and N_{pp} and N_{pc} are, respectively, the prediction step size and control step size of the MPC controller.

Then the output can be expressed as:

$$\tilde{Y}(k) = \psi_p\tilde{X}(k) + \Theta_p\Delta\tilde{v}(k), \tag{15}$$

$$\eta_p(k) = \tilde{C}_p\tilde{X}(k), \tag{16}$$

$$\text{where, } \tilde{Y}(k) = \begin{bmatrix} \eta_p(k+1|k) \\ \eta_p(k+2|k) \\ \vdots \\ \eta_p(k+N_{pc}|k) \\ \vdots \\ \eta_p(k+N_{pp}|k) \end{bmatrix}, \psi_p = \begin{bmatrix} \tilde{C}_p\tilde{B}_p \\ \tilde{C}_p\tilde{B}_p^2 \\ \vdots \\ \tilde{C}_p\tilde{B}_p^{N_{pc}} \\ \vdots \\ \tilde{C}_p\tilde{B}_p^{N_{pp}} \end{bmatrix}, \tilde{C}_p = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \Theta_p = \begin{pmatrix} \tilde{C}_p\tilde{B}_p & 0 & 0 & 0 \\ \tilde{C}_p\tilde{A}_p\tilde{B}_p & \tilde{C}_p\tilde{B}_p & 0 & 0 \\ \vdots & \ddots & \vdots & \\ \tilde{C}_p\tilde{A}_p^{N_{pc}-1}\tilde{B}_p & \tilde{C}_p\tilde{A}_p^{N_{pc}-2}\tilde{B}_p & \dots & \tilde{C}_p\tilde{A}_p\tilde{B}_p \\ \vdots & \ddots & \vdots & \\ \tilde{C}_p\tilde{A}_p^{N_{pp}-1}\tilde{B}_p & \tilde{C}_p\tilde{A}_p^{N_{pp}-2}\tilde{B}_p & \dots & \tilde{C}_p\tilde{A}_p^{N_{pp}-N_{pc}-1}\tilde{B}_p \end{pmatrix}.$$

It is also necessary to append obstacle avoidance constraints and limit the control variables for the sake of ensuring that the planning trajectory is practical. The obstacle avoidance constraints are divided into road constraints and obstacle constraints:

$$\begin{cases} \rho_{L_rep} > \frac{w_p}{2} \\ \rho_{R_rep} > \frac{w_p}{2} \\ \rho_{obs}^n > 1.2w_{obs} \end{cases} \tag{17}$$

where w_{obs} is the width of the obstacle.

In addition, the lateral acceleration of the vehicle is mainly provided by the lateral force of the tire, so it must meet the limit of tire adhesion:

$$\ddot{y} \in [-\mu g, \mu g] \quad (18)$$

where μ is the coefficient of road adhesion, and g is the acceleration of gravity.

Combining Equations (15), (18) and (19), the trajectory planning model can be expressed as:

$$\left\{ \begin{array}{l} \min \sum_{j=1}^{N_{pp}} \|\eta_p(k+j|k) - \eta_{ref}(k+j|k)\|_{Q_p}^2 + \sum_{j=1}^{N_{pc}-1} \|\Delta \bar{v}(k+j|k)\|_{R_p}^2 \\ \text{st. } \rho_{L_rep} > \frac{w_v}{2} \\ \rho_{L_rep} > \frac{w_v}{2} \\ \rho_{obs}^n > 1.2w_{obs} \\ -\mu g \leq \ddot{y} \leq \mu g \end{array} \right. \quad (19)$$

3. Obstacle Avoidance Trajectory Tracking Model

The MPC algorithm can use the dynamic prediction model to obtain the future vehicle state in a limited time domain based on the current vehicle motion state. This method has a strong ability to deal with multi-objective constraints [40]. In this work, a linear time-varying MPC controller was established to track the trajectory from the obstacle avoidance trajectory planning model.

3.1. Vehicle Dynamic Model

Considering that the longitudinal speed remains unchanged and only the front wheel angle is controlled during obstacle avoidance, the following assumptions are made in the modeling process.

(1) The lateral forces and slip angles on the left and right tires of the vehicle are symmetric and equal in the vehicle coordinate system.

(2) The test sections are all flat roads, ignoring the influence of slope and other factors on the vertical movement of vehicles.

(3) The front wheel angle is small, and the lateral force of the tire is approximately linear with the slip angle of the tire.

(4) The influence of the suspension system, transmission system, air resistance, and the longitudinal and lateral coupling force of the tire is ignored.

The monorail dynamics model is established as shown in Figure 6, and the dynamic equation of the model can be described as follows:

$$\left\{ \begin{array}{l} \dot{X} = v_x \cos(\varphi) - (v_y + l_f \dot{\varphi}) \sin(\varphi) \\ \dot{Y} = v_y \sin(\varphi) - (v_x + l_f \dot{\varphi}) \cos(\varphi) \\ I_z \ddot{\varphi} = l_f F_{yf} - l_r F_{yr} \\ m \dot{v}_x = m \dot{v}_x \dot{\varphi} + F_{xf} + F_{xr} \\ m \dot{v}_y = -m \dot{v}_y \dot{\varphi} + F_{yf} + F_{yr} \end{array} \right. \quad (20)$$

where \dot{X} and \dot{Y} represent the longitudinal and lateral speed in the geodetic coordinate system, v_x , v_y , and φ represent the longitudinal speed, lateral speed, and heading angle in the vehicle coordinate system, m represent vehicle mass, l_f and l_r represent the distance from the center of mass to the front and rear axles, F_{xf} , F_{xr} , F_{yf} , and F_{yr} represents the longitudinal and lateral forces of the front and rear axles, and I_z represent the moment of inertia.

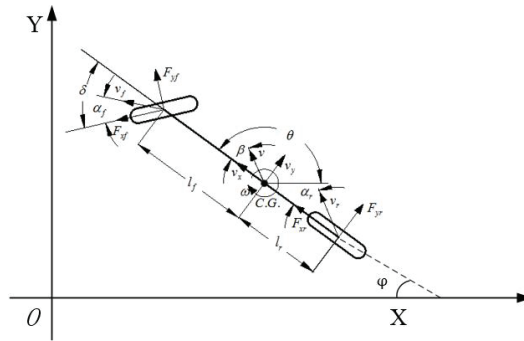


Figure 6. The vehicle monorail dynamics model.

The state variables was determined as $\hat{\xi}_c = [v_x, v_y, \varphi, \dot{\varphi}, X, Y]^T$, and the control variable was selected as $v_c = \delta_f$. To satisfy the real-time requirements of the trajectory tracking controller when the vehicle is traveling at high speeds, the nonlinear dynamic model is linearized to obtain the linear time-varying equation:

$$\dot{\xi}_c = A_c(t)\dot{\xi}_c(t) + B_c(t)v_c(t), \tag{21}$$

where $A_c(t) = \left. \frac{\partial f(\xi_c, v_c)}{\partial \xi_c} \right|_{\xi_c(t), v_c(t)}$, and $B_c(t) = \left. \frac{\partial f(\xi_c, v_c)}{\partial v_c} \right|_{\xi_c(t), v_c(t)}$.

Using the first-order difference quotient to discretize Equation (22), the discrete state space expression can be obtained:

$$\hat{\xi}_c(k+1) = A_c(k)\hat{\xi}_c(k) + B_c(k)v_c(k) \tag{22}$$

where $A_c(k) = I_c + TA_c(t)$, $B_c(k) = I_c + TB_c(t)$, $C_c = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$, and I_c is unit matrix.

3.2. Objective Function

To ensure that the trajectory tracking controller can promptly and smoothly track the expected trajectory, the following form of objective function is adopted:

$$J_c(k) = \sum_{i=1}^{N_{cp}} \|\eta_c(k+i|t) - \eta_{pref}(k+i|t)\|_{Q_c}^2 + \sum_{i=1}^{N_{cc}-1} \|\Delta U(k+i|t)\|_{R_c}^2 + \rho \varepsilon^2, \tag{23}$$

where N_{cp} and N_{cc} are the prediction step size and control step size of the controller respectively; Q_c and R_c are the weight coefficients, ε is the relaxation factor, ρ is the relaxation coefficient, and η_{pref} is the expected trajectory from the trajectory planning controller.

In Equation (24), the first item on the right side of the equal sign reflects the degree of tracking accuracy of the system; the second item is the constraint on the change of control quantity and increment of control quantity, reflecting the vehicle’s ability to maintain stability; the third item is the relaxation factor, which prevents the objective function from having no solution in the real-time calculation process.

In the objective function, it is necessary to calculate the output of the vehicle in the predictive time domain based on the linear error model, and Equation (23) was converted into:

$$\begin{cases} \tilde{\xi}_c(k+1|t) = \tilde{A}_c(k)\tilde{\xi}_c(k|t) + \tilde{B}_c(k)\Delta v_c(k|t) \\ \eta_c(k|t) = \tilde{C}_c\tilde{\xi}_c(k|t) \end{cases} \tag{24}$$

where $\widetilde{A}_c(k) = \begin{bmatrix} A_c(k) & B_c(k) \\ 0_{m \times n} & I_m \end{bmatrix}$, $\widetilde{B}_c(k) = \begin{bmatrix} B_c(k) \\ I_m \end{bmatrix}$, $\widetilde{C}_c = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$, m is the dimension of state quantity, and n is the dimension of control quantity.

To simplify the calculation, assume $k = 1, \dots, t + N - 1$, and the predicted output expression of the system can be deduced as follows:

$$Y_c(t) = \psi_c \widetilde{\xi}_c(t|t) + \Theta_c \Delta U_c(t|t) \tag{25}$$

where $Y(t) = \begin{bmatrix} \eta_c(t+1|t) \\ \eta_c(t+2|t) \\ \vdots \\ \eta_c(t+N_{cc}|t) \\ \vdots \\ \eta_c(t+N_{cp}|t) \end{bmatrix}$, $\psi_c = \begin{bmatrix} \widetilde{C}_c \widetilde{B}_c \\ \widetilde{C}_c \widetilde{B}_c^2 \\ \vdots \\ \widetilde{C}_c \widetilde{B}_c^{N_{cc}} \\ \vdots \\ \widetilde{C}_c \widetilde{B}_c^{N_{cp}} \end{bmatrix}$, $\Delta U_c(t) = \begin{bmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \vdots \\ \Delta u(t+N_c|t) \end{bmatrix}$, and $\Theta_c = \begin{pmatrix} \widetilde{C}_c \widetilde{B}_c & 0 & 0 & 0 \\ \widetilde{C}_c \widetilde{A}_c \widetilde{B}_c & \widetilde{C}_c \widetilde{B}_c & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{C}_c \widetilde{A}_c^{N_{cc}} \widetilde{B}_c & \widetilde{C}_c \widetilde{A}_c^{N_{cc}-1} \widetilde{B}_c & \dots & \widetilde{C}_c \widetilde{A}_c \widetilde{B}_c \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{C}_c \widetilde{A}_c^{N_{cp}-1} \widetilde{B}_c & \widetilde{C}_c \widetilde{A}_c^{N_{cp}-2} \widetilde{B}_c & \dots & \widetilde{C}_c \widetilde{A}_c^{N_{cp}-N_{cc}-1} \widetilde{B}_c \end{pmatrix}$.

By substituting Equation (26) into Equation (24), the complete objective function can be obtained.

3.3. Constraint Condition

One of the advantages of the MPC controller is its ability to handle multiple target constraints. On the one hand, the design of the constraints in the optimization solution should match the mechanical design constraints of the vehicle steering mechanism. On the other hand, it should also satisfy the needs of vehicle smooth control. The vehicle dynamic constraints need to be considered in the actual trajectory tracking control process, and the specific constraints include the centroid slip angle constraint, tire slip angle constraint, and road adhesion condition.

During the obstacle avoidance process, the front wheel angle and the increment of front wheel angle should satisfy the following constraints:

$$\begin{cases} -25^\circ \leq \delta_f \leq 25^\circ \\ -0.47^\circ \leq \Delta \delta_f \leq 0.47^\circ \end{cases} \tag{26}$$

The centroid slip angle directly affects the vehicle's driving stability and is an important reference index in vehicle stability control. The empirical formula of the centroid slip angle constraint is expressed as follows:

$$-\arctan(0.196\mu) \leq \beta \leq \arctan(0.196\mu), \tag{27}$$

where μ is the coefficient of road adhesion.

According to the relationship between the centroid slip angle and the front wheel angle, the tire slip angle can be expressed as:

$$\begin{cases} \alpha_f = \frac{v_y + l_f \dot{\varphi}}{v_x} - \delta_f \\ \alpha_r = \frac{v_y + l_r \dot{\varphi}}{v_x} \end{cases} \tag{28}$$

There is a linear relationship between the slip angle and the corresponding lateral force of tire when the tire slip angle is relatively small. Hence, the front tire slip angle constraint can be expressed as:

$$-6^\circ < \alpha_f < 6^\circ \tag{29}$$

The road adhesion condition determines the range of vehicle lateral force that can be provided, and it also affects vehicle control stability. The following constraints should be met between the vehicle lateral acceleration and the road adhesion condition:

$$-\mu g \leq a_y \leq \mu g \tag{30}$$

Therefore, the specific optimization problem can be equivalent to the multi-constraint quadratic programming problem, which can be expressed as:

$$\left\{ \begin{array}{l} \min \sum_{i=1}^{N_{cp}} \|\eta_c(k+i|t) - \eta_{pref}(k+i|t)\|_{Q_c}^2 + \sum_{i=1}^{N_{cc}-1} \|\Delta U(k+i|t)\|_{R_c}^2 + \rho \varepsilon^2 \\ \text{st. } \quad -25^\circ \leq \delta_f \leq 25^\circ \\ \quad \quad -0.47^\circ \leq \Delta \delta_f \leq 0.47^\circ \\ \quad \quad -6^\circ < \alpha_f < 6^\circ \\ \quad \quad -\mu g \leq a_y \leq \mu g \\ \quad \quad \varepsilon > 0 \end{array} \right. \tag{31}$$

By solving Equation (32), the increment sequence of the control quantity can be expressed as:

$$\Delta U(t) = \begin{bmatrix} \Delta u(t|t) \\ \Delta u(t+1|t) \\ \vdots \\ \Delta u(t+N_c-1|t) \end{bmatrix} \tag{32}$$

On this basis, the first increment of control quantity in Equation (33) is taken as the actual output and is superimposed with the actual output control quantity in the previous period to obtain the actual control output quantity in the current period:

$$u(t) = u(t-1) + \Delta u(t|t) \tag{33}$$

The actual output control quantity was implemented on the system, and the objective function was resolved based on the feedback state quantity in the next control cycle. Therefore, the incremental sequence of the control quantity was constantly updated to achieve the purpose of rolling optimization. Finally, the above optimization solution process was repeated to complete the vehicle trajectory tracking control.

4. Driving Simulator Experiments

To make the trajectory planned by the obstacle avoidance trajectory planning controller satisfy the safety requirements and be more human-like, it is necessary to extract the human driver's obstacle avoidance trajectory and perform statistical analysis on the trajectory characteristics, which provides a basis for the parameters design of the trajectory planning controller. Therefore, in this study, the obstacle avoidance experiments based on a driving simulator with six degrees of freedom were implemented, and the obstacle avoidance trajectories from different drivers were extracted for further analysis.

4.1. Apparatus

Considering that the actual vehicle obstacle avoidance experiment possesses certain risks, this study employed a driving simulator to perform the obstacle avoidance experiment. The driving simulator is a modified simulation technology that combines pure digital simulation with field test. The vehicle, driving field, and various types of sensors are constructed by a digital method to reproduce the real driving scene and satisfy various requirements of the vehicle test and development. The driving simulator tests are characterized by low cost, high efficiency, repeatability, and low risk coefficient. The driving simulator used in this work is presented in Figure 7. The simulator mainly includes a vibration platform with six degrees of freedom, a front view ring display system, a cockpit system, and high-performance workstation, which has a strong sense of immersion in driving operation. In addition, the driving simulator is equipped with multiple sensors for collecting the driver's operation and road environment information, including steering wheel angle sensor, accelerator pedal sensor, brake pedal sensor, virtual millimeter wave radar sensor, and virtual LIDAR sensor.

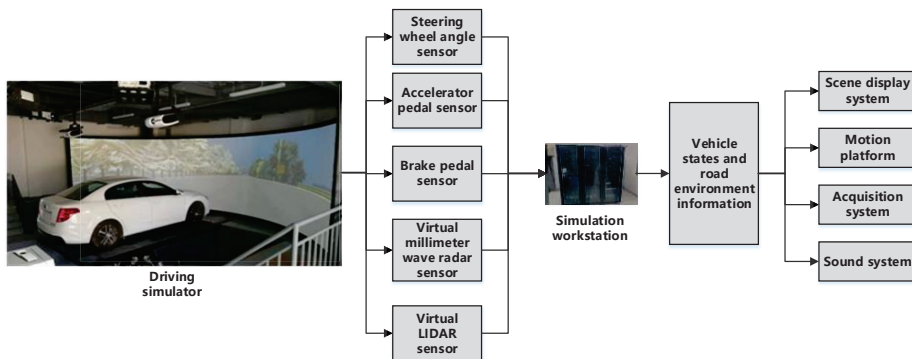


Figure 7. Composition of the driving simulator.

4.2. Participants and Experimental Program

Twenty-eight experienced drivers participated in the obstacle avoidance experiment. The ages of the drivers ranged from 23 to 48 years old, with an average age of 32.2 years (standard deviation = 5.82). Their driving experience ranged from 5 to 26 years (mean = 12.6, standard deviation = 4.6). All of the participants were non-professional drivers with a valid driver's license, normal or corrected vision, and who had experienced no serious traffic accidents over the past three years.

A two-way six-lane straight urban road with a length of 2 km was selected as the test section to implement the obstacle avoidance experiment, as exhibited in Figure 8. The obstacle was stationary and placed in the middle lane 1 km from the vehicle starting point. Each participant was required to navigate the vehicle at three speeds of 40 km/h, 60 km/h, and 80 km/h from the starting point, and drove forward along the center line of the middle lane. The participants were required to execute the obstacle avoidance operation in a safety distance according to their driving habits when they noticed the obstacle in front of the road. They were also required to return to the original lane after the completion of the obstacle avoidance operation. The size of the obstacle was 4710 × 1820 × 1500 mm. Each participant needed to complete three tests at different speeds and try to keep a constant speed during the avoidance operation.

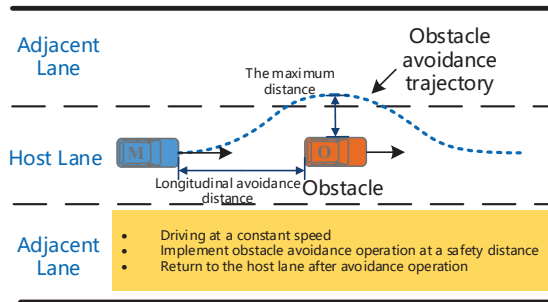


Figure 8. Schematic diagram of the obstacle avoidance experiment.

4.3. Procedures

Before the experiment, the drivers were asked to participate in a practice round for approximately 10 min to familiarize themselves with the driving simulator and testing process. Next, the test staff introduced the experimental objectives and notes. After the beginning of the experiment, the participants performed the obstacle avoidance operation as required, and relevant data would be recorded in real time. After each experiment, the participants were free to manipulate the driving simulator until the beginning of the next experiment. To alleviate driving fatigue, the participants could rest for 5 min after every testing period. During the test, the driver was required to strictly abide by the traffic rules. In case of emergency, such as the abnormal operation of the driving simulator or equipment, the unsatisfactory condition of the participants, and so on, the test would be stopped immediately and the test vehicle would be safely parked in the emergency parking zone. Participants were paid ¥100 for their participation after they had finished all the experiments.

4.4. Collected Data

The data collected during the obstacle avoidance experiments mainly included the longitudinal and lateral coordinates of the vehicle in the road coordinate system, vehicle speed, and acceleration. The sampling frequency was 100 Hz. After the test, a total of 180 groups of effective obstacle avoidance data were obtained. Then, Matlab was used to fit the collected trajectories, with the results presented in Figures 9–11.

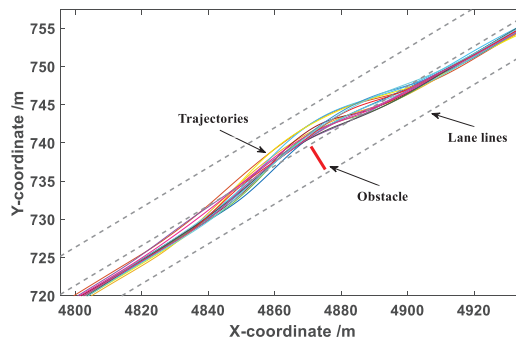


Figure 9. Obstacle avoidance trajectories under a vehicle speed of 40 km/h.

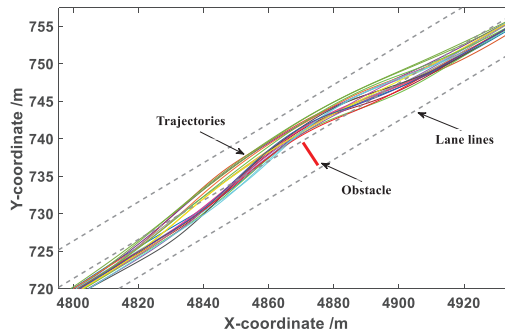


Figure 10. Obstacle avoidance trajectories under a vehicle speed of 60 km/h.

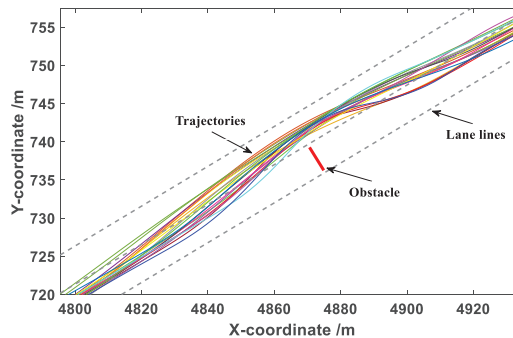


Figure 11. Obstacle avoidance trajectories under a vehicle speed of 80 km/h.

It can be seen from Figures 9–11 that the drivers in each group of tests successfully completed the obstacle avoidance operation and the obstacle avoidance trajectory was smooth, so the data collected in the test were valid data. The longitudinal distance at the beginning of obstacle avoidance and the maximum lateral distance during the obstacle avoidance were statistically analyzed under different vehicle speeds.

The coordinate point when the vehicle generated continuous lateral displacement was determined as the starting position of the obstacle avoidance operation, and the distance between the starting point and the centroid of the obstacle was defined as the longitudinal distance at the beginning of obstacle avoidance. This value can provide a basis for the determination of the A value in the elliptical repulsive potential field (shown in Figure 4). The box diagram of longitudinal distance at the beginning of obstacle avoidance under different vehicle speeds is presented in Figure 12.

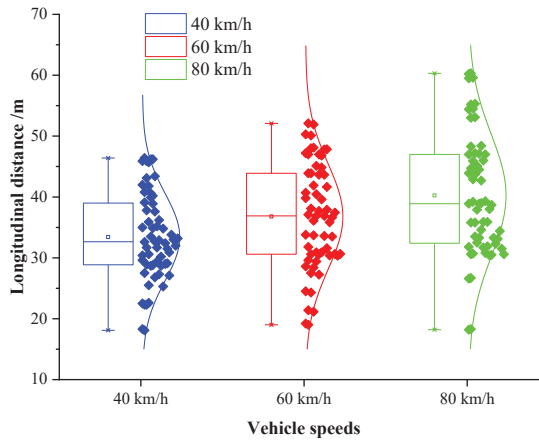


Figure 12. Box diagram of the longitudinal distance at the beginning of avoidance under different speeds.

It can be seen from Figure 12 that the average longitudinal distance at the beginning of obstacle avoidance under the speeds of 40 km/h, 60 km/h, and 80 km/h were 33.4 m, 37.5 m, and 40.6 m, respectively, and the medians were 32.7 m, 37.0 m, and 38.1 m. The longitudinal distance increased with the promotion of the vehicle speed. The results of the one-way analysis of variance indicated that the vehicle speed possessed a significant effect on the longitudinal distance at the beginning of obstacle avoidance ($p = 0.000 < 0.05$, $F(2, 177) = 9.320$). Therefore, in this paper, the vehicle speed and the longitudinal distance were determined as reference factors, and the least square method was used for linear regression fitting. The expression is as follows:

$$a = 0.1725v_p + 26.517 \quad (34)$$

where a is the longitudinal distance at the beginning obstacle avoidance, and v_p is the vehicle speed.

The maximum lateral distance was defined as the maximum lateral distance between the vehicle and the obstacle during the obstacle avoidance process. This value can provide a basis for the determination of the B value in the elliptical repulsive potential field (shown in Figure 4). The box diagram of the maximum lateral distance under different vehicle speeds is presented in Figure 13.

It can be seen from Figure 13 that the average maximum lateral distance during the process of obstacle avoidance under the speeds of 40 km/h, 60 km/h, and 80 km/h were 3.44 m, 3.57 m, and 3.65 m, respectively, and the medians were 3.51 m, 3.63 m, and 3.71 m. The maximum lateral distance increased slightly with the promotion of the vehicle speed. The results of the one-way analysis of variance indicated that the vehicle speed possessed no significant effect on the maximum lateral distance during the process of obstacle avoidance ($p = 0.254 > 0.05$, $F(2, 177) = 1.380$). Therefore, in this paper, the average of the maximum lateral distance of all data was determined as the final value of maximum lateral distance:

$$b = 3.46 \text{ m} \quad (35)$$

where b is the maximum lateral distance.

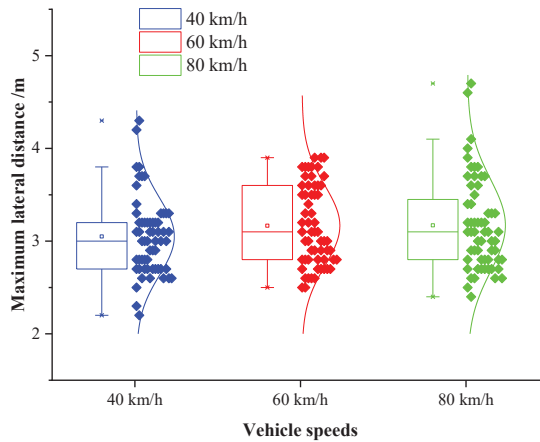


Figure 13. Box diagram of the maximum lateral distance under different speeds.

5. Co-Simulation Results Analysis

5.1. Co-Simulation Model Establishment

To verify the obstacle avoidance trajectory planning controller and the MPC trajectory tracking controller designed in this study, a co-simulation model based on CarSim and Simulink was established for simulation testing. The co-simulation model is illustrated in Figure 14.

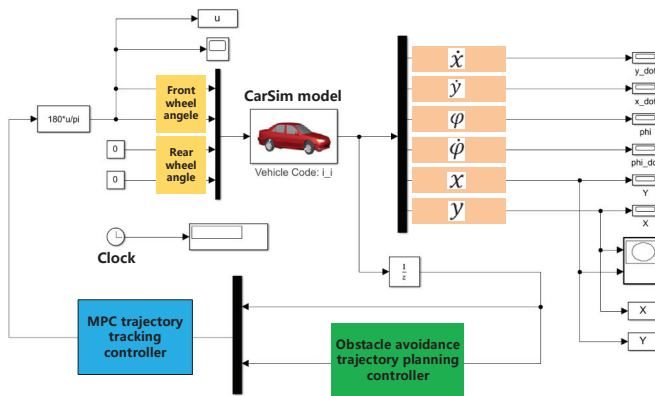


Figure 14. CarSim/Simulink co-simulation model.

As shown in Figure 14, \dot{x} is the vehicle longitudinal speed, \dot{y} is the vehicle lateral speed, φ is the vehicle heading angle, $\dot{\varphi}$ is the vehicle yaw rate, and x and y are the vehicle coordinate information in the geodetic coordinate system. CarSim was responsible for building the vehicle dynamics model, as the Vehicle Code: i_i module shown in the figure, and outputting the coordinate information, the longitudinal and lateral speeds, the heading angle and the yaw rate to the trajectory planning controller and the trajectory tracking controller, respectively. Simulink was responsible for constructing the trajectory planning model based on the modified APF algorithm and the trajectory tracking model based on the MPC algorithm. The trajectory planning controller provided a reference trajectory for the trajectory tracking controller, and the tracking module outputted the final calculated front wheel angle to the vehicle dynamics module in CarSim. Then, the updated vehicle state parameters were employed for calculation in the next control period.

The B-Class Hatchback with front-wheel drive was selected as the vehicle dynamics simulation model in CarSim, and the main parameters are shown in Table 1.

Table 1. Basic parameters of the vehicle dynamics model.

| Symbol. | Parameters | Value |
|---------|--|-----------|
| m | Vehicle sprung mass | 1563.1 kg |
| l_f | Distance between the center of mass and the front axis | 1174.2 mm |
| l_r | Distance between the center of mass and the rear axis | 1358.8 mm |
| h_g | Height of the center of mass | 716.5 mm |
| B | Wheel track | 1480 mm |
| f | Coefficient of rolling resistance | 0.015 |
| C_D | Coefficient of air resistance | 0.8 |

The specific simulation conditions were set as follows: the global reference trajectory was a straight path; the road adhesion coefficient was set as 0.8; the obstacle coordinate was set as (105, 0), and the obstacle was $4710 \times 1820 \times 1500$ mm, and the vehicle speeds were 40 km/h, 60 km/h, and 80 km/h respectively.

The specific parameters of the trajectory planning controller and trajectory tracking controller in Simulink were set as follows: the prediction step size and control step size of the trajectory planning controller were determined as $N_{pp} = 15$, and $N_{pc} = 5$; the weight matrixes of the trajectory planning

controller were determined as $Q_p = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$, $R_p = 10$; the prediction step size and control

step size of the trajectory tracking controller were determined as $N_{cp} = 20$ and $N_{cc} = 10$; and the weight

matrixes of the trajectory planning controller were determined as $Q_p = \begin{bmatrix} 2000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$, $R_p = 1.5 \times 10^5$. The control period of both controllers was 0.01 s.

5.2. Co-Simulation Results

The comparison results of the co-simulation of the obstacle avoidance trajectory planning with different algorithms under different vehicle speeds are exhibited in Figures 15–17.

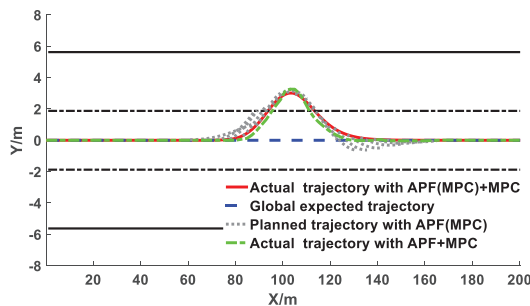


Figure 15. Co-simulation results of obstacle avoidance trajectory planning under 40 km/h.

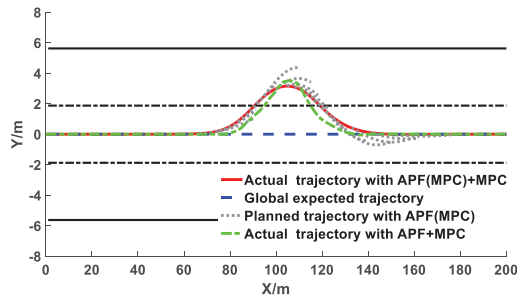


Figure 16. Co-simulation results of obstacle avoidance trajectory planning under 60 km/h.

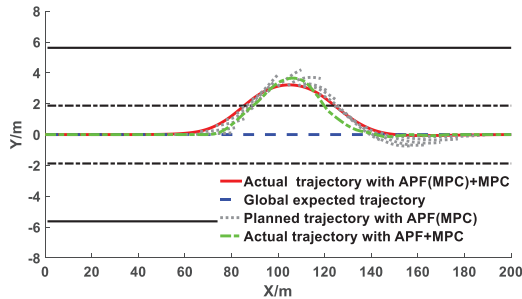


Figure 17. Co-simulation results of obstacle avoidance trajectory planning under 80 km/h.

It can be observed from Figures 15–17 that under different vehicle speeds, the obstacle avoidance controllers based on the APF with MPC model derived from previous study [41] and the APF(MPC) with MPC model proposed in this study can effectively plan and track the local obstacle avoidance trajectory that satisfies the obstacle and road boundary constraints in real time in the predicted time domain, and the trajectory tracking controllers based on the linear time-varying MPC algorithm can promptly and accurately track the first two reference points of the local planned trajectory from the trajectory planning controllers in real time. The vehicles avoided obstacles smoothly under different speeds, which indicated that the trajectory planning and tracking controllers were both feasible and effective. However, under different vehicle speeds, the longitudinal distance at the beginning of the obstacle avoidance derived from the controller proposed in this study were larger than that of resulting from the other controller (APF with MPC). The longitudinal distance from the proposed model in this study under the vehicle speeds of 40 km/h, 60 km/h, and 80 km/h were 34 m, 38 m, and 41 m, respectively; while the values from the previous model were 29 m, 34 m, and 37 m, respectively. With the increase of vehicle speed, the obstacle avoidance trajectory planning was advanced and the longitudinal distance was promoted. In addition, the maximum lateral distance during the obstacle avoidance process remained basically unchanged, and the value under the different vehicle speeds from the proposed model in this study were 3.48 m, 3.50 m, and 3.51 m, respectively, while the values from the previous model were 3.69 m, 3.77 m, and 3.87 m, respectively. The specific results during the process of the obstacle avoidance control are exhibited in Table 2 and Figure 18.

Table 2. Comparison results of performance parameters.

| Models | Speed | $ \delta_{f,max} $ | $ \varphi_{max} $ | $ \dot{\varphi}_{max} $ | $ a_{y,max} $ | $ \dot{j}_{y,max} $ |
|--------------|---------|--------------------|-------------------|-------------------------|-----------------------|-----------------------|
| APF+MPC | 40 km/h | 6.02° | 11.81° | 20.91°/s | 0.79 m/s ² | 3.34 m/s ³ |
| | 60 km/h | 5.21° | 10.51° | 23.94°/s | 0.72 m/s ² | 3.80 m/s ³ |
| | 80 km/h | 5.35° | 9.46° | 23.58°/s | 0.95 m/s ² | 3.84 m/s ³ |
| APF(MPC)+MPC | 40 km/h | 5.73° | 11.14° | 18.69°/s | 0.59m/s ² | 2.82 m/s ³ |
| | 60 km/h | 4.92° | 9.82° | 21.17°/s | 0.70 m/s ² | 2.86 m/s ³ |
| | 80 km/h | 4.88° | 8.79° | 21.19°/s | 0.79 m/s ² | 2.91 m/s ³ |
| Human driver | 40 km/h | 4.85° | 10.77° | 14.74°/s | 0.52 m/s ² | 2.31 m/s ³ |
| | 60 km/h | 4.24° | 9.36° | 19.44°/s | 0.69 m/s ² | 2.04 m/s ³ |
| | 80km/h | 3.62° | 7.68° | 20.91°/s | 0.71 m/s ² | 2.85 m/s ³ |

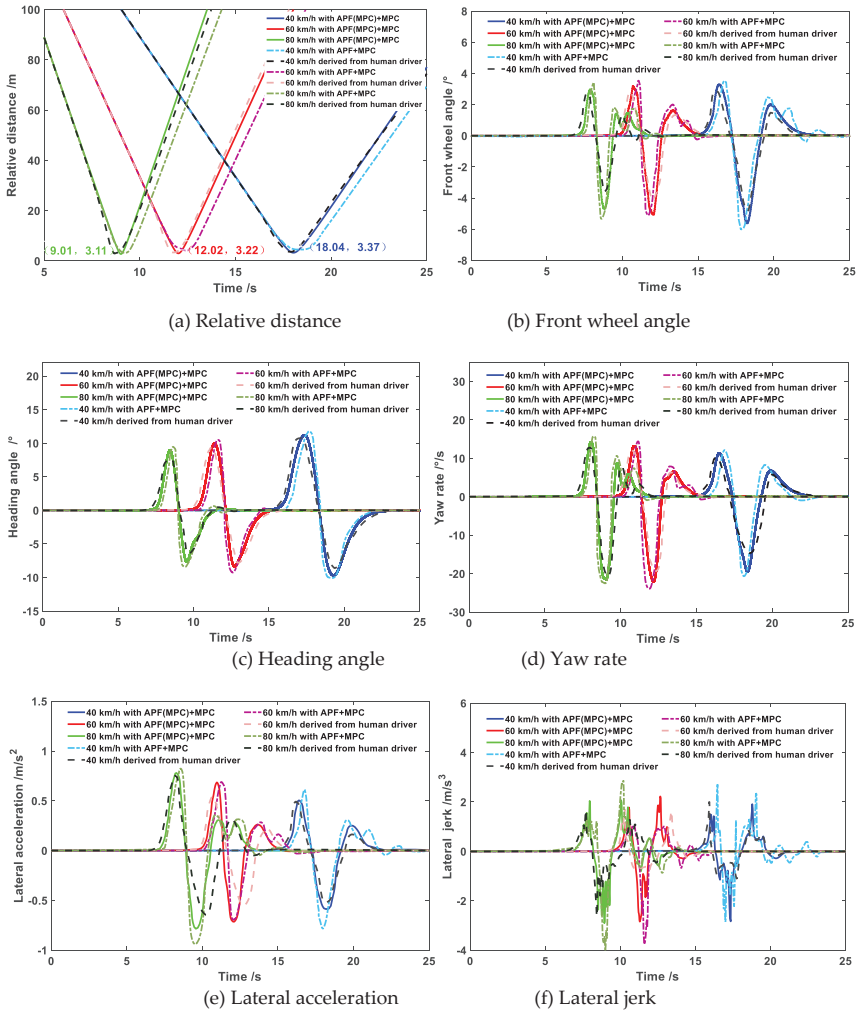


Figure 18. Co-simulation results of vehicle states during the obstacle avoidance process under different speeds.

As shown in Table 2, under different vehicle speeds, the maximum values of the front wheel angle, heading angle, yaw rate, lateral acceleration, and lateral jerk during the obstacle avoidance trajectory tracking process derived from the previous study model (APF+MPC) were obviously greater than that of derived from the proposed model in this study and human drivers. Since the prediction time domain cannot be designed too large in the APF with MPC model, the obstacle avoidance trajectory would possess a smaller longitudinal distance and a larger lateral distance, which would affect the smoothness of the trajectory tracking process. Similarly, larger maximum values of the lateral acceleration and lateral jerk would also reduce the passenger's comfort. Since the APF(MPC) with MPC model proposed in this study combined the APF and MPC in the trajectory planning layer, the trajectory planning controller would take into account the vehicle kinematics constraints in advance, and the additional MPC was equivalent to further improving the model prediction time domain, so that the controller can better simulate the driver's preview behavior. Moreover, too many complex constraints often made it impossible for MPC controller to obtain the optimal solution. The additional MPC in the planning layer could relieve the computational pressure of the MPC algorithm in the trajectory tracking layer. The kinematics and other constraints of the vehicle had been taken into account during the trajectory planning process, and the MPC in the tracking layer can focus on solving the vehicle dynamics constraints, which can improve the effectiveness of the controller in solving the optimal value. Therefore, on the one hand, the results of the longitudinal distance and maximum lateral distance derived from the controller designed in this study were more in accordance with human driver's obstacle avoidance trajectory characteristics in Section 4, and on the other hand, the results of the front wheel angle, heading angle, yaw rate, lateral acceleration, and lateral jerk during the trajectory tracking process derived from the proposed model in this study were more smooth and more human-like, which can effectively improve the acceptance of the autonomous driving system or the intelligent driving system.

The comparison results of the relative distance between the vehicle and obstacle, front wheel angle, heading angle, yaw rate, lateral acceleration, and lateral jerk derived from the APF with MPC model, APF(MPC) with MPC model, and human drivers during the obstacle avoidance process are presented in Figure 18.

As shown in Figure 18a, the minimum distance between vehicle and obstacle derived from the APF(MPC) with MPC model under the speeds of 40 km/h, 60 km/h, and 80 km/h were 3.37 m, 3.22 m, and 3.11 m, respectively; the value from the APF with MPC model were 3.34 m, 3.18 m, and 3.10 m, respectively; the value from the human driver were 3.36 m, 3.21 m, and 3.11 m, respectively. The minimum distances from different models under different speeds were all greater than the safe distance of 2.8 m (the distance of vehicle mass center to the right front corner added the distance of obstacle mass center to the left rear corner), which indicated that the vehicle would keep a reasonably safe distance from the obstacle during the obstacle avoidance process. As shown in Figure 18b–d, the front wheel angle derived from the APF(MPC) with MPC model under all of the different speeds did not exceed 6° , which satisfied the kinematic constraints of the vehicle. The front wheel angle and heading angle decreased with the increase of the vehicle speed, which ensured the smoothness and comfort of the obstacle avoidance process during high speed driving. The range of yaw rate was basically consistent under different speeds, and all of them satisfied the requirements of comfort. However, during the process of changing back to the middle lane, the front wheel angle and yaw rate derived from the APF with MPC model would produce slight vibrations, which would affect the smoothness of the obstacle avoidance trajectory. As shown in Figure 18e,f, the lateral acceleration and lateral jerk improved with the increase of the vehicle speed. Since the longitudinal distance at the beginning of the obstacle avoidance derived from the APF with MPC model was the smallest, the maximum acceleration was the largest and the acceleration changed dramatically, which would affect the smoothness and comfort. In summary, the trajectory planning and tracking controllers designed in this work can satisfy the static obstacle avoidance requirements at different speeds.

The variations of the relevant parameters during the obstacle avoidance process were more human-like, and the avoidance operation was completed on the premise of ensuring smoothness and comfort.

The simulation results with multiple obstacles are presented in Figure 19. The coordinates of the obstacles are (100, 0), (160, 4), (170, -3.75), and (200, 1.8), respectively.

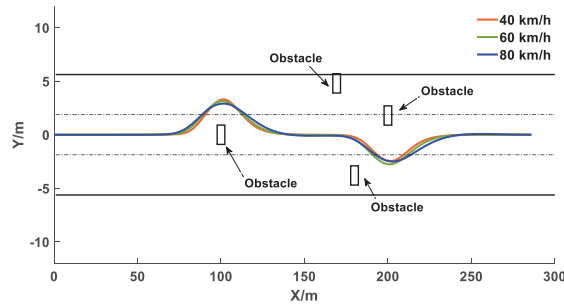


Figure 19. Co-simulation results of obstacle avoidance trajectory tracking under different speeds.

As shown in Figure 19, under different vehicle speeds, the proposed obstacle avoidance controller successfully achieved the goal of avoiding multiple obstacles, and the actual trajectories were smooth and continuous. In addition, there was no phenomenon that the vehicle fell into a local minimum point and the target was unreachable. Therefore, the co-simulation results demonstrated that the proposed trajectory planning controller and the trajectory tracking controller can effectively ensure the safety of obstacle avoidance operations.

6. Conclusions

In this work, an obstacle avoidance trajectory planning controller based on a modified APF algorithm and the MPC algorithm and a trajectory tracking controller based on the linear time-varying MPC algorithm were designed for the AV to realize the active obstacle avoidance function. The modified APF model proposed in this paper added a road boundary repulsive potential field and ameliorated the obstacle repulsive potential field based on the traditional APF model, overcoming some defects of the traditional model. To make the modified APF model satisfy the kinematic constraints of the vehicle, the MPC algorithm was combined with the modified APF model, and a reasonable objective function was constructed to minimize the deviation between the planning trajectory of the modified APF model and the predicted trajectory of the MPC algorithm. Considering that there were many kinds of constraints during vehicle lateral control and for the sake of guaranteeing real-time capability, accuracy, and robustness of the trajectory tracking control algorithm at different speeds, a linear time-varying model predictive trajectory tracking controller was established on the basis of linearizing the vehicle monorail dynamic model. The controller determined the vehicle front wheel angle as the control variable, and multiple constraints of vehicle dynamics and kinematics were combined to design the objective function that could achieve the requirements of fast and accurate tracking of the desired trajectory.

Ameliorating the human-like degree of the planning trajectory is the core of improving the acceptance of the autonomous driving system. Therefore, in this study, a human driver's obstacle avoidance experiment was implemented based on a six-degree-of-freedom driving simulator equipped with multiple sensors, including a steering wheel angle sensor, accelerator pedal sensor, brake pedal sensor, virtual millimeter wave radar sensor, and virtual LIDAR sensor. The obstacle avoidance trajectories under different speeds from different drivers were collected, and the longitudinal distance at the beginning of the obstacle avoidance operation and the maximum distance during the obstacle avoidance process underwent statistical analysis. These two parameters can provide

a basis for the determination of the A value and B value in the elliptical repulsive potential field (shown in Figure 4), making the planned trajectory more human-like.

Finally, a co-simulation model based on CarSim/Simulink was established for the off-line simulation testing of the obstacle avoidance trajectory planning controller and the trajectory tracking controller designed in this study. The co-simulation results demonstrated that the vehicles could smoothly avoid obstacles under different speeds. The results of relevant parameters during the obstacle avoidance process were in accordance with the human drivers' obstacle avoidance trajectory characteristics in Section 4, which indicated that the proposed trajectory planning controller and the trajectory tracking controller were more human-like under the premise of ensuring the safety and comfort of the obstacle avoidance operation.

A few deficiencies in this study need to be improved in the future work. Different road environments may have an impact on driver's obstacle avoidance behavior. A future study will pay close attention to collect the driver's operation data under different road environments and analyze the difference. In addition, the parameters of the obstacle avoidance controller in complex scenarios need to be further optimized.

Author Contributions: Q.S., Y.G. and R.F. conceived of and designed the research; C.W., Q.S. and W.Y. conducted the experiments; Q.S., C.W., Y.G. and R.F. wrote the manuscript. All authors discussed and commented on the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1600500, in part by National Natural Science Foundation of China (51908054, 51775053), in part by the Key Research and Development Program of Shaanxi under Grant (2020GY-163, 2019ZDLGY03-09-02), and in part by the Fundamental Research Funds for the Central Universities, CHD 300102220202.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rosolia, U.; De Bruyne, S.; Alleyne, A.G. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Trans. Control Syst. Technol.* **2016**, *25*, 469–484. [[CrossRef](#)]
2. Villa, J.; Aaltonen, J.M.; Koskinen, K.T. Path-following with lidar-based obstacle avoidance of an unmanned surface vehicle in harbor conditions. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1812–1820. [[CrossRef](#)]
3. Zhang, C.; Chu, D.; Liu, S.; Deng, Z.; Wu, C.; Su, X. Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 19–40. [[CrossRef](#)]
4. Matveev, A.S.; Teimoori, H.; Savkin, A.V. A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance. *Automatica* **2011**, *47*, 515–524. [[CrossRef](#)]
5. Guo, J.; Hu, P.; Wang, R. Nonlinear coordinated steering and braking control of vision-based autonomous vehicles in emergency obstacle avoidance. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3230–3240. [[CrossRef](#)]
6. Chen, J.; Zhao, P.; Liang, H.; Mei, T. Motion planning for autonomous vehicle based on radial basis function neural network in unstructured environment. *Sensors* **2014**, *14*, 17548–17566. [[CrossRef](#)]
7. Li, L.; Ota, K.; Dong, M. Humanlike driving: Empirical decision-making system for autonomous vehicles. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6814–6823. [[CrossRef](#)]
8. Hodge, N.E.; Shi, L.Z.; Trabia, M.B. A distributed fuzzy logic controller for an autonomous vehicle. *J. Robot. Syst.* **2004**, *21*, 499–516. [[CrossRef](#)]
9. Thomas, D.; Kooor, B.C. A genetic algorithm approach to autonomous smart vehicle parking system. *Procedia Comput. Sci.* **2018**, *125*, 68–76. [[CrossRef](#)]
10. Pan, C.Z.; Lai, X.Z.; Yang, S.X.; Wu, M. An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics. *Expert Syst. Appl. An. Int. J.* **2013**, *40*, 1629–1635. [[CrossRef](#)]
11. Xu, H.; Xu, X.; Li, Y.; Zhu, X.; Wang, L. A method for path planning of autonomous robot using A* algorithm. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Shenzhen, China, 12–14 December 2014.
12. Wang, Q.; Wang, G.C.; Fei, F.; Lü, S. Multi objective path finding based on adding-weight a* algorithm. *Adv. Mater. Res.* **2015**, *1079–1080*, 711–715. [[CrossRef](#)]

13. Islam, F.; Narayanan, V.; Likhachev, M. A*-Connect: Bounded suboptimal bidirectional heuristic search. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016.
14. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* **2010**, *29*, 485–501. [[CrossRef](#)]
15. Lavelle, S.M.; Kuffner, J.J. Randomized kinodynamic planning. *IEEE Int. Conf. Robot. Autom.* **2002**, *20*, 378–400.
16. Webb, D.J.; Berg, J.V.D. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
17. Shkolnik, A.; Walter, M.; Tedrake, R. Reachability-guided sampling for planning under differential constraints. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
18. Du, M.; Mei, T.; Chen, J.; Zhao, P.; Tao, X. Rrt-based motion planning algorithm for intelligent vehicle in complex environments. *Robot* **2015**, *37*, 443–450.
19. Tsuji, T.; Tanaka, Y.; Morasso, P.G.; Sanguineti, V.; Kaneko, M. Bio-mimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Trans. Syst. Man Cybern. -Part. C Appl. Rev.* **2003**, *32*, 426–439. [[CrossRef](#)]
20. Matoui, F.; Boussaid, B.; Abdelkrim, M.N. Distributed path planning of a multi-robot system based on the neighborhood artificial potential field approach. *Simulation* **2019**, *95*, 637–657. [[CrossRef](#)]
21. Bahiraei, M.; Heshmatian, S.; Moayedi, H. Artificial intelligence in the field of nanofluids: A review on applications and potential future directions. *Powder Technol.* **2019**, *353*, 276–301. [[CrossRef](#)]
22. Rasekhipour, Y.; Khajepour, A.; Chen, S.K.; Litkouhi, B. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1255–1267. [[CrossRef](#)]
23. Choe, T.S.; Hur, J.W.; Chae, J.S.; Park, Y.W. Real-time collision avoidance method for unmanned ground vehicle. In Proceedings of the International Conference on Control, Automation and Systems, Seoul, Korea, 14–17 October 2008.
24. Bounini, F.; Gingras, D.; Pollart, H.; Gruyer, D. Modified Artificial Potential Field Method for Online Path Planning Applications. In Proceedings of the IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017.
25. Raja, R.; Dutta, A.; Venkatesh, K.S. New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover. *Robot. Auton. Syst.* **2015**, *72*, 295–306. [[CrossRef](#)]
26. Zhang, S.Y.; Shen, Y.K.; Cui, W.S. Path planning of mobile robot based on improved artificial potential field method. *Appl. Mech. Mater.* **2014**, *644–650*, 154–157. [[CrossRef](#)]
27. Kenealy, A.; Primiano, N.; Keyes, A.; Lyons, D.M. Thorough exploration of complex environments with a space-based potential field. In Proceedings of the Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques, San Jose, CA, USA, 18–19 January 2010.
28. Zhao, P.; Chen, J.; Song, Y.; Tao, X.; Xu, T.; Mei, T. Design of a control system for an autonomous vehicle based on adaptive-pid. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 1. [[CrossRef](#)]
29. Birla, N.; Swarup, A. Optimal preview control: A review. *Optim. Control Appl. Methods* **2015**, *36*, 241–268. [[CrossRef](#)]
30. Liu, R.; Duan, J. A path tracking algorithm of intelligent vehicle by preview strategy. In Proceedings of the 32nd Chinese Control Conference, Xi'an, China, 26–28 July 2013.
31. Park, M.W.; Lee, S.W.; Han, W.Y. Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In Proceedings of the 14th International Conference on Control, Automation and Systems, Seoul, Korea, 22–25 October 2014.
32. Benclouf, A.M.; Nguyen, A.T.; Sentouh, C.; Popieul, J.C. Cooperative trajectory planning for haptic shared control between driver and automation in highway driving. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9846–9857. [[CrossRef](#)]
33. Trabia, M.B.; Shi, L.Z.; Hodge, N.E. A Fuzzy Logic Controller for Autonomous Wheeled Vehicles. In *Mobile Robots, Moving Intelligence*; Advanced Robotic Systems International: London, UK, 2006; pp. 175–200.

34. Wu, Y.; Wang, L.; Zhang, J.; Li, F. Path following control of autonomous ground vehicle based on nonsingular terminal sliding mode and active disturbance rejection control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6379–6390. [CrossRef]
35. Velhal, S.; Thomas, S. Improved ltvmpc design for steering control of autonomous vehicle. *J. Phys. Conf. Ser.* **2017**, *783*, 012028. [CrossRef]
36. Kong, J.; Pfeiffer, M.; Schildbach, G.; Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015; pp. 1094–1099.
37. Zanon, M.; Janick, V.; Frasch, M.V. Model Predictive Control of Autonomous Vehicles. *Optim. Optim. Control Automot. Syst.* **2014**, *455*, 41–57.
38. Zhenhai, F.A.G.; Liyong, S.B.J. Optimal preview trajectory decision model of lane-keeping system with driver behavior simulation and Artificial Potential Field. In Proceedings of the IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009.
39. Kumar, P.B.; Rawat, H.; Parhi, D.R. Path planning of humanoids based on artificial potential field method in unknown environments. *Expert Syst.* **2019**, *36*, 1–12. [CrossRef]
40. Giovanni, P.; Osvaldo, B.; Stefano, S.; Luigi, G. An integrated ltv-mpc lateral vehicle dynamics control: Simulation results. *Automot. Model. Predict. Control* **2010**, *402*, 231–255.
41. Snapper, E. Model-based Path Planning and Control for Autonomous Vehicles Using Artificial Potential Fields. Master's Thesis, Delft University of Technology, Delft, The Netherlands, January 2018.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Establishment of the Complete Closed Mesh Model of Rail-Surface Scratch Data for Online Repair

Yanbin Guo ^{1,†}, Lulu Huang ^{1,†}, Yingbin Liu ¹, Jun Liu ² and Guoping Wang ^{1,*}

¹ Hubei Bioinformatics & Molecular Imaging Key Laboratory, Department of Biomedical Engineering, College of Life Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China; guoyanbin@hust.edu.cn (Y.G.); yipiaojun@foxmail.com (L.H.); liuyingbin@hust.edu.cn (Y.L.)

² School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China; liu_jun@whu.edu.cn

* Correspondence: wangguoping@hust.edu.cn; Tel.: +86-136-1863-4882

† These authors contributed equally to this work.

Received: 18 July 2020; Accepted: 19 August 2020; Published: 21 August 2020

Abstract: Rail surface scratching occurs with increasing frequency, seriously threatening the safety of vehicles and humans. Online repair of rail-surface scratches on damaged rails with scratch depths >1 mm is of increased importance, because direct rail-replacement has the disadvantages of long operation time, high manpower and high material costs. Advanced online repair of rail-surface scratch using three-dimensional (3D) metal printing technology such as laser cladding has become an increasing trend, desperately demanding a solution for the fast and precise establishment of a complete closed mesh model of rail-surface scratch data. However, there have only been limited studies on the topic so far. In this paper, the complete closed mesh model is well established based on a novel triangulation algorithm relying on the topological features of the point-cloud model (PCM) of scratch-data, which is obtained by implementing a scratch-data-computation process following a rail-geometric-feature-fused algorithm of random sample consensus (RANSAC) performed on the full rail-surface PCM constructed by 3D laser vision. The proposed method is universal for all types of normal-speed rails in China. Experimental results show that the proposed method can accurately acquire the complete closed mesh models of scratch data of one meter of 50 Kg/m-rails within 1 min.

Keywords: automatic rail-surface-scratch recognition and computation; triangulation algorithm; complete closed mesh model; online rail-repair

1. Introduction

In recent years, the railway industry has developed rapidly in China. The frequency and degree of rail rolling have increased sharply, and rail-surface scratches have become more and more aggravated [1,2]. A rail-surface scratch is a common rail defect caused by the metal plastic deformation due to the friction between wheels and rails and could lead to other surface defects such as peeling and fracture, which may seriously affect the safety of the train [3,4]. Rapid recognition and online-repair of rail-surface scratches is key to ensuring rails remain in good working condition, which is of great importance to safety maintenance in the railway industry [5–7].

According to the provisions of the TG/GW102-2019 in China [8], the existing ways to deal with damaged rails are presented below. Grinding [9,10] and welding [11,12] are the regular maintenance methods for rails with surface scratch depths of less than 1 mm. However, the direct replacement of the damaged rail with a brand-new one is mandatory according to the provisions mentioned above when the surface scratch depth is larger than 1 mm, which is a complicated engineering undertaking with the disadvantages of long operation time, high manpower and material costs, seriously restricting the rapid progress of the railway industry.

Recently, laser cladding, as a novel metal additive manufacturing technology, has achieved marked progress [13,14], showing the capability of repairing damaged rails online [15–20] with the advantages of convenience, high quality of the repair layer and low cost compared with the traditional replacement method. However, the precondition for online rail-repair using the laser cladding technology is to acquire the complete closed mesh model of the scratch data. The existing related works focus mostly on the detection of the damaged rail surface. Min et al. [21] and Wei et al. [22] proposed a rail surface detection system based on machine vision. Jang et al. [23] Shang et al. [24] Faghih-Roohi et al. [25] and Song et al. [26] detected the rail surface damage by using the deep learning methods. In recent years, measurement methods based on 3D laser vision have become widely used for rail surface detection. Zhimin et al. [27] proposed a 3D laser profiling system for detecting the defects on rail surface which integrated a laser scanner, odometer, inertial measurement unit (IMU) and global position system (GPS). Wang et al. [28] presented a novel method based on the local affine invariant feature descriptor for the calibration of distorted profiles obtained by traditional rail measurement system. Cao et al. [29] proposed a defect inspection method of rail surface based on the line-structured light. Santur et al. [30] described a computer-vision based approach allowing for a fast-contactless detection of the rail surface and lateral defects such as fracture, scouring and wear with high accuracy. However, so far very few studies have reported methods for establishing a complete closed mesh model of scratch data which is the precondition of online rail-repair as mentioned above.

In this paper, a method for establishing a complete closed mesh model of the rail-surface scratch data is presented. The process of the method is shown in Figure 1.

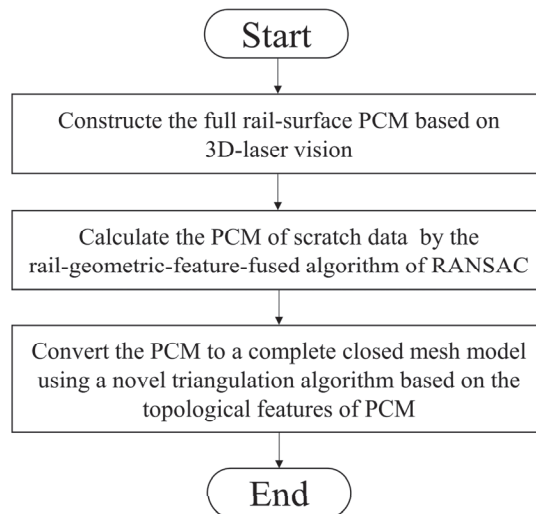


Figure 1. Flowchart of the proposed method.

The main contributions of this paper are presented below:

- (1) A systematic procedure based on a homemade 3D-laser vision system for constructing the PCM is developed.
- (2) An algorithm for calculating the scratch-data PCM is presented. The algorithm is based on the RANSAC [31] fused with rail-geometric-features.
- (3) A novel triangulation algorithm based on the topological features of the PCM is described. The triangulation algorithm can convert the scratch-data PCM to the complete closed mesh model required for the online rail-repair by the laser cladding technology.

- (4) Experiments for verifying the proposed method are carried out. Experimental results show that our method performs well for the acquisition of a complete closed mesh model of the rail-surface scratch data.

The rest of this paper is organized as follows: Section 2 introduces the 3D-laser vision-based procedure for constructing the PCM. Section 3 describes the algorithm for calculating the scratch-data PCM. Section 4 presents the triangulation algorithm based on the point-cloud topology. Section 5 explains the experiments for verifying the proposed method. Section 6 discusses the experimental results. Finally, Section 7 draws the conclusions of the study.

2. 3D Modeling of Rail Surface

The principle of the 3D laser vision adopted in our paper is shown in Figure 2. The laser line moves linearly along the measured object and scans its surface, simultaneously acquiring the point cloud. The details of the procedure are described below.

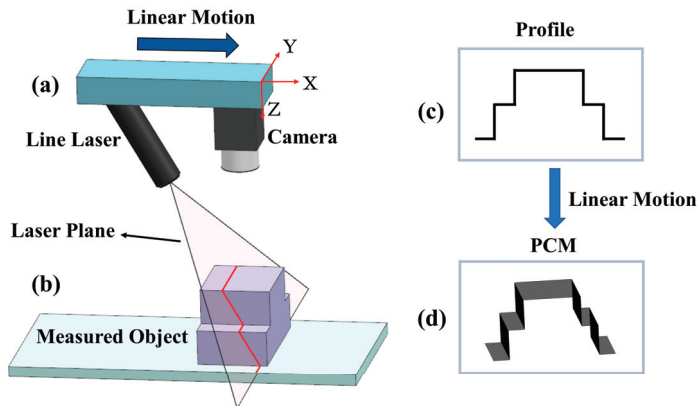


Figure 2. The principle of 3D laser vision. (a) The measurement module comprising the camera and the laser; (b) The laser line on the measured object located on the laser plane; (c) The single point-cloud profile obtained by the calculation using Equations (1) and (2); (d) The PCM formed with a series of point-cloud profiles schemed as in (c).

In order to acquire the single point-cloud profile of the measured object, the calibration is first performed on the laser vision system. The camera model is given by the following equation:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \text{ with } K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where (u, v) is the pixel coordinates in the pixel coordinate system, and (X_C, Y_C, Z_C) is the 3D coordinates in the camera coordinate system (C-CS). The parameters in matrix K are obtained by the classical calibration method described in [32]. Distortion parameters are ignored since a distortionless camera [33] is used in this research.

Additional constraints imposed on the laser line must be considered, since Equation (1) is not sufficient for calculating the 3D coordinates. All points on the laser line are located on the laser plane as shown in Figure 2b, thus the points on the laser line in the C-CS satisfy:

$$aX_C + bY_C + cZ_C + d = 0 \quad (2)$$

where a , b , c and d are the parameters of the laser plane equation in the C-CS, which can be obtained by the planar target calibration method [34].

After the above calibration of the laser vision system, the 3D coordinates in the C-CS of a single point-cloud profile of the measured object (Figure 2c) can be calculated using Equations (1) and (2). A series of point-cloud profiles corresponding to camera frames can also be acquired when the measurement module comprising the camera and the laser (Figure 2a) moves from the origin of the world coordinate system at a constant speed, v , and samples 1 frame every t seconds along the X-axis of the C-CS. The 3D coordinates (X_w, Y_w, Z_w) in the world coordinate system can be converted from the ones in the C-CS as follows:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} X_{ci} \\ Y_{ci} \\ Z_{ci} \end{bmatrix} + \begin{bmatrix} vti \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

where (X_{ci}, Y_{ci}, Z_{ci}) is the 3D coordinates of frame i in the C-CS. The 3D PCM of the entire measured object is acquired after the above operations (Figure 2d).

3. Calculation of the Scratch Data PCM

The first step of acquiring the scratch-data PCM is to accurately recognize the scratch-area of the rail surface PCM, which can differentiate between the damaged and undamaged areas. In this paper, a novel algorithm for the scratch-recognition on rail surface is proposed by combining RANSAC with the geometric features of the rail profile.

According to the hot-rolled rails for railway technology standard GB2585-2007 [35], the geometry of all types of the original undamaged rail can be illustrated as Figure 3. The fixed geometry of the cross section (Figure 3a) can be extended in the direction perpendicular to itself to form the entire rail (Figure 3b).

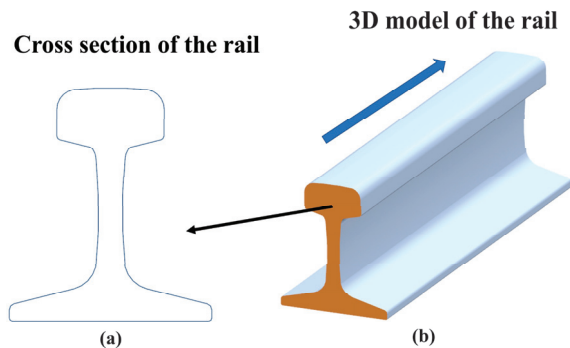


Figure 3. The geometry of the undamaged rail. (a) The cross section of the rail; (b) 3D model of the rail.

As represented in Figure 4, the PCM obtained by 3D scanning of the physical rail using the procedure described in Section 2 comprises a series of equidistant point-cloud profiles. An analysis reveals that all the point-cloud profiles corresponding to the undamaged portions of the rail-surface have the same geometric features as illustrated in Figure 4a, but the ones corresponding to the damaged areas have varying shapes and no uniform geometric features as indicated in Figure 4b. Thus, the point-cloud profiles corresponding to the undamaged areas can be easily and accurately filtered out by using the above observation, resulting in the recognition of scratch-areas.

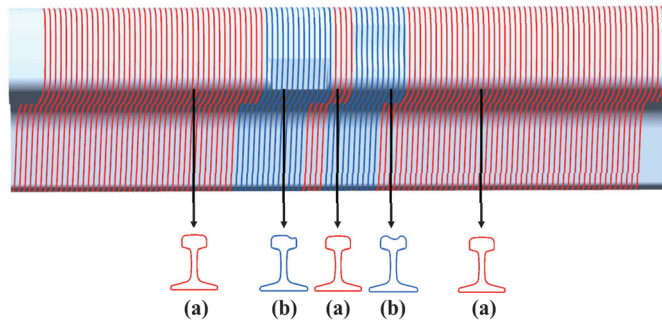


Figure 4. Geometric features of the rail profiles. (a) The point-cloud profiles of the undamaged areas with the same geometric features; (b) The point-cloud profiles of the damaged areas with a variety of shapes and no uniform geometric features.

The mathematical method based on the RANSAC fused with the geometric features of the rail for filtering out the point-cloud profiles on undamaged areas is described as follows. Figure 5 presents an idealized approximation of the geometric features (Figure 5a, Table S1b) by fitting a line segment and two $\frac{1}{4}$ -arcs of the same radii (Figure 5c) using the RANSAC algorithm. A subset of points from the entire point-cloud profile of the rail are randomly selected as the start of RANSAC flow. The selected point-subset is fitted to mathematical models using a least square's procedure to obtain preliminary model parameters, which are subsequently used to calculate the deviation of all 3D points in the point-cloud profile. If the deviation is less than a predetermined threshold, the 3D point is classified to an inlier, otherwise, an outlier. After performing a finite number of iterations of the aforementioned process, the model parameters corresponding to the largest number of inliers can be selected as the best model parameter estimates [36].

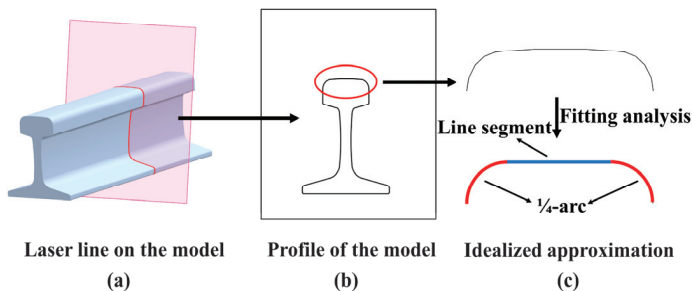


Figure 5. The uniform geometry of the laser profile of the undamaged rail. (a) The laser line on the model; (b) The single profile acquired from (a); (c) The idealized approximation of the profile indicated with the red circle in (b).

In order to recognize the scratch-area of the rail surface PCM, the algorithm is developed based on the above mathematical method as follows:

- (1) Filter the rail surface PCM to remove the noise points and outliers for improving its quality [37] by the method based on the neighborhood radius as shown in Figure 6. Set the radius of the neighborhood as r_n and the minimum number of points in a neighborhood as n . If the minimum number of points in a neighborhood with the radius of r_n is less than n , the point will be filtered out. As shown in Figure 6, the red point and the green point will be filtered out when $n = 2$.

- (2) Split the rail surface PCM into a series of point-cloud profiles and process them separately by the method presented in Figure 7, classifying them into the damaged area or undamaged area, respectively.
- (3) Recognize the scratch-area of the PCM based on the above classification result. A few point-cloud profiles may be not classified correctly, because of errors caused by various reasons, and should be restored as follows. The method shown in Figure 8 is proposed based on the knowledge that the damaged area and the undamaged area on the rail-surface PCM are consecutive within a certain width range. When the sliding window with a certain width scans the profiles of different classified areas, if the majority of profiles belong to the damaged area, then the minority will be re-classified into the damaged area and vice versa as displayed in Figure 8.

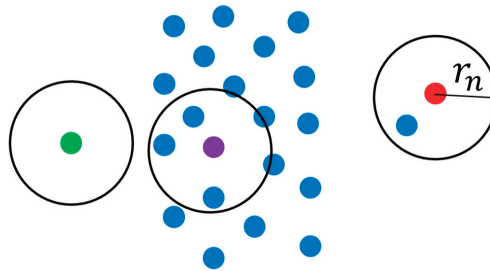


Figure 6. The point cloud filtering method based on the neighborhood radius.

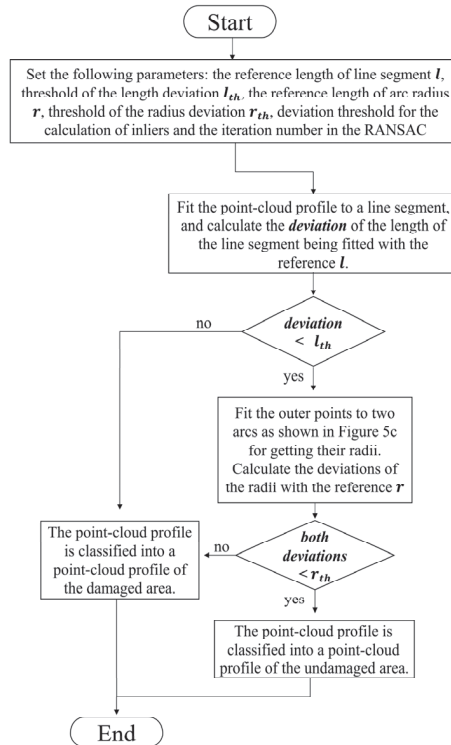


Figure 7. Flowchart of the method for classifying the point-cloud profiles.

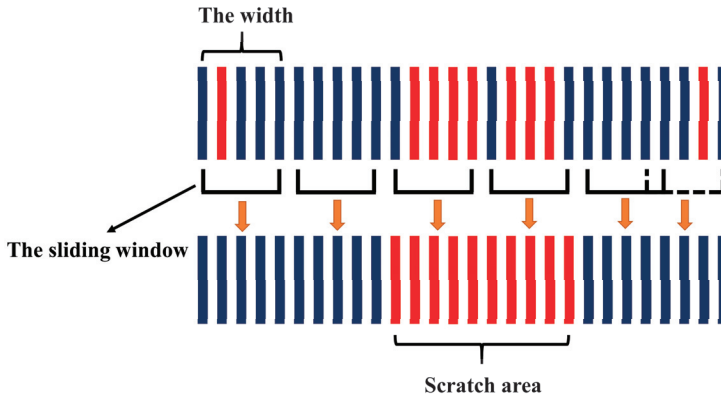


Figure 8. The method for re-classifying profiles by the sliding window with a certain threshold width. The red lines represent the profiles in scratched areas and the blue lines represent the profiles in undamaged areas.

The scratch-surface PCM can be accurately acquired by recognizing the scratch area of the rail-surface PCM with the above-mentioned algorithm, which is subsequently used to calculate the difference with the reference PCM constructed by the method described below.

According to the geometry of the rail presented in Figure 3, the reference PCM can be constructed by extending the undamaged profile in the certain direction. The mathematical formulation is as follows:

$$\begin{bmatrix} X_{SS} \\ Y_{SS} \\ Z_{SS} \end{bmatrix} = \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix} + \begin{bmatrix} x_{\Delta} \frac{n_y}{n_x} \\ x_{\Delta} \frac{n_z}{n_x} \\ x_{\Delta} \frac{n_x}{n_x} \end{bmatrix} \tag{4}$$

where x_{Δ} is the extended step length, $\vec{v} = (n_x, n_y, n_z)$ is the extension vector, (X_S, Y_S, Z_S) is the point of the undamaged profile, and (X_{SS}, Y_{SS}, Z_{SS}) is the constructed point in the reference PCM.

The most critical step for constructing the reference PCM is to accurately calculate the extension vector: $\vec{v} = (n_x, n_y, n_z)$. In Figure 9, the point-cloud profiles on unscratched region are approximated with a line segment of equal length and two 1/4 arcs of equal radii as described previously. The two sets of corresponding endpoints of line segments are fitted respectively with a model of line whose direction is defined as the preliminary extension vector. Thus, two preliminary vectors, $\vec{v}_1 = (n_{x1}, n_{y1}, n_{z1})$ and $\vec{v}_2 = (n_{x2}, n_{y2}, n_{z2})$ are obtained for the calculation of final extension vector, \vec{v} , by using the following equation:

$$\vec{v} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \theta_1 \begin{bmatrix} n_{x1} \\ n_{y1} \\ n_{z1} \end{bmatrix} + \theta_2 \begin{bmatrix} n_{x2} \\ n_{y2} \\ n_{z2} \end{bmatrix} \text{ with } \theta_1 + \theta_2 = 1 \tag{5}$$

where θ_1 and θ_2 are the weighting parameters.

After the final extension vector $\vec{v} = (n_x, n_y, n_z)$ is acquired, the reference PCM can be constructed. Finally, the differences between the reference PCM and the scratch-surface PCM can be calculated and the PCM pair with a difference larger than a certain threshold will be selected to form the PCM of scratch-data.

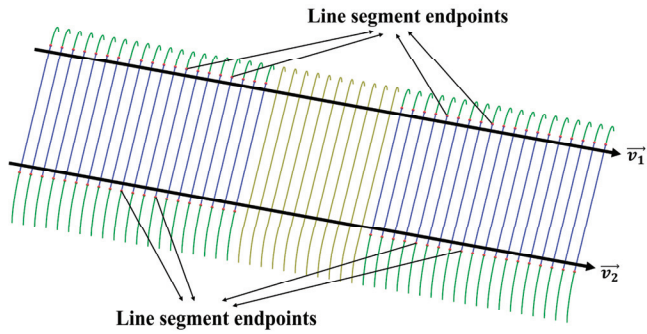


Figure 9. Fitting the endpoints of line segments with a model of line to acquire the preliminary extension vectors \vec{v}_1 and \vec{v}_2 .

4. 3D Point-Cloud Triangulation

The PCM of scratch data obtained in Section 3 cannot be used directly for the online rail-repair relying on the laser cladding technology since it is not a complete closed mesh model and does not meet the requirements for layering of 3D printing [38]. In order to quickly and efficiently achieve a complete closed mesh model, a novel triangulation-algorithm [39] based on the topological features of the PCM is proposed in this section.

Firstly, filtering of the PCM is performed to remove the noise points and outliers caused by the calculation errors using the method presented in Figure 6. Secondly, the filtered-PCM is decomposed into a series of line-profiles equidistant in the X-axis direction since the obtained PCM has an ordered structure (Figure 10) according to the 3D modeling procedure proposed in Section 2. Finally, the triangulation algorithm is implemented as illustrated in Figure 11.

Each point-cloud profile is concatenated with lines as shown in Figure 11a. Then the triangle-meshes between the adjacent point-cloud profiles are sequentially constructed following Figure 11b–d. In Figure 11b, the quadrilaterals are constructed first, then in Figure 11c, triangle-meshes are formed within the above quadrilaterals. In Figure 11d, the remaining points in one of the point-cloud profiles are connected with the end point of the counterpart, finishing the triangulation of the adjacent point-cloud profiles. By carrying out the similar procedure described above between all adjacent point-cloud profiles, the triangulation algorithm for the PCM is completed with a result of Figure 11e.

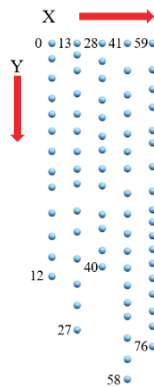


Figure 10. The topological features of the 3D surface PCM labelled with the index numbers.

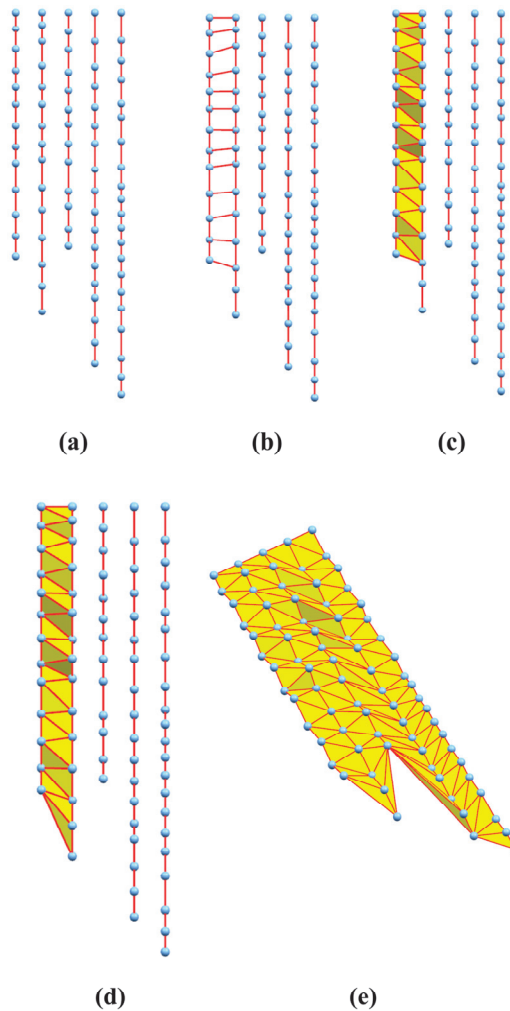


Figure 11. The schematized flow of the triangulation algorithm of 3D point-cloud. (a) Concatenate each point-cloud profile with lines; (b) Construct the quadrilaterals; (c) Form the triangle-meshes; (d) Finish the triangulation of the adjacent point-cloud profiles; (e) Complete the triangulation algorithm for the PCM.

Thus, the triangulation of scratch-data PCM can be done by separately triangulating the reference PCM (Figure 12b) and the scratch-surface PCM (Figure 12c) using the algorithm described above since they compose scratch-data PCM as illustrated in Figure 12a, which also can be found in Section 3. Subsequently, the final complete closed mesh model of the scratch-data (Figure 13c) can be successfully achieved by stitching the triangle-meshes of the reference PCM (Figure 13a) and the scratch-surface PCM (Figure 13b) through the boundary mesh as schemed in Figure 13.

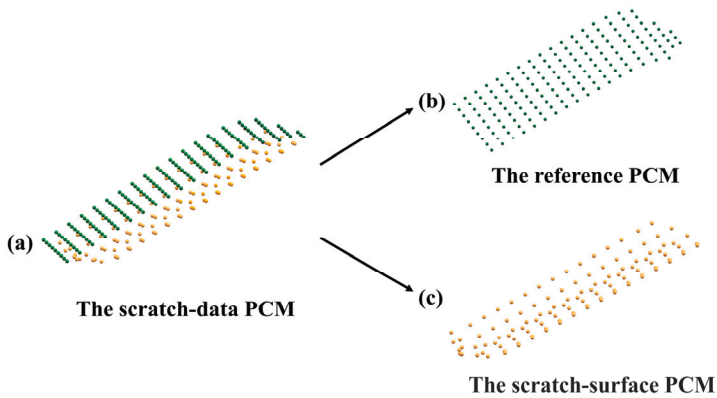


Figure 12. The composition of the scratch-data PCM. (a) The scratch-data PCM; (b) The reference PCM; (c) The scratch-surface PCM.

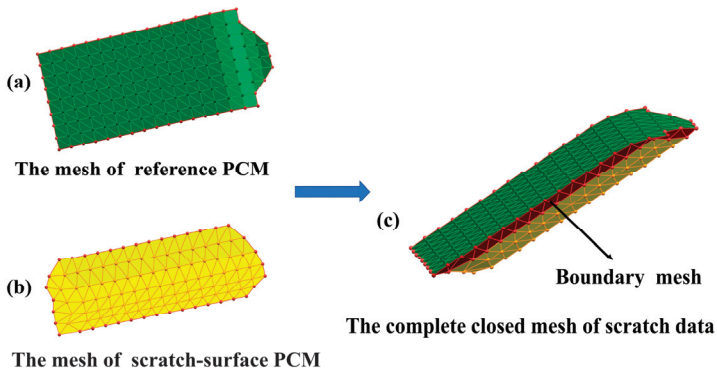


Figure 13. The process of constructing the complete closed mesh model of scratch data. (a) The mesh of reference PCM; (b) The mesh of scratch-surface PCM; (c) The complete closed mesh model of scratch data stitched by (a,b) through the boundary mesh.

5. Experiment

Experiments are carried out on both the artificial rail and a practical rail to verify the proposed method. All algorithms are run on a computer whose specifications are listed in Table 1.

Table 1. The specifications of the computer used in the experiment.

| Parameter | Value |
|----------------------|---------------------|
| Memory size | 16 GB |
| CPU type | Intel Core i5-9400F |
| GPU type | NVIDIA RTX2060 |
| Graphics memory size | 6 GB |

The homemade 3D laser vision system proposed in Section 2 is used for constructing the PCM of the rail-surface. The main specifications of the line laser and the camera used in the system are listed in Tables 2 and 3 respectively. The parameters in the system mentioned in Section 2 are listed in Table 4.

Table 2. The main specifications of the line laser.

| Parameter | Value |
|--------------------|-----------------------|
| Type | ZLM5AL650-16GD0.15 |
| Overall dimension | 16 mm × 16 mm × 70 mm |
| Power | 5 mw |
| Wavelength | 650 nm |
| Minimum line width | 0.15 mm |

Table 3. The main specifications of the camera.

| Parameter | Value |
|--------------------|-----------------------|
| Type | MV-GE134GC-T-CL |
| Overall dimension | 29 mm × 29 mm × 40 mm |
| Pixel size | 1,300,000 |
| Resolution | 1280 × 1024 |
| Maximum frame rate | 91 FPS |

Table 4. The parameters in the 3D laser vision system.

| Parameter | Value |
|--|---|
| The intrinsic matrix of the camera after calibration | $K = \begin{bmatrix} 691.69732 & 0 & 366.62759 \\ 0 & 691.79845 & 241.07942 \\ 0 & 0 & 1 \end{bmatrix}$ |
| The laser plane equation | $0.006X_C - 1.05453Y_C - Z_C + 478.982 = 0$ |
| The speed of the measurement module | $v = 0.04$ m/s |
| The sampling frequency of the camera | 80 HZ |
| The sampling interval | $t = 1/80 = 0.0125$ s |

The artificial 50 Kg/m-rail presented in Figure 14a is used in the first experiment. The length of the rail is 1 m, the scanning time for obtaining the data is $1/0.04 = 25$ s and the time of data processing to form the 3D PCM of the rail-surface (Figure 14b) is 10.63 s.

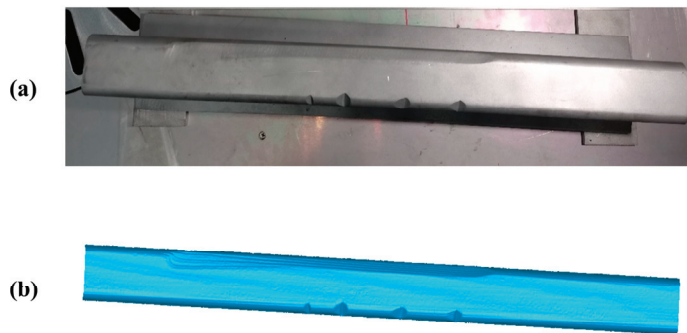


Figure 14. The artificial rail of 50 Kg/m and its surface PCM. (a) The artificial damaged-rail; (b) The surface PCM of the artificial damaged-rail.

The scratch-recognition algorithm presented in Section 3 is performed on the PCM of the rail-surface. There is no need to carry out the filtering algorithm because of the fine quality of the PCM as shown in Figure 14b. A series of point-cloud profiles are generated by splitting the rail-surface PCM and subsequently processed separately following the method described in Figure 7 with the values of the parameters listed in Table 5. The classification result of the point-cloud profiles is reflected in

Figure 15a, where the red profiles and the blue profiles are classified into the damaged area and the undamaged area, respectively. A few point-cloud profiles incorrectly classified are restored with the method expressed in Figure 8 in which the width of the sliding window is set as 5 profiles. Figure 15b indicates the scratch-surface PCM which is the final result of the scratch-recognition algorithm with the total running time of 1.27 s.

Table 5. The values of the parameters mentioned in Figure 7.

| Parameter | Value |
|-------------------------------|--------|
| l | 50 mm |
| l_{th} | 1 mm |
| r | 15 mm |
| r_{th} | 1 mm |
| Deviation threshold in RANSAC | 0.2 mm |
| Iteration number in RANSAC | 1000 |

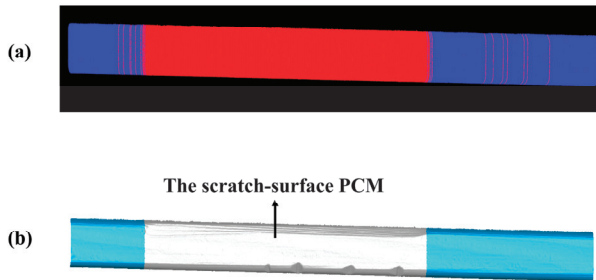


Figure 15. The result of the scratch-recognition algorithm performed on the rail-surface PCM. (a) The classification result of the point-cloud profiles displaying the damaged area (red) and the undamaged area (blue); (b) The scratch-surface PCM identified by the algorithm as indicated in the gray-white area.

The reference PCM can easily be acquired by extending an undamaged point-cloud profile selected from the rail-surface PCM using the Equation (4) with the extending step length $x_{\Delta} = 0.5$ mm. The extension vector in Equation (4) can be calculated from preliminary vectors (see Figure 9) using the Equation (5) with $\theta_1 = 0.5$, $\theta_2 = 0.5$. Table 6 displays the calculation results of the parameters mentioned here.

Table 6. The calculation results of the extension vector.

| Vector | Value |
|-------------|--------------------------------|
| \vec{v}_1 | (0.999923, 0.011243, 0.005168) |
| \vec{v}_2 | (0.999936, 0.010394, 0.004364) |
| \vec{v} | (0.999930, 0.010819, 0.004766) |

After the above calculations, the reference PCM is constructed as seen in Figure 16a. Then, the depth-difference between the reference PCM and the scratch-surface PCM shown in Figure 16b is calculated. If the depth-difference is larger than the set threshold of 0.2 mm, the reference PCM and the scratch-surface PCM are selected to construct the original scratch-data PCM shown in Figure 16c. The total time for acquiring the scratch-data PCM starting from the reference PCM construction is 3.23 s.

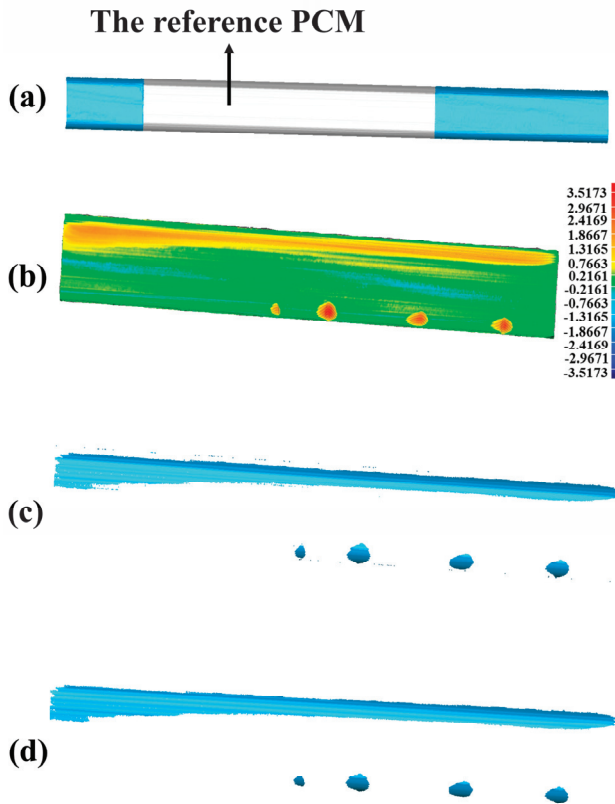


Figure 16. The acquisition of the scratch-data PCM. (a) The result of constructed reference PCM; (b) The depth-difference between the reference PCM and the scratch-surface PCM; (c) The original scratch-data PCM with noise points; (d) The filtered scratch-data PCM.

The 3D triangulation-algorithm proposed in Section 4 is performed on the scratch-data PCM. The filtering of the PCM is firstly done before the formal triangulation based on the method of neighborhood radius presented in Figure 6 with $r_n = 4$ mm and $n = 50$, leading to the result of Figure 16d. Then, the triangulation of the reference PCM (Figure 17a) and the scratch-surface PCM (Figure 17b) are finished by using the algorithm presented in Figure 11. Finally, a complete closed mesh model required for laser cladding technology as shown in Figure 18a (the magnified details presented in Figure 18b) is well established by stitching the triangle-meshes of the reference PCM and the scratch-surface PCM through the boundary mesh, which is the end of the whole experiment. The triangulation-algorithm costs 5.22 s in our experiment. Table 7 lists the time required for each step in the experiment and the total time is 45.35 s.

The second experiment is carried out on the practical 50 Kg/m-rail presented in Figure 19a to further verify the reliability of the proposed method. The final complete closed mesh model of the practical damaged rail is established by the same method described in the first experiment, which is shown in Figure 19b (the magnified details presented in Figure 19c). The total time for the second experiment is 47.51 s. The detailed process of the second experiment which is similar to the one presented above for the artificial rail, is well described with Figures S1–S5 and Tables S1–S3 included in the supplementary materials.



Figure 17. The triangulation of the PCM. (a) The triangle-meshes of the reference PCM; (b) The triangle-meshes of the scratch-surface PCM.

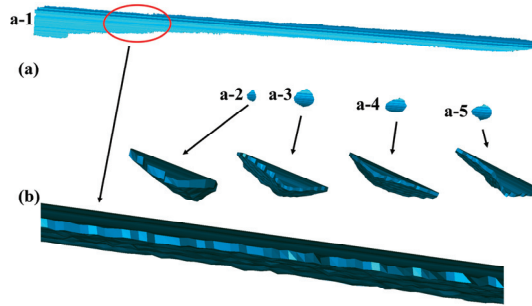


Figure 18. The final complete closed mesh models of the scratch-data of the artificial damaged rail. (a) Five complete closed mesh models corresponding to five scratch-data; (b) The local magnified model of a-1 and the full magnified ones of a-2, a-3, a-4, a-5, respectively.

Table 7. The time required in the experiment for the artificial damaged rail.

| Time | Value |
|------------------------|---------|
| Scanning time | 25 s |
| 3D PCM constructing | 10.63 s |
| Scratch-recognition | 1.27 s |
| Scratch-data acquiring | 3.23 s |
| 3D triangulation | 5.22 s |
| Total time | 45.35 s |

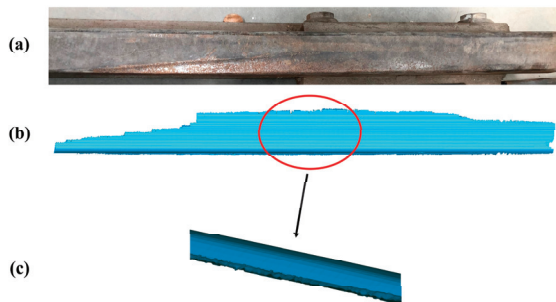


Figure 19. The practical damaged rail and final result in the second experiment. (a) The practical damaged-rail; (b) The final complete closed mesh model of the practical damaged rail; (c) The local magnified model of b.

6. Discussion

In the first experiment for the artificial damaged rail, Figure 16b obviously shows that the scratch-depth of most rail surface is larger than 1 mm and the specific places of scratch-depth > 1 mm is presented in Figure 20. Table 8 displays the further analysis result in the form of max depth, min depth and Root Mean Square (RMS) acquired by Geomagic Studio 2014 (64 bit). The counterpart results in the second experiment for the practical damaged rail are presented in Figure 21, Figure 22, and Table 9, respectively. Based on the above results, both the artificial rail and the practical rail in the experiment should be replaced with new rails, which is mandatory required by the provisions of the TG/GW102-2019 in China.

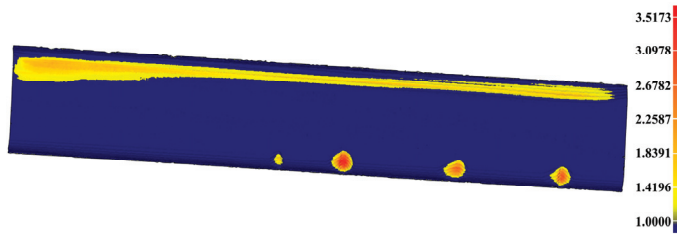


Figure 20. The specific places on the artificial rail surface with scratch-depth larger than 1 mm.

Table 8. The analysis result of the artificial damaged rail.

| Max Depth | Min Depth | RMS |
|-----------|------------|-----------|
| 3.5173 mm | -0.7009 mm | 0.5579 mm |

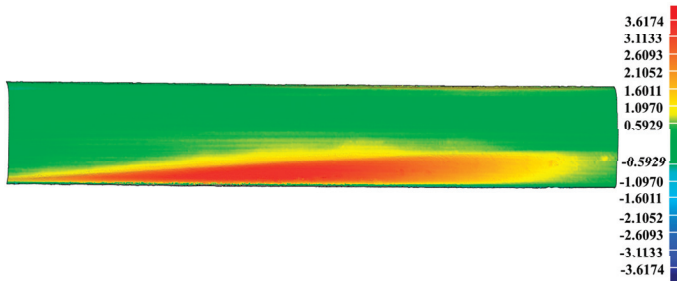


Figure 21. The depth-difference between the reference PCM and the scratch-surface PCM on the practical damaged rail.

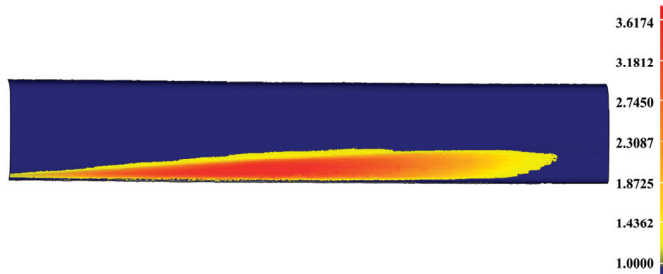


Figure 22. The specific places on the practical damaged rail surface with scratch-depth larger than 1 mm.

Table 9. The analysis result of the practical damaged rail.

| Max Depth | Min Depth | RMS |
|-----------|------------|-----------|
| 3.6174 mm | −0.7363 mm | 1.0520 mm |

In order to verify the accuracy of the complete closed mesh models of the scratch-data acquired in the experiment, the virtual repairs of the rails by cladding the scratch-area with the models are carried out, leading to the results of Figures 23a and 24a, which are corresponding to the artificial rail and the practical rail, respectively. The difference calculated between the repaired artificial rail model and the reference model is shown in Figure 23b and further analyzed in Table 10. The similar results for the practical rail are presented in Figure 24b and Table 11.

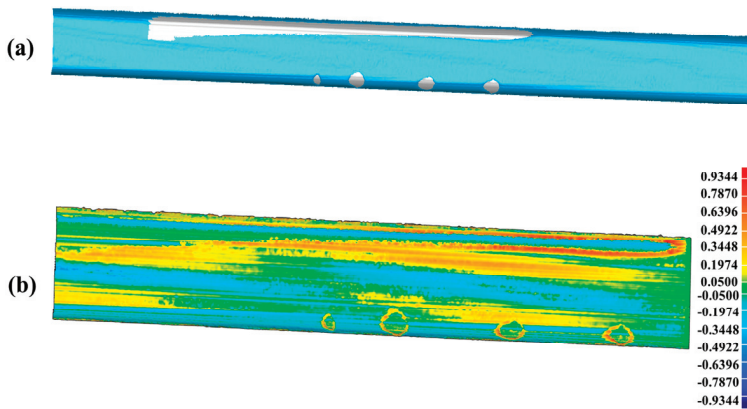


Figure 23. The accuracy analysis of the scratch-data acquired in the experiment for the artificial damaged rail. (a) The result of virtual repair of the artificial rail by using the scratch-data; (b) The difference between the repaired artificial rail model and the reference model indicating the scratch-depth on the repaired artificial rail is less than 1 mm.

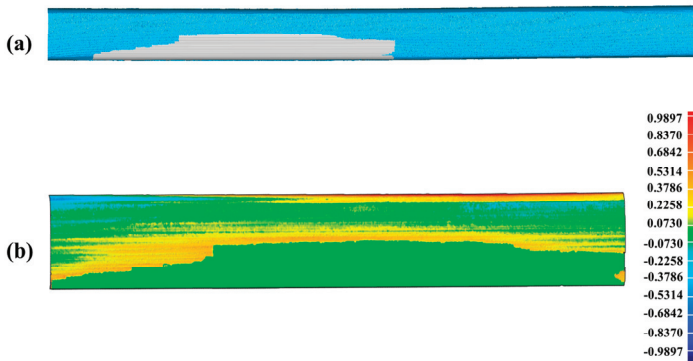


Figure 24. The accuracy analysis of the scratch-data acquired in the experiment for the practical damaged rail. (a) The result of virtual repair of the practical rail by using the scratch-data; (b) The difference between the repaired practical rail model and the reference model indicating the scratch-depth on the repaired practical rail is less than 1 mm.

Table 10. The analysis result of the repaired artificial rail.

| Max Depth | Min Depth | RMS |
|-----------|------------|-----------|
| 0.9344 mm | −0.7009 mm | 0.1928 mm |

Table 11. The analysis result of the repaired practical rail.

| Max Depth | Min Depth | RMS |
|-----------|------------|-----------|
| 0.9897 mm | −0.7363 mm | 0.1824 mm |

The above results show that the scratch-depth on the repaired rails is less than 1 mm, which well meets the requirements in the provisions of the TG/GW102-2019 in China, proving that the complete closed mesh model of the scratch-data established herein is precise enough for the use of online repair. Also, the total time required in the experiments (45.35 s for the artificial rail and 47.51 s for the practical one) is much less than 1 min, fully demonstrating the crucial capability of real-time required by online rail-repair based on the laser cladding technology, so the method proposed in our paper is practical for the online repairing of damaged rails in terms of accuracy and real-time performance.

7. Conclusions

In this paper, an efficient and accurate method is well proposed for the establishment of the complete closed mesh model of rail-surface scratch data, solving the precondition of online rail-repair based on the laser cladding technology. Related experiments are performed on both the artificial rail and the practical rail, and the corresponding results reveal the capability of real-time and accuracy required by the online rail-repair, which could further promote the development of the field.

Supplementary Materials: The following are available online at <http://www.mdpi.com/1424-8220/20/17/4736/s1>, Figure S1: The practical rail of 50 Kg/m and its surface PCM, Figure S2: The result of the scratch-recognition algorithm performed on the rail-surface PCM of the practical rail, Figure S3: The acquisition of the scratch-data PCM, Figure S4: The triangulation of the PCM, Figure S5: The final complete closed mesh models of the scratch-data of the practical rail, Table S1: The values of the parameters used in the scratch-recognition algorithm, Table S2: The calculation results of the extension vector, Table S3: The time required in the experiment.

Author Contributions: Conceptualization, Y.G. and G.W.; funding acquisition, G.W.; investigation, Y.L. and J.L.; methodology, Y.G., L.H. and G.W.; project administration, G.W.; software, Y.G. and L.H.; writing—original draft, Y.G., L.H. and G.W.; writing—review & editing, Y.G. and G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61875062 and in part by the Fundamental Research Funds for the Central Universities.

Acknowledgments: The authors would like to thank Professor Xiaoyan Zeng at Wuhan National Laboratory for Optoelectronics, Huazhong University of Science & Technology, for providing the practical damaged rail.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cannon, D.F.; Edel, K.O.; Grassie, S.L.; Sawley, K. Rail defects: An overview. *Fatigue Fract. Eng. Mater. Struct.* **2003**, *26*, 865–887. [\[CrossRef\]](#)
2. Bao, X.D. Urban rail transit present situation and future development trends in China: Overall analysis based on national policies and strategic plans in 2016–2020. *Urban Rail Transit* **2018**, *4*, 1–12. [\[CrossRef\]](#)
3. He, C.G.; Chen, Y.Z.; Huang, Y.B.; Liu, Q.Y.; Zhu, M.H.; Wang, W.J. On the surface scratch and thermal fatigue damage of wheel material under different braking speed conditions. *Eng. Fail. Anal.* **2017**, *79*, 889–901. [\[CrossRef\]](#)
4. Wei, T.J.; Liu, L.Y.; Li, J.Y.; Zuo, X. Analysis the reason of passenger line's rail scratch. *J. Railw. Sci. Eng.* **2015**, *12*, 489–494.
5. Yuan, D.; Lu, Z.; Zhang, J.; Li, X.; Ma, T. Integrative design of an emergency resource predicting-scheduling-repairing method for rail track faults. *IEEE Access* **2019**, *7*, 155686–155700. [\[CrossRef\]](#)

6. Ravitharan, R. Safer rail operations: Reactive to proactive maintenance using state-of-the-art automated in-service vehicle-track condition monitoring. In Proceedings of the 2018 IEEE International Conference on Intelligent Rail Transportation (ICIRT), Singapore, 12–14 December 2018; pp. 1–4.
7. Filograno, M.L.; Guillen, P.C.; Rodriguez-Barrios, A. Real-time monitoring of railway traffic using fiber bragg grating sensors. *IEEE Sens. J.* **2012**, *12*, 85–92. [[CrossRef](#)]
8. State Railway Administration. *Rules on Maintenance of General Speed Rail Track*; China Railway Publishing House: Beijing, China, 2019; pp. 40–46.
9. Uhlmann, E.; Lypovka, P.; Hochschild, L.; Schröder, N. Influence of rail grinding process parameters on rail surface roughness and surface layer hardness. *Wear* **2016**, *366*, 287–293. [[CrossRef](#)]
10. Sroba, P.; Roney, M. Rail grinding best practices. In Proceedings of the AREMA Annual Conference, Chicago, IL, USA, 5–8 October 2003; p. 63.
11. Mortazavian, E.; Wang, Z.; Teng, H. Repair of light rail track through restoration of the worn part of the railhead using submerged arc welding process. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 3315–3332. [[CrossRef](#)]
12. Jun, H.K.; Seo, J.W.; Jeon, I.S. Fracture and fatigue crack growth analyses on a weld-repaired railway rail. *Eng. Fail. Anal.* **2016**, *59*, 478–492. [[CrossRef](#)]
13. Seede, R.; Shoukr, D.; Zhang, B.; Whitt, A.; Gibbons, S.; Flater, P.; Elwany, A.; Arroyave, R.; Karaman, I. An ultra-high strength martensitic steel fabricated using selective laser melting additive manufacturing: Densification, microstructure, and mechanical properties. *Acta Mater.* **2020**, *186*, 199–214. [[CrossRef](#)]
14. Liu, J.; Yu, H.; Chen, C.; Weng, F.; Dai, J. Research and development status of laser cladding on magnesium alloys: A review. *Opt. Lasers Eng.* **2017**, *93*, 195–210. [[CrossRef](#)]
15. Zhu, S.; Chen, W.; Ding, L.; Zhan, X.; Chen, Q. A mathematical model of laser cladding repair. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 3265–3278. [[CrossRef](#)]
16. Mortazavian, E.; Wang, Z.; Teng, H. Thermal-mechanical study of 3D printing technology for rail repair. In Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition, Pittsburgh, PA, USA, 9–15 November 2018; p. V002T02A052.
17. Mortazavian, E.; Wang, Z.; Teng, H. Thermal-kinetic-mechanical modeling of laser powder deposition process for rail repair. In Proceedings of the ASME 2019 International Mechanical Engineering Congress and Exposition, Salt Lake City, UT, USA, 11–14 November 2019; p. V02AT02A052.
18. Lai, Q.; Abrahams, R.; Yan, W.; Qiu, C.; Mutton, P.; Paradowska, A.; Soodi, M. Investigation of a novel functionally graded material for the repair of premium hypereutectoid rails using laser cladding technology. *Compos. Part B Eng.* **2017**, *30*, 174–191. [[CrossRef](#)]
19. Zheng, Z.; Qi, H.; Zhang, H.; Ren, D. Laser repairing of damaged steel rail with filler wire. In Proceedings of the Sixth International Conference on Measuring Technology & Mechatronics Automation, Zhangjiajie, China, 10–11 January 2014; pp. 383–386.
20. Seo, J.W.; Kim, J.-C.; Kwon, S.J.; Jun, H.K. Effects of laser cladding for repairing and improving wear of rails. *Int. J. Precis. Eng. Manuf.* **2019**, *20*, 1207–1217. [[CrossRef](#)]
21. Min, Y.; Xiao, B.; Dang, J.; Yue, B.; Cheng, T. Real time detection system for rail surface defects based on machine vision. *EURASIP J. Image Video Process.* **2018**, *2018*, 1–13. [[CrossRef](#)]
22. Wei, D.; Wei, X.; Liu, Y.; Jia, L.; Zhang, W. The identification and assessment of rail corrugation based on computer vision. *Appl. Sci.* **2019**, *9*, 3913. [[CrossRef](#)]
23. Jang, J.; Shin, M.; Lim, S.; Park, J.; Kim, J.; Paik, J. Intelligent image-based railway inspection system using deep learning-based object detection and weber contrast-based image comparison. *Sensors* **2019**, *19*, 4738. [[CrossRef](#)]
24. Shang, L.; Yang, Q.; Wang, J.; Li, S.; Lei, W. Detection of rail surface defects based on CNN image recognition and classification. In Proceedings of the 20th IEEE International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea, 11–14 February 2018; pp. 45–51.
25. Faghieh-Roohi, S.; Hajizadeh, S.; Núñez, A.; Babuska, R.; De Schutter, B. Deep convolutional Neural Networks for detection of rail surface defects. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2584–2589.
26. Song, Y.; Zhang, H.; Liu, L.; Zhong, H. Rail surface defect detection method based on yolov3 deep learning networks. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018.

27. Zhimin, X.; Qingquan, L.; Qingzhou, M.; Qin, Z. A 3D laser profiling system for rail surface defect detection. *Sensors* **2017**, *17*, 1791.
28. Wang, C.; Ma, Z.; Li, Y.; Zeng, J.; Jin, T.; Liu, H. Distortion calibrating method of measuring rail profile based on local affine invariant feature descriptor. *Measurement* **2017**, *110*, 11–21. [[CrossRef](#)]
29. Cao, X.; Xie, W.; Ahmed, S.M.; Li, C.R. Defect detection method for rail surface based on line-structured light. *Measurement* **2020**, *159*, 107771. [[CrossRef](#)]
30. Santur, Y.; Karaköse, M.; Akın, E. Learning based experimental approach for condition monitoring using laser cameras in railway tracks. *Int. J. Appl. Math. Electron. Comput.* **2016**, *4*, 1–5. [[CrossRef](#)]
31. Derpanis, K.G. Overview of the RANSAC Algorithm. *Image Rochester N.Y.* **2010**, *4*, 2–3.
32. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *22*, 1330–1334. [[CrossRef](#)]
33. Wu, Y.; Li, A.; Chen, C.; Shen, J.; Bian, Y.; Zheng, Z. A design method of the distortionless catadioptric panoramic imaging based on freeform surface. In Proceedings of the SPIE/COS Photonics Asia, Beijing, China, 13–17 October 2014; p. 9273.
34. Zhou, F.Q.; Zhang, G.J. Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations. *Image Vis. Comput.* **2005**, *23*, 59–67. [[CrossRef](#)]
35. Wang, L.M. Summary of the standard GB2585-2007 about hot rails for railway. *Metall. Stand. Qual.* **2008**, *46*, 1–8.
36. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum* **2007**, *26*, 214–226. [[CrossRef](#)]
37. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L. A review of algorithms for filtering the 3D point cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [[CrossRef](#)]
38. Attene, M. As-exact-as-possible repair of unprintable STL files. *Rapid Prototyp. J.* **2018**, *24*, 855–864. [[CrossRef](#)]
39. Niantao, L.; Bingxian, L.; Guonian, L.; A-Xing, Z.; Liangchen, Z. A delaunay triangulation algorithm based on dual-spatial data organization. *PFG J. Photogram. Remote Sens. Geoinform. Sci.* **2019**, *87*, 19–31.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Data-Driven Analysis of Bicycle Sharing Systems as Public Transport Systems Based on a Trip Index Classification

Mark Richard Wilby ¹, Juan José Vinagre Díaz ^{1,*}, Rubén Fernández Pozo ¹, Ana Belén Rodríguez González ¹, José Manuel Vassallo ² and Carmen Sánchez Ávila ¹

¹ Group Biometry, Biosignals, Security, and Smart Mobility, Departamento de Matemática Aplicada a las Tecnologías de la Información y las Comunicaciones, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense 30, 28040 Madrid, Spain; mrwilby@etsit.upm.es (M.R.W.); ruben.fernandez@upm.es (R.F.P.); abrodriguez@etsit.upm.es (A.B.R.G.); carmen.sanchez.avila@upm.es (C.S.Á.)

² Grupo de Investigación en Planificación del Transporte, Transport Research Centre (TRANSyT), Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos, Universidad Politécnica de Madrid, 28040 Madrid, Spain; josemanuel.vassallo@upm.es

* Correspondence: juanjose.vinagre@upm.es

Received: 9 July 2020; Accepted: 30 July 2020; Published: 2 August 2020

Abstract: Bicycle Sharing Systems (BSSs) are exponentially increasing in the urban mobility sector. They are traditionally conceived as a last-mile complement to the public transport system. In this paper, we demonstrate that BSSs can be seen as a public transport system in their own right. To do so, we build a mathematical framework for the classification of BSS trips. Using trajectory information, we create the *trip index*, which characterizes the intrinsic purpose of the use of BSS as *transport* or *leisure*. The construction of the trip index required a specific analysis of the BSS shortest path, which cannot be directly calculated from the topology of the network given that cyclists can find shortcuts through traffic lights, pedestrian crossings, etc. to reduce the overall traveled distance. Adding a layer of complication to the problem, these shortcuts have a non-trivial existence in terms of being intermittent, or short lived. We applied the proposed methodology to empirical data from BiciMAD, the public BSS in Madrid (Spain). The obtained results show that the trip index correctly determines transport and leisure categories, which exhibit distinct statistical and operational features. Finally, we inferred the underlying BSS public transport network and show the fundamental trajectories traveled by users. Based on this analysis, we conclude that 90.60% of BiciMAD's use fall in the category of transport, which demonstrates our first statement.

Keywords: bicycle sharing systems; public transport systems; data-driven classification of trips; BSS underlying network; trip index

1. Introduction

Bicycle sharing systems (BSSs) are one illustrative example of the new sharing economy, which is exponentially increasing at present, with great impact on mobility [1]. At first instance, they have arisen as a perfect complement for public transport in cities, covering the last-mile segment [2]. In this respect, BSSs provide the convenience of unidirectional short hops, which enhance the connections to other transport modes: bus, metro, train, etc. Under this perspective, BSSs are just conceived as supplementary to the main transport network. However, BSSs can also be considered a transport mode in their own right, which contributes to the overall objective of reducing car use [3]. This reduction implies a direct positive impact on environment and health [4].

BSSs show specific characteristics that make them different from most transport modes: their use. Cars, buses or trains are primarily used as a means of moving people from one place to another. These trips are performed to reach the workplace, the residence, a commercial area, a sport center, or a tourist site, among others. In this respect, we can observe that the purpose of these trips is mainly defined by their destination. BSSs are fundamentally different as the users' purpose is inherent to the trip: not only can they be performed to reach a destination, but also trips themselves can be a form of sport or tourism, for example [5].

With greater understanding of their usage patterns, BSSs could be optimized, tailoring them to each city and neighborhood. The current knowledge about BSS usage is still embryonic and the need to identify sub-modes of the different mobility activities taken by BSS users is an essential step in this process. However, as our understanding increases, so will our design criteria and planning. BSSs can be re-purposed and redeployed with very little capital investment and virtually no infrastructure change. These small budget requirements will allow the expansion of BSSs, which offers a high degree of granularity, making them an extremely useful tool in the upgrading of mobility services.

Human mobility is a complex problem. Fortunately, some aspects of this complexity assist in the problem analysis: (i) people move according to regular patterns [6]; and (ii) they return to a small number of places [7]. To move, people choose public transport or private vehicles considering measurable factors, such as travel and transit times or economic cost [8], and other non-measurable factors, like convenience, safety, comfort, or environment [9].

This extensive set of factors complicates the task of modeling human mobility and, in particular, the election of a BSS as a transport mode. In the absence of data, we must rely on probabilistic methods [10]. However, in this case, although not complete, we do have access to some information. BSSs often provide data regarding the occupancy of docking stations, which we can employ to infer the demand between them [11]. From this information we can then extract a set of behavioral profiles for each docking station, such as *morning peak*, *morning arrival*, *flat* or *anomalous* [12].

Presently, 4th generation BSSs provide trajectory data, which allow us to construct more accurate profiles [13]. These trajectories can also be extracted from other by-products like call detail records (CDR) collected by phone companies [14]. However, this approach is often biased by the specific population that formed the dataset [15].

In this work we analyze trajectory data from BiciMAD, a 4th generation BSS in Madrid, Spain, and create a data-driven methodology to classify trips into two main groups depending on their intrinsic purpose: transport and leisure. We understand as *transport trips* those performed to reach a specific destination in the shortest possible time. This reflects that the main purpose of this category of trips is getting to the destination to develop certain activity like working, studying, shopping, etc. On the other hand, we understand as *leisure trips* those performed to wander around the city or as sport. In this respect, the fundamental objective of leisure trips is not the destination itself, but the route traveled along a sequence of points of interest (tourism) or the overall length of the path (sport).

In addition, this methodology allows us to infer the underlying public transport network of a BSS. Most transport modes are composed by a permanent set of nodes (stations or stops) and a structure of fixed edges [16], i.e., the specific routes of buses or trains. On the contrary, a BSS has no fixed edges given that users can cycle following whichever route they prefer through the road system, parks, bicycle lanes, etc. This makes the extraction of the underlying transport network a challenging task. Our methodology provides a rigorous way of inferring the BSS transport network from trajectory data, thus contributing to reach a complete understanding of the overall multiplex public transport network [17] of a city.

Consequently, the main contributions of this work are:

- The creation a quantitative framework to classify BSS trips as *transport* or *leisure*.
- The definition of a distance-based index that builds the basis for this classification of trips.
- The mathematical characterization of the shortest path distance in a BSS, considering the set of shortcuts that bikers can use in their routes.

- The application of this framework to classify trips in a real BSS.
- Statistical and operational analysis to confirm the validity of the obtained results.
- The extraction the underlying BSS public transportation network.

In the following sections we first propose a mathematical framework to classify BSS trips (Section 2). Next, in Section 3 we present and validate the results of applying this framework to a real dataset, providing the methodology we used to calculate the different variables it involves. From the basis of the resulting classification of trips, Section 4 shows the underlying transport and leisure networks we extract from the real trajectories falling on each category. Finally, Section 5 summarizes the work and discusses its benefits for municipalities and BSS managers. Please note that this structure does not include a specific section devoted to analyzing scientific works in the field. Instead we have placed each reference in the particular section that required it. This way the reader can easily follow the text eliminating the need for returning to a previous section to check each reference.

2. Data-driven Classification of Trips

The present section is devoted to describing the mathematical framework that builds the basis of the method of trip classification we propose. First (Section 2.1) we set up the starting premise upon which we will construct the trip classification methodology. This starting premise is based on observing trajectories and detecting how close they are to the shortest path. Then we create a trip index as the fundamental metric, which compares the actual trajectory to a reference (Section 2.2). However, this reference is not unique as BSS users can find shortcuts to reduce their overall traveled distance; thus, we study the resulting spaces and trajectories in Section 2.3. Finally, in Section 2.4 we define the shortest path applicable to a BSS and the subsequent trip index.

2.1. Starting Premise

Our first purpose is to construct a methodology and a mathematical framework that allow us to classify BSS trips depending on their intrinsic purpose. In this respect, we can observe four types of BSS users [5]: (a) commuters: who move between residence and work place, or secondary transport node; (b) utility users: who need to reach commercial, cultural or sport facilities; (c) leisure users: who cycle for fun and sport; and (d) tourists: who visit tourist sites and attractions. To reach our main objective, we define two categories of BSS trips: *transport*, which includes group types (a) and (b); and *leisure*, which include group types (c) and (d). Please note that we do not classify *users*, but *trips* as our final goal is to characterize the BSS mobility and its intrinsic network. In this sense, a determined user can always perform trips that fall in either category.

A trip is a sequence of locations and time stamps. From this set of spatio-temporal points we can obtain different variables that describe the trip, like speed and length. Considering the former, we could expect the speed of leisure trips to be lower than that of transport trips. However, speed may not be adequate for this purpose as it depends on factors like age, physical abilities, steepness of the road, or traffic. On the other side, the length of the trip is intrinsically impacted by the separation between origin and destination, thus it cannot directly reflect the purpose of the journey as an absolute value.

So, let us observe the trajectory of the trip. Research in public transportation systems assume that users apply some utility function to their decision to choose this transport mode [18,19]. Consequently, we build our framework based on this starting premise: transport trips describe trajectories close to the shortest path.

2.2. Trip Index

Our objective is to define a *trip index* α that allows us to classify trips into transport or leisure. To calculate this index, we will rely on trajectories. In this respect, note that trajectories do not need to be identical to describe a specific type of trip; they just have to share some common semantic meaning, which in our case, will be given by the trip index. Following our starting premise, the trip index must represent the deviation of the user's actual trajectory from the shortest path.

In other scientific disciplines, for example Biology and Hydraulic, researchers have faced similar problems in describing animal's movements searching for food [20] and the course of rivers through valleys [21] respectively. These works make use of the *sinuosity index*, SI , which relates the distance of the linear and actual paths:

$$SI = \frac{d_p}{d_L},$$

where d_p is the length of the actual trajectory and d_L is the Euclidean distance between origin and destination.

However, linear trajectories are not viable in cities most of the time, because of the intrinsic topology of the infrastructure. Thus, we must take the shortest possible path as the base reference. Considering these premises, we define the trip index α_p of a trip p from origin i to destination j as:

$$\alpha_p = \frac{d_{SP}(i, j)}{d_p}, \quad (1)$$

where d_p is the actual distance traveled in trip p and d_{SP} is the length of the shortest viable path from origin to destination. However, the peculiarities of a BSS insert a level of complexity in calculating this shortest path.

2.3. Spaces, Trajectories and Shortcuts in a BSS

To represent trips in a BSS system, we must consider two spaces. On one hand, BSSs are deployed on a *real physical space*, with constraints. Some of these constraints are immutable, such as buildings or rivers; and others are subject to change, like one-way streets or shared lanes (which cars and bicycles can simultaneously use). On the other hand, this real physical space with constraints defines an *underlying graph*, which represents the actual allowed travel space. The underlying graph continuously evolves in time as it is affected by mutable constraints.

In addition, given the particularities of BSSs, we have to contend with the fact that the underlying graph is not complete, and we must also consider missing connections in the real space. This is because cyclists can navigate through some restrictions using, for example, pedestrian crossings or sidewalks. We will collectively refer to these extra graph elements as *shortcuts*. Furthermore, the existence of these shortcuts has a complex time dependency as they may be impacted by factors that make them viable or not at specific moments. For example, consider the possibility of crossing a street on a traffic light; only when it is green, this shortcut becomes a viable alternative to reduce the minimum distance. Thus, shortcuts can be observed as graph edges with a time-to-live.

Considering these premises, we can define four different trajectories regarding a determined BSS trip:

- The linear trajectory, with length d_L , i.e., the direct Euclidean distance between origin and destination, which only depends on the real physical space.
- The retrievable shortest path between origin and destination given the underlying graph, which we will refer to as the *orthodox* trajectory, with length d_O .
- The shortest path between origin and destination using the set of available shortcuts, namely the *heterodox* trajectory, with length d_H .
- The actual path of the trip traveled by the user, with length d_p .

2.4. Characterizing the Shortest Path in a BSS Trip

Given a trip p , the Euclidean distance d_L from origin i to destination j defines the length scale to the problem: the relative size of graph links to the Euclidean distance from origin to destination.

On the other hand, the orthodox trajectory is the one that minimizes the overall distance traveled from origin to destination, using the corresponding city infrastructure (streets and bicycle lanes) and respecting the regulation applicable to bicycles. This orthodox trajectory does not embed the heterodoxy that results from inserting shortcuts in the graph, which effectively reduces d_O .

In addition, we must take into account that these shortcuts evolve in time. This variation implies that we cannot aim at defining a *fixed* shortest path, but a *set of viable* shortest paths. Effectively, we need some way of characterizing the statistical distribution of this set of available shortest paths.

The problem of directed paths is intrinsic to a large number of different disciplines, polymer physics in the presence of forces, percolation theory, direct random walks, to name but a few. Using some of the more basic concepts from these fields we can infer what the basic structure of the probability distribution of shortest paths could be. This distribution function is built from a collection of paths fixed to have the same start and end points, which are separated by d_L . In effect, we are considering a distribution function constructed from a set of paths defined by a fixed extension. The actual distribution will be complex, where details of its form will depend on the nature of the paths available, but we are proposing that the grosser characteristics are defined by only a limited number of parameters. The simplest spatial representation of this statistical distribution of paths is that of an ellipsoid with origin and destination located on its vertexes. The width of the ellipsoid is dictated by allowed variance and frequency of deviations from the direct path. Consequently, we obtain a crude characterization of this distribution by looking at the defining geometry of the ellipse. Hence, the solution to our problem revolves around the calculation of the major and minor axes of this ellipsoid. The former is the linear trajectory, with length d_L . Therefore, let us concentrate on the latter.

The minor axis reflects the distribution width caused by deviations from the linear path of the actual trip. We can find a set of approaches within the scientific literature in the field of urban mobility that estimate deviations from trajectories using map-matching methods or path-searching problems [22]. These approaches often aim at coupling a set of GPS data points to one of the possible trajectories a vehicle can describe through the underlying city's graph. Among them, researchers have used methods like the A* search algorithm [23], the Manhattan Distance [24], or the Hausdorff distance [25].

However, our problem is intrinsically different as we need to characterize the distribution function of a set of viable and time-dependent shortest paths. This set of shortest paths presents an absolute minimum, the linear path. Consequently, we can evaluate deviations using a similarity metric to the linear path [26]. For this purpose, we chose the Fréchet distance [27], which measures the similarity between two curves, considering the location and order of points along them. Intuitively, imagine a person walking a dog with a leash. The person and the dog describe finite paths f and g respectively. Both can vary their speed, but they cannot walk back. The Fréchet distance between curves f and g could be seen as the minimum length of the leash that allows these two trajectories.

Formally [28], let (M, d) be a metric space; we define a *curve* as a continuous mapping:

$$f: [v_0, v_1] \rightarrow M, \quad v_0, v_1 \in \mathbb{R}, \quad v_0 \leq v_1.$$

Given two curves $f: [v_0, v_1] \rightarrow M$ and $g: [w_0, w_1] \rightarrow M$, their *Fréchet distance* is defined as:

$$\delta_F(f, g) = \inf_{\substack{v: [0, 1] \rightarrow [v_0, v_1] \\ w: [0, 1] \rightarrow [w_0, w_1]}} \max_{t \in [0, 1]} d(f(v(t)), g(w(t))),$$

where v and w are arbitrary continuous nondecreasing functions, with $v(0) = v_0$, $v(1) = v_1$, $w(0) = w_0$, and $w(1) = w_1$ and t is the path parameter, typically time.

For trajectories formed as a collection of points rather than a continuous function, we can use the discrete Fréchet distance, also called the *coupling distance* [29]. It is an approximation of the Fréchet metric for polygonal curves. Following a similar intuitive example mentioned before, the discrete Fréchet distance replaces the man and the dog with a pair of leaping frogs.

Using the formal definition in [29], let P and Q be two polynomial curves with endpoints of their line segments forming the sequences $\sigma(P) = (v_1, v_2, \dots, v_p)$ and $\sigma(Q) = (w_1, w_2, \dots, w_q)$. A coupling C between P and Q is a sequence of distinct pairs of endpoints in $\sigma(P) \times \sigma(Q)$ taken in order:

$$C = ((v_{a_1}, w_{b_1}), (v_{a_2}, w_{b_2}), \dots, (v_{a_m}, w_{b_m})).$$

The first and last pairs are formed by the two origins and two destinations, respectively. The remaining pairs are formed in an iterative sequence. In each step, we couple the previous or the next endpoints in $\sigma(P)$ to the previous or the next in $\sigma(Q)$. This method implies that there is not a unique coupling; every particular combination of endpoints will generate a coupling C_k with $k \in [1, K]$, whose length is determined by the longest distance between every pair of endpoints, i.e.

$$\|C_k\| = \max_{i=1,2,\dots,m} d(v_{a_i}, w_{b_i}).$$

Finally, the *discrete Fréchet distance* between the polygonal curves P and Q is defined as the minimum length among all the possible K couplings between them:

$$\delta_{DF}(P, Q) = \min_{k \in [1, K]} \{\|C_k\|\}. \quad (2)$$

We use the discrete Fréchet distance to calculate the maximum deviation of the orthodox trajectory to the linear trajectory, i.e., $\delta_{DF}(O, L)$. Considering this maximum deviation, the length of the heterodox trajectory lies between those two distances, i.e., $d_H \in [d_L, d_O]$. Thus, we define the length of the heterodox trajectory as the minimum distance between origin and destination, given a deviation of $\delta_{DF}(O, L)$ from the linear trajectory, which is calculated as:

$$d_H = 2\sqrt{(\delta_{DF}(O, L))^2 + \frac{d_L^2}{4}} = \sqrt{4(\delta_{DF}(O, L))^2 + d_L^2} \quad (3)$$

We will take d_H as an approach to the shortest available path between the origin i and destination j of a particular trip p . Consequently, the trip index defined in Equation 1 is finally defined as:

$$\alpha_p = \frac{d_H}{d_p} = \frac{\sqrt{4(\delta_{DF}(O, L))^2 + d_L^2}}{d_p} \quad (4)$$

The trip index in Equation 4 incorporates all the available information about a trip and the possible trajectories in the underlying physical and topological spaces of the BSS. Following the premise stated in Section 2.1, we will use the trip index to classify trips as leisure or transport.

3. Application to a Real BSS

We applied the trip index to a dataset of real trips performed in BiciMAD, the public BSS in the city of Madrid, Spain. In this section, we will describe the dataset and the methodology we followed to obtain indexes and classify the trips accordingly. Finally, we will analyze the characteristics of the resulting groups of transport and leisure trips to validate the classifier under both statistical and operational perspectives.

3.1. Dataset

BiciMAD is a public BSS managed by the Empresa Municipal de Transportes (EMT) within the Municipality of Madrid (Spain). BiciMAD uses electric bicycles and includes 169 docking stations currently in operation.

We used data from February 2019 accounting for near 500 MB carrying information corresponding to 303 962 trips. Each trip includes the following data:

- Time stamp: pick up time with 1 hour definition, for privacy and anonymity issues.
- User's identifier: unique encrypted identifier of user, refreshed daily.
- Type of user: annual, eventual, staff.
- User's range of age: 6 intervals [0,16], [17,18], [19–26], [27–40], [41–65], [66,∞), and *unknown*.
- Identifier of the origin docking station.
- Identifier of the destination docking station.

- Travel time: time from pick up to drop off.
- Track: collection of geographical coordinates ordered in time recorded on a 1-minute basis during the trip.

3.2. Applying the Mathematical Framework to the Dataset

To apply the mathematical framework, we developed in Section 2 to this dataset, we need to perform a sequence of tasks that we next describe.

3.2.1. Preprocessing

We first carry out a preprocessing phase to guarantee the quality of the data we employ. Errors in the database are mainly due to the low accuracy of GPSs in an urban scenario, which results in either inconsistent positions or time slots with no recorded location. Furthermore, there are trips performed by staff members that we do not consider in the study.

We apply this preprocessing to 303 962 trips connecting 25 818 origin and destination pairs resulting in 142 968 movements and 22 929 origin-destination (OD) pairs.

In addition, we filter out those trips that start and end on the same docking station as they cannot be considered to be transport. In this filtering we treat as *twin* docking stations those that are located at the same point of interest. These trips follow their own internal dynamics and are also intention-based. They are possibly retrieval trips (target destination and return). They could also be tourist/leisure-based activity, or they could just represent an interpretation of the tariffing, where the user considers it more effective to retain the bike for whatever reason. Such paths should be considered separately, but with no knowledge of the destination, they cannot be used to analyze the transport aspects of the service. In any case, this restriction forces us to work on the worst-case scenario for our analysis.

The remaining set includes 139 956 trips and 22 750 OD pairs.

3.2.2. Calculation of the Trip Index

For a given trip p from origin i to destination j , the calculation of the trip index α_p defined in Equation 4 involves the lengths of four trajectories: the actual trajectory of the trip, d_p ; the linear trajectory, d_L ; the orthodox trajectory, d_O ; and the heterodox trajectory, d_H .

The actual trajectory of the trip is retrieved from the track data in the BiciMAD database. The linear trajectory is built as the segment connecting the origin i to the destination j . On the other hand, the orthodox trajectory is retrieved from Google Maps, selecting *bicycle* as the transport mode. This way, the orthodox trajectory is defined by a polygonal curve that includes n intermediate points.

To calculate the Fréchet distance from the orthodox (O) to the linear (L) trajectories, we construct a linear interpolation of n segments that link origin i to destination j . Next, we apply the calculation of the discrete Fréchet distance $\delta_{DF}(O, L)$ specified in Equation 2 to these two polygonal curves.

Finally, the length of the heterodox trajectory is calculated from the linear and the orthodox trajectories as defined in Equation 3.

3.3. Results of the Classification of BSS Trips

We represented the classification of BSS trips as a data-driven problem given the lack of a previously well-defined set of trips falling on either category: transport or leisure. Consequently, we classify BSS trips according to the results obtained from applying the trip index to the specific set of empirical data, and analyzing their inherent features.

Figure 1 shows the histogram of the trip indexes corresponding to the 139 956 trips registered in February 2019, in BiciMAD, resulting from the preprocessing phase. We observe two clear behaviors of the trip index, which suggests a concatenation of uniform and Gaussian distributions.

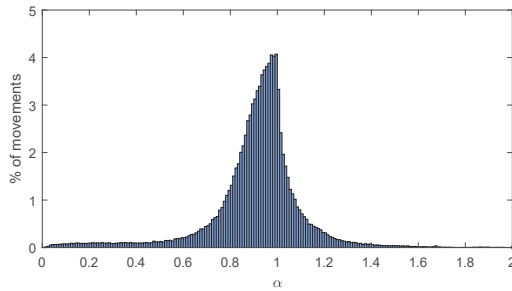


Figure 1. Histogram of trip indexes of trips in the dataset.

Please note that there are trip indexes greater than 1, which correspond to trips achieving deviations under the Fréchet distance of the orthodox trajectory to the linear path. This fact is perfectly consistent with the proposed methodology. On the other hand, it opens up a new set of questions regarding the actual statistical distribution of the shortest available routes between two points in a BSS, to be specifically addressed in a particular research line that will be discussed in Section 5.

To determine the optimal threshold that separates each of these two behaviors, we use the *elbow method*, commonly employed in data-driven clustering models. In this respect, Figure 2 shows the trip indexes sorted in ascending order.

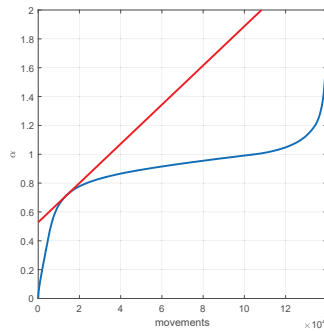


Figure 2. In blue: trip indexes sorted in ascending order; in red: the tangent line obtained by the elbow method.

The optimal value for the threshold of the trip index is $\alpha^* = 0.7$. We then use this value to classify trips as transport or leisure.

3.4. Validation of the Results

The characterization of mobility often lacks a solid ground-truth [30]. In these cases, we must rely on the analysis of a set of features shown by each resulting group and test whether they actually represent distinct behaviors.

Consequently, in this work, we perform two different analyses to validate the classification methodology based on the trip index: statistical and operational.

3.4.1. Statistical Analysis

First, we complete the statistical analysis of three defining features of the trip: distance, duration, and speed. Figure 3 shows the results we obtained.

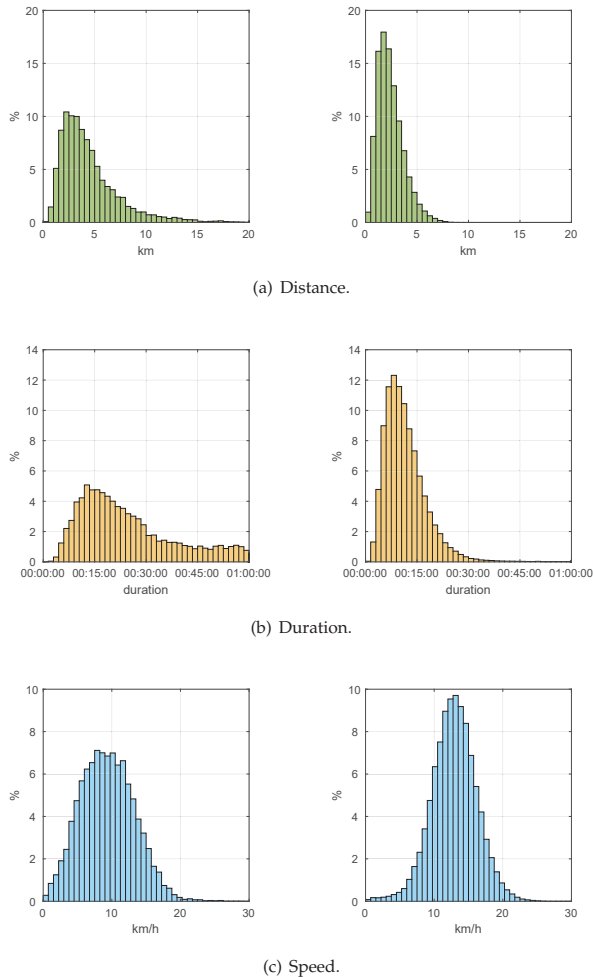


Figure 3. Statistical characterization of trips: *leisure* ($\alpha_p < \alpha^*$) on the left, and *transport* ($\alpha_p \geq \alpha^*$) on the right.

We can observe that the shapes of the statistical distributions corresponding to *leisure* (left) and *transport* (right) trips are fundamentally different. Let us study in some depth each of these features and compare them to similar scientific results obtained from BSSs.

Traveled distances in leisure and transport trips are clearly different as they show particular ranges, average values and standard deviations as we can observe in Table 1. Leisure trips are characterized by significantly higher maximum distances, which reflect the user is *wandering*. This result agrees with the conclusions in [31] where authors state that *daily members* of a BSS (*leisure*) perform longer trips as opposed to *annual members* (*transport*). In addition, the average distances we obtained fit those resulting from the *non-commute* and *commute* trips in [32].

Table 1. Distance Statistics (km).

| | Min. | Max. | Mean | Std. Dev. |
|---|------|------|------|-----------|
| Leisure ($\alpha_p < \alpha^*$) | 0.3 | 47.9 | 4.5 | 3.0 |
| Transport ($\alpha_p \geq \alpha^*$) | 0.1 | 10.0 | 2.4 | 1.2 |

The specific statistical values of the distribution of the duration of the trip are included in Table 2. In this case, the ranges are similar, but we can observe significant differences among the average and standard deviations. Transport trips are typically shorter in time, which coincides with the results in [32], where the authors conclude that duration of BSS trips is highly correlated with distance. In addition, the authors in [33] affirm that commuters spend 1 hour a day total in their outward and return trips; in our study, Figure 3b shows that the tail of the distributions of duration of trips is negligible beyond 30 min.

Table 2. Travel Time Statistics.

| | Min. | Max. | Mean | Std. Dev. |
|---|----------|----------|----------|-----------|
| Leisure ($\alpha_p < \alpha^*$) | 00:02:22 | 05:57:57 | 00:37:20 | 00:38:22 |
| Transport ($\alpha_p \geq \alpha^*$) | 00:00:58 | 05:59:26 | 00:11:56 | 00:09:52 |

Finally, Table 3 shows the numerical results for the distribution of speeds. We can observe a significantly higher average speed in transport trips. This is particularly relevant as bicycles in BiciMAD are electric, which avoids the speed being biased by the physical condition of the user. Higher speeds were also observed in commuters in [32].

Table 3. Speed Statistics (km/h).

| | Min. | Max. | Mean | Std. Dev. |
|---|------|------|------|-----------|
| Leisure ($\alpha_p < \alpha^*$) | 0.3 | 28.7 | 9.4 | 3.9 |
| Transport ($\alpha_p \geq \alpha^*$) | 0.2 | 29.4 | 12.9 | 3.3 |

3.4.2. Operational Analysis

In addition to the statistical analysis, we perform a second validation on the results of the classification of trips. In this case, we focus on the docking stations that generated or absorbed most of the trips of each kind.

Figure 4 shows the top 10 docking stations that acted as origin (red circles) or destination (green squares) of leisure (Figure 4a) and transport (Figure 4b) trips. Leisure trips have destinations located in Madrid's historic center and some important tourist sites like Puerta de Alcalá (right most green square); Madrid Río (left most green square), a 7-kilometers linear park along the Manzanares river; Matadero de Madrid (bottom most green square) that offers avant-garde cultural exhibitions and performance; or Mercado de Maravillas (top most green square), one of the biggest markets in Europe. On the other side, leisure trips have origins at docking stations mainly located on the historic center and close surroundings. This matches a typical behavior of tourists, who often choose accommodations close to the tourist sites. However, 40% of the top origins and destinations of leisure trips do not share a common docking station; this means that they can be seen as one-way trips. This behavior is opposite to what we observe in transport trips (Figure 4b), where we find 9 docking stations that are simultaneously origin and destination. This matches the usual commuter behavior: from residence to work and back.

We recognized this same behavior on the docking stations activity profile [34], i.e., the net flow of bicycles. Selecting 2 as the number of groups to be generated by the proposed clustering algorithm resulted in two distinct classes of docking stations corresponding to commuter and non-commuter behaviors. In the present work, we reach analogous profiles facing the problem from a different perspective: trajectories instead of docking stations.

This operational analysis confirms that the classification effectively separates trips of either category.

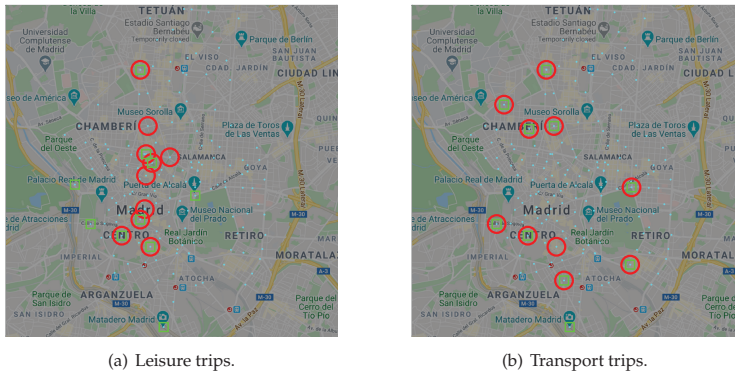


Figure 4. Main origins (red circles) and destinations (green squares) of trips.

4. Underlying BSS Public Transport Network

A public transport system relies on a multiplex network in which different transport modes complement one another. An optimally designed public transport network should provide full coverage, which ideally would connect any pair of points in the space [35]. BSSs provide a public transport network with virtually full coverage, as users can travel the city in a single transport mode. In addition, it may contribute to reduce the travel time avoiding traffic congestions and crowded stops [36]. Furthermore, BSSs add a second functionality to the network: leisure, which is not found in other transport modes.

However, the flexibility in choosing routes throughout a BSS complicates the extraction of its underlying network. The classification methodology we propose allows us to identify the set of trips that were performed using the BSS as a transport mode or just for leisure. Consequently, we can derive the underlying BSS public transport network from the trajectories of these trips and study their fundamental purpose.

Figure 5 shows the trajectories of the 50 OD pairs that account for the greater number of leisure and transport trips. Leisure trips (Figure 5a) required 61 docking stations in order to reach this top 50 OD pairs, while transport trips (Figure 5b) needed only 40. Merging these two particular networks, we can infer the overall BSS transport network.

As we can observe, leisure and transport networks show significant differences in structure: dispersion, compactness, loose links, etc. To highlight these differences, let us consider the complete set of leisure and transport trips, which form the underlying network of each category. These two networks can be analyzed as graphs, calculating their density. The corresponding results are shown in Table 4.

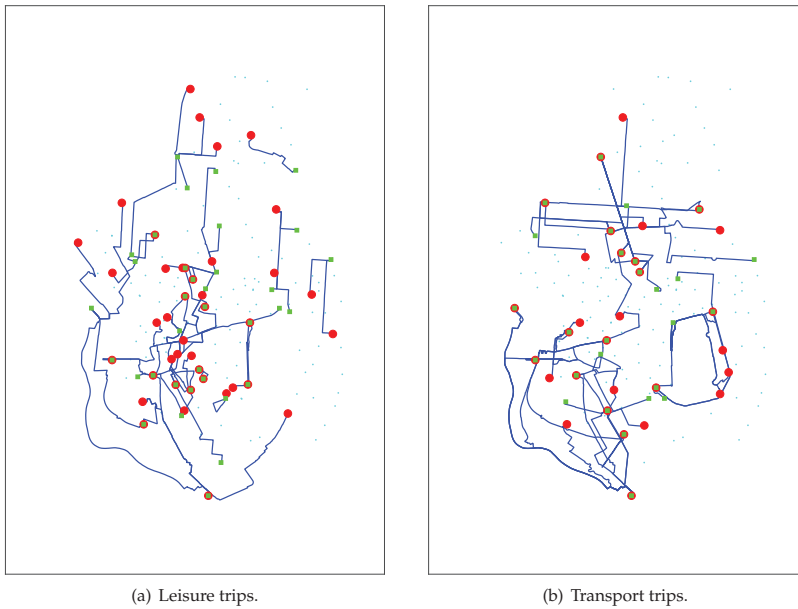


Figure 5. Main leisure and transport trajectories.

Table 4. Comparison of Networks.

| | Total | Leisure | Transport |
|----------------|---------|---------|-----------|
| trips | 139 956 | 13 156 | 126 800 |
| order | 169 | 169 | 169 |
| size | 22 750 | 7 617 | 21 947 |
| DENSITY | 0.80 | 0.27 | 0.77 |

Every docking station in the BSS (169) act as origin or destination of trajectories in both networks. Consequently, their density must be lower than that of the total network. As we can observe, the variation in density of the leisure network to the total network is significantly higher (-66.25%) than the one corresponding to the transport network (-3.75%). This fact demonstrates the distinct underlying structure of both networks.

Let us now focus on the underlying network induced by transport trips. Figure 6 shows the sequential formation of this network considering an increasing percentage of trips. Observing Figure 6d we can clearly state that BiciMAD's underlying public transport network provides almost full coverage (most blank spaces correspond to parks). In addition, this analysis provides meaningful information about the importance of every route in the network. This information could build a solid basis for support decision tools that would help municipalities in the design of new infrastructures or changes in the urban regulation to promote the use of BSSs.

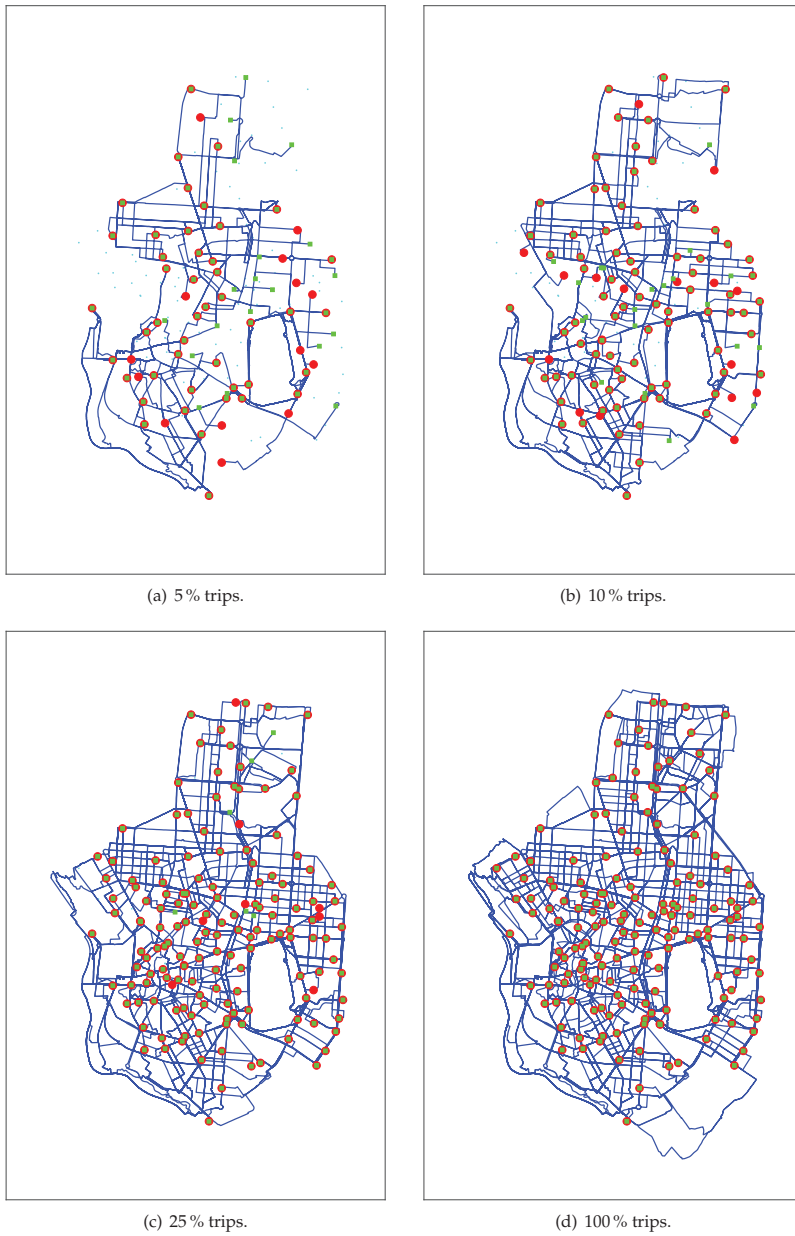


Figure 6. Formation of BiciMAD public transport network.

Finally, we can conclude that 90.60% of BiciMAD's trips were performed as a public transport mode, which highlights the significant contribution of BSSs to public transport in a city.

5. Conclusions and Future Research

This work proposes a methodology to classify trips depending on their intrinsic purpose: transport or leisure. This classification is based on the trip index, which measures the actual travel distance versus the available shortest path. In a BSS, this is not a straightforward measure given that cyclists may use shortcuts that are not contemplated in the underlying network (streets and bicycle lanes) and the regulatory restrictions. In addition, these shortcuts have a time-to-live. Consequently, we constructed a mathematical framework to estimate the heterodox trajectory and take it as the reference for the trip index.

We validated the classification methodology empirically, using data from BiciMAD and analyzing the results from both statistical and operational perspectives.

The analysis of trajectories in BSSs opens a set of future research lines. First, in our study we have found trip indexes greater than 1. This implies that some users achieved trajectories that showed a deviation from the linear path below the Fréchet distance of the orthodox trajectory. This means that under certain circumstances, we can find shortcuts to reduce the traveled length in a BSS. The detailed study of the spatio-temporal existence of these short cuts will be addressed in a specific research.

This analysis will use the Fréchet distance to show the deviation of each actual trip to the linear path between origin and destination. The resulting set of deviations will form a probability distribution, which will eventually model the overall *directivity* of trajectories in a particular BSS. This new concept embeds information regarding both the user and the underlying topology. Thus, using the proposed methodology to restrict the analysis to transport trips will lead us to a global characterization of BSS networks. The resulting methodology will then be applied to datasets of BSS trajectories in different cities to compare their relative performance as a public transport mode.

In addition, trajectories will also be used to predict future occupancy levels in docking stations. This research will be based on a metaheuristic approach based on swarm intelligence, which will characterize how groups of bicycle users flow from one docking station to another.

On the other hand, identifying users' behavior plays a key role in understanding their needs to provide them with optimized services. Companies invest huge amounts of money a year for this type of findings. Public institutions such as municipalities find this knowledge even more important, given that they must select where to invest public money so that they provide the best services to their citizens. Importantly from our perspective, it provides foundation information for planning and design, which will lead to optimization of the deployment of this type of transport mode. Our work provides mathematical and empirical evidence on the type of users a BSS has. This allows municipalities to configure tariffs to promote this type of transport, invest in new bicycle lanes that follow the actual routes users are traveling, etc.

In this respect we have demonstrated that BSSs are a form of public transport not only for the last mile, but in its own right. Consequently, municipalities may start considering them in more depth as a solution for urban transport. Our methodology provides a framework to generate meaningful information to be employed as a decision support tool to the process of reengineering the bicycle infrastructure and the corresponding regulation. This information builds a solid ground of knowledge for BSS managers and municipalities.

Author Contributions: Conceptualization, M.R.W., J.J.V.D., R.F.P. and A.B.R.G.; Formal Analysis, J.M.V. and C.S.Á.; Investigation, J.J.V.D., A.B.R.G.; Writing—Original Draft Preparation, M.R.W. and R.F.P.; Writing—Review & Editing, J.M.V. and C.S.Á.; Funding Acquisition, J.J.V.D. and A.B.R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was developed within the project Co-Mov, co-funded by Comunidad de Madrid (Spain), the European Social Fund, and the European Regional Development Fund, with grant number [Y2018/EMT-4818].

Acknowledgments: The authors would like to thank the contributions and support from: Empresa Municipal de Transportes de Madrid (EMT) and Sociedad Ibérica de Construcciones Eléctricas (SICE).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chiariotti, F.; Pielli, C.; Zanella, A.; Zorzi, M. A Dynamic Approach to Rebalancing Bike-Sharing Systems. *Sensors* **2018**, *18*, 512. [[CrossRef](#)] [[PubMed](#)]
2. Yang, X.H.; Cheng, Z.; Chen, G.; Wang, L.; Ruan, Z.Y.; Zheng, Y.J. The impact of a public bicycle-sharing system on urban public transport networks. *Transp. Res. Part Policy Pract.* **2018**, *107*, 246–256. [[CrossRef](#)]
3. Fishman, E.; Washington, S.; Haworth, N. Bike share's impact on car use: Evidence from the United States, Great Britain, and Australia. *Transp. Res. Part Transp. Environ.* **2014**, *31*, 13–20. [[CrossRef](#)]
4. Rydin, Y.; Bleahu, A.; Davies, M.; Dávila, J.D.; Friel, S.; De Grandis, G.; Groce, N.; Hallal, P.C.; Hamilton, I.; Howden-Chapman, P.; et al. Shaping cities for health: Complexity and the planning of urban environments in the 21st century. *The Lancet* **2012**, *379*, 2079–2108. [[CrossRef](#)]
5. O'Brien, O.; Cheshire, J.; Batty, M. Mining bicycle sharing data for generating insights into sustainable transport systems. *J. Transp. Geogr.* **2014**, *34*, 262–273. [[CrossRef](#)]
6. Song, C.; Koren, T.; Wang, P.; Barabási, A.L. Modelling the scaling properties of human mobility. *Nature Phys.* **2010**, *6*, 818–823. [[CrossRef](#)]
7. González, M.C.; Hidalgo, C.; Barabási, A.L. Understanding individual human mobility patterns. *Nature* **2008**, *453*, 779–782. [[CrossRef](#)]
8. Koppelman, F.S.; Wen, C.H. Alternative nested logit models: structure, properties and estimation. *Transp. Res. Part Methodol.* **1998**, *32*, 289–298. [[CrossRef](#)]
9. Chen, J.; Li, S. Mode Choice Model for Public Transport with Categorized Latent Variables. *Math. Probl. Eng.* **2017**, *2017*, 1–11. [[CrossRef](#)]
10. González, F.; Melo-Riquelme, C.; de Grange, L. A combined destination and route choice model for a bicycle sharing system. *Transportation* **2016**, *43*, 407–423. [[CrossRef](#)]
11. Li, Y.; Zheng, Y. Citywide Bike Usage Prediction in a Bike-Sharing System. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1079–1091. [[CrossRef](#)]
12. Sarkar, A.; Lathia, N.; Mascolo, C. Comparing cities' cycling patterns using online shared bicycle maps. *Transportation* **2015**, *42*, 541–559. [[CrossRef](#)]
13. Shen, S.; Wei, Z.Q.; Sun, L.J.; Su, Y.Q.; Wang, R.C.; Jiang, H.M. The Shared Bicycle and Its Network—Internet of Shared Bicycle (IoSB): A Review and Survey. *Sensors* **2018**, *18*, 2581. [[CrossRef](#)] [[PubMed](#)]
14. Olmos, L.E.; Tadeo, M.S.; Vlachogiannis, D.; Alhasoun, F.; Espinet Alegre, X.; Ochoa, C.; Targa, F.; González, M.C. A data science framework for planning the growth of bicycle infrastructures. *Transp. Res. Part C Emerg. Technol.* **2020**, *115*, 102640. [[CrossRef](#)]
15. Calabrese, F.; Diao, M.; Di Lorenzo, G.; Ferreira, J.; Ratti, C. Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. *Transp. Res. Part C Emerg. Technol.* **2013**, *26*, 301–313. [[CrossRef](#)]
16. Curzel, J.; Lüders, R.; Fonseca, K.; Rosa, M. Temporal Performance Analysis of Bus Transportation Using Link Streams. *Math. Probl. Eng.* **2019**, *2019*, 1–18. [[CrossRef](#)]
17. Aleta, A.; Meloni, S.; Moreno, Y. A Multilayer perspective for the analysis of urban transportation systems. *Sci. Rep.* **2017**, *7*, 44359. [[CrossRef](#)]
18. Domencich, T.; McFadden, D. Statistical estimation of choice probability function. In *Urban Travel Demand—A Behavioral Analysis*; North-Holland Publishing Company Limited: Oxford, UK, 1975; pp. 101–125.
19. De Dios Ortúzar, J.; Willumsen, L. *Modelling Transport*; John Wiley & Sons, Ltd.: NY, USA, 2011.
20. Bovet, P.; Benhamou, S. Spatial analysis of animals' movements using a correlated random walk model. *J. Theor. Biol.* **1988**, *131*, 419–433. [[CrossRef](#)]
21. Mueller, J.E. An Introduction to the Hydraulic and Topographic Sinuosity Indexes. *Ann. Assoc. Am. Geogr.* **1968**, *58*, 371–385. [[CrossRef](#)]
22. Pearl, J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*; Addison-Wesley Longman Publishing Co., Inc.: Reading, MA, USA, 1984.
23. Quddus, M.; Washington, S. Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transp. Res. Part Emerg. Technol.* **2015**, *55*, 328–339. [[CrossRef](#)]
24. Zhang, J.; Pan, X.; Li, M.; Yu, P.S. Bicycle-Sharing System Analysis and Trip Prediction. In Proceedings of the 2016 17th IEEE International Conference on Mobile Data Management (MDM), Porto, Portugal, 13–16 June 2016; Vol. 1, pp. 174–179.

25. Jiang, Z.; Evans, M.; Oliver, D.; Shekhar, S. Identifying K Primary Corridors from urban bicycle GPS trajectories on a road network. *Inf. Syst.* **2016**, *57*, 142–159. [[CrossRef](#)]
26. Guo, N.; Ma, M.; Xiong, W.; Chen, L.; Jing, N. An Efficient Query Algorithm for Trajectory Similarity Based on Fréchet Distance Threshold. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 326. [[CrossRef](#)]
27. Fréchet, M. Sur quelques points du calcul fonctionnel. *Rend. Circ. Matem. Palermo.* **1906**, *22*, 1–72. [[CrossRef](#)]
28. Alt, H.; Godau, M. Measuring the Resemblance of Polygonal Curves. In Proceedings of the Eighth Annual Symposium on Computational Geometry (SoCG '92), Berlin, Germany, 10–12 June 1992; p. 102–109.
29. Eiter, T.; Mannila, H. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Technische Universität Wien, Wien, Austria, 1994.
30. Scherrer, L.; Tomko, M.; Ranacher, P.; Weibel, R. Travelers or locals? Identifying meaningful sub-populations from human movement data in the absence of ground truth. *EPJ Data Sci.* **2018**, *7*, 19. [[CrossRef](#)]
31. Faghih-Imani, A.; Eluru, N. Analysing bicycle-sharing system user destination choice preferences: Chicago's Divvy system. *J. Transp. Geogr.* **2015**, *44*, 53–64. [[CrossRef](#)]
32. Broach, J.; Dill, J.; Gliebe, J. Where do cyclists ride? A route choice model developed with revealed preference GPS data. *Transp. Res. Part A Policy Pract.* **2012**, *46*, 1730–1740. [[CrossRef](#)]
33. Gallotti, R.; Bazzani, A.; Rambaldi, S. Understanding the variability of daily travel-time expenditures using GPS trajectory data. *EPJ Data Sci.* **2015**, *4*, 18. [[CrossRef](#)]
34. Vinagre Díaz, J.J.; Fernández Pozo, R.; Rodríguez González, A.B.; Wilby, M.R.; Sánchez Ávila, C. Hierarchical Agglomerative Clustering of Bicycle Sharing Stations Based on Ultra-Light Edge Computing. *Sensors* **2020**, *20*, 3550. [[CrossRef](#)]
35. Domènech, A.; Gutiérrez, A. A GIS-Based Evaluation of the Effectiveness and Spatial Coverage of Public Transport Networks in Tourist Destinations. *Int. J. Geo-Inf.* **2017**, *6*, 83. [[CrossRef](#)]
36. Hamilton, T.L.; Wichman, C.J. Bicycle infrastructure and traffic congestion: Evidence from DC's Capital Bikeshare. *J. Environ. Econ. Manag.* **2018**, *87*, 72–93.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Impact on Road Safety and Operation of Rerouting Traffic in Rural Travel Time Information System

Mariusz Kiec ¹, Carmelo D'Agostino ^{2,*} and Sylwia Pazdan ¹

¹ Faculty of Civil Engineering, Cracow University of Technology, Warszawska 24, 31-155 Cracow, Poland; mkiec@pk.edu.pl (M.K.); sylwia.pazdan@pk.edu.pl (S.P.)

² Transport & Roads Division, Faculty of Engineering, Lund University, John Ericssons väg 1, Box 118, 221 00 Lund, Sweden

* Correspondence: carmelo.dagostino@tft.lth.se; Tel.: +46-(0)730-74-4126

Received: 26 June 2020; Accepted: 23 July 2020; Published: 25 July 2020

Abstract: The Travel Time Information System (TTIS) is an Intelligent Traffic Control System installed in Poland. As is common, travel time is the only factor in the decision about rerouting traffic, while a route recommendation may consider multiple criteria, including road safety. The aim of the paper is to analyze the safety level of the entire road network when traffic is rerouted on paths with different road categories, intersection types, road environments, and densities of access points. Furthermore, a comparison between traffic operation and road safety performance was carried out, considering travel time and delay, and we predicted the number of crashes for each possible route. The results of the present study allow for maximizing safety or traffic operation characteristics, providing an effective tool in the management of the rural road system. The paper provides a methodology that can be transferred to other TTISs for real-time management of the road network.

Keywords: ITS; road safety; travel time information system; safety performance function

1. Introduction

The Travel Time Information System (TTIS) was implemented in Poland in 2012 with the aim to solve the problem of seasonal congestion of the road network in the recreation area of the Malopolska region. This complex Intelligent Transport System (ITS) covers both national and regional rural and suburban roads. The aim of the TTIS is the effective exploitation of capacity reserves existing in the road network by providing information to road users about alternative routes with a lower traffic density and shorter travel time.

The traffic redistribution on the road network not only impacts the environment (fuel consumption and an increase in pollution) but it can affect in a not negligible way the safety of the road users. The common problem in the implementation of this kind of complex system is that the travel time (measured as traffic volume in relation to the infrastructure capacity) is the only factor in the decision about rerouting traffic instead of considering multiple criteria [1]. The present research aims to analyze the effects of the TTIS system on road safety and travel time by verifying the overall safety and traffic operation performance of the road network, including national and regional roads and their intersections where traffic is rerouted. The road safety assessment was carried out by calibrating Safety Performance Functions (SPFs) for national and regional roads (with a greater density of access points), for both sections and intersections. The use of SPFs calibrated on the basis of empirical data offered the advantage to predict the safety conditions of the whole road network included in the TTIS for different values of rerouted traffic. Furthermore, by assessing the safety level of each road section and intersection as a function of the Annual Average Daily Traffic (AADT) and geometric parameters, it was possible to simulate in real time the network performance in terms of road safety. This was possible since the effects of the TTIS system did not change the risk related to the different road categories and

intersection types (the estimated crash modification factor for the TTIS was close to 1, as shown in [2]). In fact, by rerouting traffic, the TTIS affects crash frequency in non-homogenous road categories and intersection designs. A comparison of the traffic operation and road safety was also assessed, looking for the optimum of two measures of system operation, changes in crashes and travel time.

This paper is a further step to develop a preliminary analysis on the safety performance of the network when traffic is rerouted between paths with different road segment categories and characteristics [2]. The mentioned study has high reliability in the methodology and data but was conducted only in terms of road safety for road sections, while intersections were not included in the analysis.

2. Literature Review

The TTIS, recommending alternative routes on the basis of travel time, is commonly used. It improves the level of service and may have controversial effects on road safety on the main regional and national road networks based on the magnitude of the rerouted traffic [2]. This preliminary study [2] considers the effects of the system by developing simulated scenarios of traffic redistribution on the network. The main problem of those applications is that the TTIS may also influence the change in traffic at intersections. It may lead to a greater number of crashes than those related only to road sections, based on the different risks related to the specific crossed intersections [3].

The added value of including intersections in the overall analysis of the real-time safety conditions of the network is due to the fact that drivers may more frequently change their travel routes, preferring routes with low priority and a shorter travel time. This may result in a greater number of dangerous maneuvers at intersections and consequently a greater probability of multiple vehicle road crashes.

The TTIS is more often implemented in urban areas than on rural roads. For example, such a system operates in the Norwegian city of Trondheim [3] and Hong Kong [4]. A similar experience was carried out in London, and showed an increase in the number of crashes in connection with the increase in the proportion of vehicles equipped with connected on-board tools for rerouting [4]. Assuming a total share of vehicles equipped with on-board tools to be 100%, the costs of road crashes will increase by 1.5% [5]. Other studies suggest that the distribution of traffic in the suburban road network, relying on the shortest travel time while still maintaining an acceptable level of service, led to an increase in the risk of crashes (considering a non-linear relationship between crashes and traffic volume) [6,7]. In all of those systems, the basic principle is to reroute traffic in the road network to minimize the delays (travel time) of users. Based on the real-time traffic volume, the system calculates the traffic performance and gives information about alternative routes to users, often without considering the impact of rerouting on future traffic conditions [8]. Research [9] found that driver route decisions depend not only on travel time information, but also on route scenery, the number of intersections, and traffic signals along the alternative route. The mentioned approach also lacks knowledge about the safety conditions of the network, which require a specific analysis based on the risk level related to road characteristics and intersection types.

The analysis of the effects of route recommendations on accident risk in urban networks [8] indicated that accident reductions, resulting from a more efficient distribution of traffic in congested networks, are small. The use of minor roads can reduce travel time, but at the same time can increase the accident frequency.

The variation of network-wide accidents caused by traffic redistribution, subject to various levels of dynamic route guidance, market penetration, and the potential of a new safety-enhanced route guidance system based on different levels and pattern simulation [10], showed approximately a 10% increase in accidents.

The available worldwide experience and research suggest the need to take into account not only the traffic performance but also an assessment of the safety performance on the alternative routes in the road networks covered by TTIS [11–13]. Furthermore, a reliable balance or comparison between traffic operation and road safety was never carried out for the existing TTIS for rural two-lane roads.

The aim of the paper is an assessment of the road network covered by the TTIS in terms of road safety and traffic operation, considering a dynamic traffic distribution. The paper proposes the next stage of the study of road safety on roads included in the TTIS [2] with consideration for road safety and the impact of travel time at intersections as part of the network. Therefore, for road networks included in the TTIS: 1) a detailed systematic road safety and traffic operation analysis has been done, considering the predicted number of crashes on the basis of ad hoc calibrated SPFs, 2) traffic data and their monthly variability were considered by getting data from the measuring devices implemented in the system, and 3) the relation between traffic volume and speed was developed by the authors based on empirical data from the TTIS.

3. Travel Time Information System for Rural Roads and Data

The aim of the Travel Time Information System (TTIS) implementation was to improve traffic performance (reduction of travel time), by rerouting traffic in the road network covered by the system between two tourist sites, Zakopane (Z) and Rabka (R), in a recreational region in Poland (Figure 1). In Figure 1, the possible routes in the TTIS and the abbreviations of town names are presented.

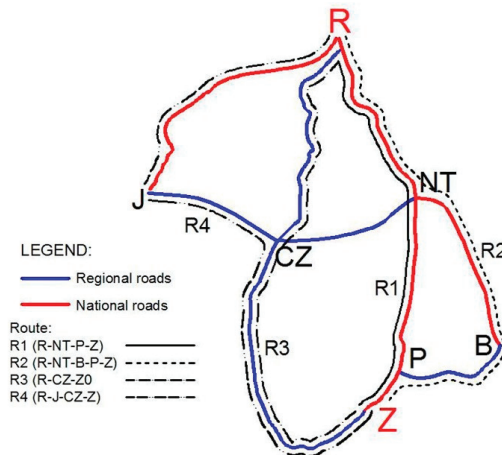


Figure 1. National and regional road network included in the analysis and alternative route identification.

3.1. Structure of the TTIS

To collect traffic data (traffic volume as well as travel and spot speed), the TTIS consists of a series of devices and sensors, i.e.,:

- Sixty Remote Traffic Microwave Sensors (RMTSs), which register traffic data (traffic volume, vehicle speed, types of vehicles) located on each route at more or less constant distances;
- Forty-four HD cameras to provide real-time control of the traffic situation, located on all routes;
- Eight Automatic Number Plate Recognition (ANPR) cameras and 16 Variable Message Signs (VMSs) to provide information to users about travel time. They are located at each intersection where a change of route is possible; and
- Ten Weather Stations (WSs) to collect weather data and provide warnings for drivers. WSs are located at the ANPR and VMS locations.

The data collected by the sensors are used to estimate travel times from one location to another for different sections of roads (RMTSs) and routes (ANPR). The driver can select a route based on data on travel times provided via a VMS (Figure 2), an internet website, or a mobile app. As a result, the traffic

volume distribution may vary depending on the system recommendations and the drivers' decisions to change route between the cities of Zakopane (Z) and Rabka (R) (Figure 1).



Figure 2. Variable message signs for travel time of the Travel Time Information System (TTIS).

3.2. Data Sample

The analyzed road network consists of national and regional two-lane rural and suburban roads, and various types of intersections (roundabouts, signalized and non-signalized intersections).

The TTIS reroutes traffic from the main route R1 (national roads R-NT-P-Z) to alternative routes (regional and national roads between Z and R, including changing points J, CZ, P, and NT) (Figure 1) characterized by various geometric standards. The road sample covered by the TTIS is made up of 156.4 km of road (including 81.5 km of regional road and 74.9 km of national road) [2].

Furthermore, the Safety Performance Functions (SPFs) for road sections and intersections were calibrated on a larger sample composed of data from two-lane roads. Those roads are located in the same region close to the routes covered by the TTIS and with similar geometric and traffic characteristics, but not affected by the system. This did not introduce any bias because there is a negligible effect of the TTIS on safety in the primary road network. This emerged from the results of the estimation of a Crash Modification Factor (CMF) which assumed a value not far from 1 [2]. That additional sample is made up of 322.9 km of road, including 184.3 km of regional road and 138.6 km of national road. This approach allows us to predict the average crash frequency for different road categories (regional and national) and intersection types (roundabouts, signalized and non-signalized intersections) comprising the network influenced by the TTIS. Tables 1 and 2 report the summary statistics of the variables describing the sample used for the calibration of the SPFs. Roads in the system were divided into homogenous segments in terms of traffic volume (AADT), area (rural, suburban), and horizontal alignment. Intersections were categorized based on traffic control organization. Single-lane roundabouts and signalized and non-signalized four-leg intersections were distinguished. For each homogeneous segment [14,15], the segment length and the Curvature Change Rate (CCR) (Table 1) were defined as geometric covariates. For each segment and intersection, the AADT (for intersections, both the major and minor road AADT were considered) and the number of crashes, fatalities, and injuries were collected to provide the final dataset used in the SPF calibration (Table 1, while Table 2 reports the same for intersections). AADT and crash data were recorded from 2009 to 2014.

In order to assess the TTIS in terms of travel time and safety performance, the different alternative routes, consisting of sections (Table 3), were distinguished, i.e., R1 (main, the most selected route), R2, R3, and R4 (Figure 1 and Table 4). Each route is a combination of national or/and regional road sections and may be selected by the drivers between Rabka (R) and Zakopane (Z). Routes are made up of sections defined between the main intersections and can include the same segments, e.g., Route R2 (R-NT-B-P-Z) contains sections R-NT and P-Z, which are part of route R1 as well. The alternative route (for the main route R1) consisting of only regional roads is route R3.

Table 1. Summary statistics of the dataset of road segments (Annual Average Daily Traffic—AADT—is the minimum and maximum in the whole period of analysis/CCR—Curvature Change Rate).

| | | No. of Road Segments | Length [km] | | AADT [veh/day] | | CCR [deg/km] | | Number of | | | |
|----------------|-------------------|----------------------|-------------|------|----------------|------|--------------|-----|-----------|------------|---------|-----|
| | | | Min | Max | Min | Max | Min | Max | Crashes | Fatalities | Injured | |
| National roads | TTIS | suburban | 37 | 0.15 | 2.14 | 4892 | 17,564 | 0.0 | 325.7 | 228 | 19 | 318 |
| | | rural | 22 | 0.19 | 2.22 | 6226 | 17,023 | 0.0 | 99.6 | 112 | 12 | 177 |
| | additional sample | suburban | 110 | 0.15 | 2.14 | 4892 | 22,022 | 0.0 | 708.5 | 320 | 41 | 442 |
| | | rural | 80 | 0.19 | 2.42 | 4892 | 18,283 | 0.0 | 287.9 | 370 | 41 | 563 |
| Regional roads | TTIS | suburban | 54 | 0.17 | 2.38 | 2040 | 18,850 | 0.0 | 1095.9 | 145 | 7 | 190 |
| | | rural | 19 | 0.12 | 2.16 | 2040 | 6732 | 0.0 | 449.3 | 62 | 9 | 104 |
| | additional sample | suburban | 152 | 0.17 | 2.38 | 2268 | 10,101 | 0.0 | 1044.9 | 249 | 22 | 312 |
| | | rural | 76 | 0.13 | 2.43 | 2268 | 9312 | 0.0 | 796.3 | 151 | 13 | 249 |

Table 2. Summary statistics of the intersection dataset.

| | | Number of Intersections | Crashes | Fatalities | Injured |
|----------------|-------------------|-------------------------|---------|------------|---------|
| | | | | | |
| | additional sample | 18 | 30 | 11 | 34 |
| signalized | TTIS | 3 | 3 | 2 | 4 |
| | additional sample | 7 | 21 | 1 | 23 |
| non-signalized | TTIS | 19 | 41 | 9 | 56 |
| | additional sample | 43 | 59 | 34 | 37 |

Table 3. Description of the analyzed road network covered by the TTIS by section.

| (a) Sections | R-NT | NT-P | P-Z | NT-B | B-P | R-CZ | CZ-Z | R-J | J-CZ |
|-------------------------------------|---------|---------|--------|---------|---------|----------|----------|-----------|---------|
| length [km] | 18.5 | 16 | 4.9 | 16.9 | 11.5 | 21.8 | 29.2 | 25.2 | 12.4 |
| min AADT [veh/day] | 14,218 | 14,223 | 17,564 | 9012 | 2040 | 3810 | 3810 | 7255 | 4281 |
| max AADT [veh/day] | 17,023 | 15,823 | 17,564 | 15,106 | 5891 | 3874 | 8954 | 7255 | 4751 |
| number of segments (Suburban/Rural) | 12 | 10 | 4 | 13 | 11 | 17 | 27 | 26 | 7 |
| road network (National/Regional) | (7S/5R) | (5S/5R) | (4S) | (9S/4R) | (8S/3R) | (11S/6R) | (20S/7R) | (13S/13R) | (3S/4R) |
| | N | N | N | N | R | R | R | N | R |

Table 4. Description of the analyzed road network covered by the TTIS by route.

| (b) Routes | R1 (R-NT-P-Z) | R2 (R-NT-B-P-Z) | R3 (R-CZ-Z) | R4 (R-J-CZ-Z) |
|-------------------------------------|---------------|-----------------|-------------|---------------|
| length [km] | 39.4 | 51.8 | 51.0 | 66.8 |
| min AADT [veh./day] | 14,218 | 5428 | 3810 | 3810 |
| max AADT [veh./day] | 17,564 | 17,564 | 8954 | 8954 |
| number of sections (Suburban/Rural) | 26 | 40 | 44 | 60 |
| road network (National/Regional) | (16S/10R) | (28S/12R) | (31S/13R) | (36S/24R) |
| | N | N,R | R | N,R |

4. Methodological Approach

The evaluation of the TTIS safety performance and travel time was carried out with the use of the following methodologies:

1. the calibration of SPFs for each road category and location (i.e., national/regional and rural/suburban roads) and for each intersection type (roundabouts, signalized and non-signalized intersections). This study aims to assess road safety in the entire road network included in the TTIS by juxtaposing the total predicted number of crashes for routes in various configurations of traffic distribution within the road network covered by the system; and

2. the assessment of travel time for routes included in the TTIS, based on the observed relationship between traffic volume and speed (for road sections) and delay (for intersections) with reference to traffic volume variability.

Regardless of the model calibration for segments or intersections, crashes observed at a site i in the year t ($Y_{i,t}$) are typical time series data across years and can, therefore, be represented by the following simplified model structure Equation (1):

$$Y_{i,t} = \text{trend} + \text{regression term} + \text{random effects} + \text{local factors}, \quad (1)$$

where “trend” refers to a long-term movement due to a change in the risk factors with time, the “regression term” is of the same form as the Safety Performance Functions (SPFs), “random effects” account for latent variables across the sites, and the “local factors” refer to the dispersion between the normal safety level for similar locations and the safety level for the specific site. Random effects and local factors both contribute to the dispersion of crash counts as compared to the mean value estimated by the regression term.

The use of the Negative Binomial (NB) distribution to represent the distribution of crash counts is commonly accepted [16]. Therefore, when excluding trend effects (i.e., the phenomenon is stationary), Generalized Linear Models (GLMs) are especially useful in the context of traffic safety, for which the distribution of accident counts in a population often follows the negative binomial distribution [17,18]. In the present research work, the analysis was performed without considering possible variation in the predicted number of crashes due to the time trends because of the limited period of analysis and the target of the research work.

Considering all this, and consistent with the state of the art in developing these models, a generalized linear modeling approach and model form was used in the elaboration, considering a negative binomial error distribution for either SPF calibrated for road sections or intersections. The important property of the GLM is the flexibility in specifying the probability distribution for the random component [19–21]. The model parameter estimation was performed following the maximum likelihood calibration methodology. The dispersion parameter obtained by the model calibration indicates how far the model is from a Poisson distribution, which is typically lower when a longer period is considered (lower data dispersion). Therefore, the value of the intercept is the average value in the whole period of 6 years [22–24].

4.1. Safety Performance Function Calibration for Road Segments

To compare the safety performance in terms of predicted crashes due to the changes in the Annual Average Daily Traffic (AADT) (which is the only parameter which varies due to the TTIS), ad hoc SPFs were calibrated using, as independent AADT variables, the horizontal alignment (the value of the Curvature Change Rate—CCR) and the section length on different categories of roads (national/regional) and in different locations (rural/suburban). The inclusion of other covariates, such as the segment length (L) and the horizontal alignment, helps in isolating the contribution of AADT. The inclusion of exponents for both L and AADT improves the adaptability of the model to different conditions for other variables not included in the model [25].

As a result of the previous consideration in developing SPF models, Equation (2) shows the selected model form:

$$E(Y) = \exp(\alpha) * \text{AADT}^\beta * L^\gamma * \exp(\delta * \text{CCR}), \quad (2)$$

where: $E(Y)$ is the yearly predicted number of crashes; L is the segment length [m]; AADT is the annual average daily traffic [veh./day]; and α , β , γ , and δ are regression terms.

For the regional suburban area, the variable CCR was not statistically significant and, therefore, was removed from the model.

The results of the regression analysis, obtained by using a maximum likelihood calibration methodology, are reported in Table 5. Those SPFs returned the predicted average number of crashes per year for every road section of the network based on road category and location.

Table 5. Regression coefficient, standard error, and p-value of the Safety Performance Functions (SPFs) for road segments.

| National Rural | | | | | | |
|----------------------|----------|----------|----------------|----------------------------|----------|------------|
| Parameter | | Estimate | Standard Error | Wald 95% Confidence Limits | | Pr > ChiSq |
| Intercept | α | -22.4297 | 2.6687 | -26.56 | -16.1 | <0.0001 |
| AADT | β | 1.564 | 0.2594 | 1.0556 | 2.0725 | <0.0001 |
| L | γ | 1.0802 | 0.1513 | 0.7836 | 1.3769 | <0.0001 |
| CCR | δ | 0.0029 | 0.0015 | -0.0001 | 0.0058 | 0.0495 |
| Dispersion parameter | | 0.5404 | 0.139 | 0.2679 | 0.8128 | - |
| National suburban | | | | | | |
| Intercept | α | -10.3533 | 2.1904 | -13.5478 | -4.9615 | <0.0001 |
| AADT | β | 0.4942 | 0.20089 | 0.0847 | 0.9037 | 0.018 |
| L | γ | 0.7954 | 0.103 | 0.5936 | 0.9973 | <0.0001 |
| CCR | δ | 0.0024 | 0.0007 | 0.001 | 0.0038 | 0.0006 |
| Dispersion parameter | | 0.4894 | 0.1341 | 0.2265 | 0.7523 | - |
| Regional rural | | | | | | |
| Intercept | α | -15.6614 | 3.3756 | -21.1789 | -7.9467 | <0.0001 |
| AADT | β | 0.9918 | 0.3427 | 0.3201 | 1.6635 | 0.0038 |
| L | γ | 0.907 | 0.1606 | 0.5921 | 1.2218 | <0.0001 |
| CCR | δ | -0.0017 | 0.0009 | -0.0034 | 0.0000 | 0.0411 |
| Dispersion parameter | | 0.4855 | 0.1989 | 0.0958 | 0.8753 | - |
| Regional suburban | | | | | | |
| Intercept | α | -15.4732 | 2.0491 | -18.3907 | -10.3585 | <0.0001 |
| AADT | β | 0.9772 | 0.224 | 0.5383 | 1.4162 | <0.0001 |
| L | γ | 0.9009 | 0.114 | 0.6775 | 1.1243 | <0.0001 |
| CCR | δ | - | - | - | - | - |
| Dispersion parameter | | 0.4268 | 0.1357 | 0.1608 | 0.6928 | - |

The best safety performances were observed on sections of national roads in rural areas (because of better geometrical standards), and the worst were in suburban areas (because of the high observed speed). Regional roads have similar safety performances in rural and suburban areas (Figure 3).

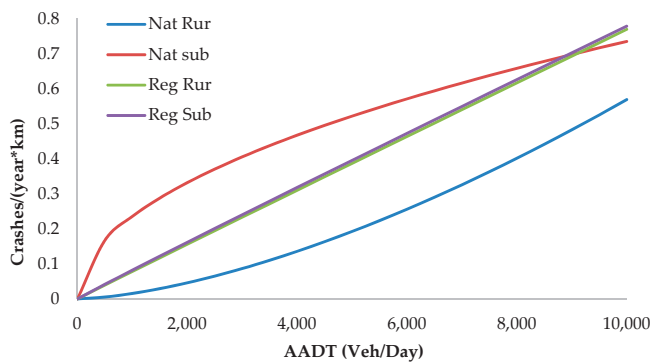


Figure 3. SPF diagram for segments in different road locations and road categories, with CCR equal to zero (tangent).

4.2. Safety Performance Function Calibration for Intersections

To estimate the predicted crash frequency for intersections, a unique SPF was calibrated using, as a categorical variable, the different intersection types, i.e., NS: non-signalized, R: roundabout, S: signalized, with a similar approach to [26]. This difference in the approach to the regression analysis between road sections and intersections was mainly due to the small sample size for each single intersection type. Only AADT was statistically significant, with a p-value lower than 0.05, and therefore it was used in the models for the major and minor roads. The model form is shown in Equation (3) and the results of calibration are shown in Table 6 and Figure 4.

$$E(Y) = \exp(\alpha) * AADT_{ma}^\beta * AADT_{mi}^\gamma * \exp(\delta_i * Cat), \tag{3}$$

where: E(Y) is the yearly predicted number of crashes; AADT_{ma} is the average annual daily traffic for major roads [veh./day]; AADT_{mi} is the average annual daily traffic for minor roads [veh./day]; Ca is the categorical variable related to the type of intersection (NS: non-signalized, R: roundabout, S: signalized); α, β, and γ are regression terms of the continuous variables; and δ_i is the regression term of the categorical variables.

Table 6. Regression coefficient, standard error, and p-value of the intersection SPFs.

| Parameter | | Estimate | Standard Error | Chi-Square | Pr > ChiSq |
|---------------------|----------------|----------|----------------|------------|------------|
| Intercept | α | -11.0055 | 2.694 | 16.69 | <0.0001 |
| AADT _{ma} | β | 0.8682 | 0.2598 | 11.17 | 0.0008 |
| AADT _{mi} | γ | 0.4813 | 0.1702 | 7.99 | 0.0089 |
| Non-signalized (NS) | δ _i | 0.2605 | 0.4077 | 0.41 | |
| Roundabout (R) | δ _i | -0.2313 | 0.4411 | 0.27 | |
| Signalized (S) | δ _i | 0 | 0 | . | |
| Dispersion | | 0.6943 | 0.2181 | | |

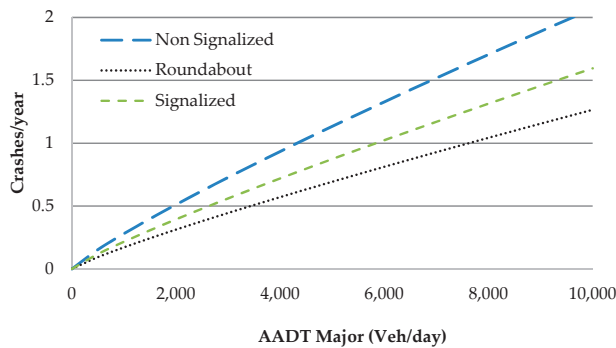


Figure 4. SPF diagram for different intersection types.

The safest intersections are the roundabouts followed by signalized intersections, while the worst performance is from non-signalized intersections, as expected (Figure 4). Therefore, the predicted crash likelihood of users traveling on alternative routes will be dependent on the type of road, the length of travel, and the number and types of intersections on the selected route.

4.3. Assessment of Travel Time and Variability of Traffic Volume

In order to evaluate the impact of traffic distribution on road safety, it is important to assess travel time for each route based on individual road segments and over entire networks, similar to [27]. Travel time has an impact on route selection by drivers, and as result, it affects the traffic distribution.

The relationships between traffic volume and speed (for each road category and road location) were estimated by the authors.

Based on empirical data, from the TTIS for each section, Figure 5 presents the relationships between speed and directional traffic volumes for national and regional roads and rural and suburban areas. In order to evaluate the traffic performance for intersections, delays as a measure of effectiveness were calculated based on the Highway Capacity Manual approach [28]. To calculate delays at the intersections, 10% of the share of peak-hour AADT was assumed. Based on travel time for sections and delays for intersections, travel time for each route was computed and compared with data from the TTIS to validate the approach.

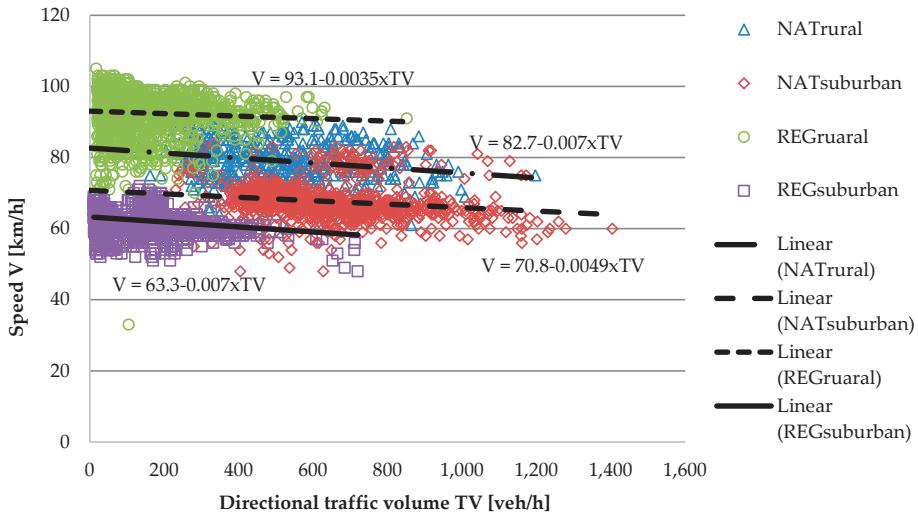


Figure 5. Impact of traffic volume on speed for various sections of the TTIS.

In order to evaluate traffic volume variability in the TTIS, yearly traffic distributions for each route were compared (Figure 6). The rerouting of traffic during peak periods is reported in Figure 6 as it was used in Scenario 3.

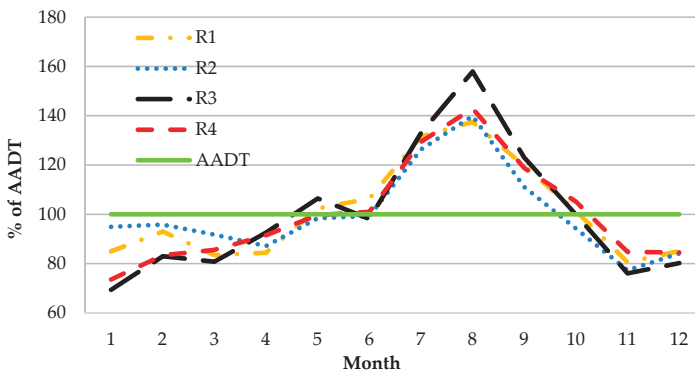


Figure 6. Variability of traffic volume during the year based on data from the TTIS.

The results presented in [1] confirm the need for an overall assessment of the safety performance of the system, not only for road sections, but also for intersections. The assessment of safety and traffic performance of road networks is a complex problem due to possible changes in traffic distribution at the intersections. Therefore, three different scenarios were assumed for the analysis of the impact of traffic distribution on road safety and travel time:

- an increase in AADT on main route R1 to 150% of AADT with a 10% step;
- an increase in traffic in order to balance travel time for two of the most important routes, R1 and R3; and
- an increase in traffic for all routes based on the rate from summertime (peak period).

The operation of the TTIS in terms of road safety and travel time were evaluated based on crash and traffic data.

The first scenario allows us to assess the impact of traffic volume on travel time and road safety for the main route R1, in order to show how the system operates (an increase in traffic volume with a 10% step) and indicate threshold values of traffic volume that should activate the TTIS. The second scenario allows us to assess the impact on road safety when travel time is balanced for the fastest routes, which means the system should start to work. The last scenario shows how the TTIS is working in the peak period (summertime) during traffic rerouting based on travel time.

5. Results and Discussion

In the present research work, three scenarios of rerouting are presented. Scenario 0 simulates the actual conditions based on observed data; an alternative scenario simulates an increase of 150% in traffic volume (Scenario 1) for the best route in terms of road standards, i.e., R1; a second alternative scenario considers the travel time of R1 equal to the route which consists of only regional roads and is the most selected alternative route for R1, i.e., R3; and the third alternative scenario considers the highest traffic volume (peak traffic) registered by the system (in August) for all routes. Those three alternative scenarios were helpful in getting values related to road safety measures or travel time and provide a basis for comparison among the different routes in different conditions.

The results of all analyzed scenarios are included in Table 7. In this table, the ratio of values for crashes and travel time as a sum of both road sections and intersections are also included. These values are calculated in relation to the main route R1 (Equation (4)).

$$\text{ratio} = R_i/R_1, \quad (4)$$

where: R_i is the value of the number of crashes or travel time for the i -th route.

A value of the ratio lower than 1 indicates that the conditions of safety or/and travel time, in comparison to the main route R1, are better.

The results indicate that for a traffic volume equal to the observed AADT (Scenario 0), route R1 has the lowest travel time (at least 41% in comparison to route R2, and even 85% of route R4), but the lowest number of crashes are predicted for route R3 (45% of crashes of R1).

The increase in traffic only for main route R1 to 150% of AADT causes an increase in travel time and the number of crashes, as expected, taking into account the increase in risk exposure. In Figure 7, the impact of the increase in AADT for route R1 on safety and travel time by *ratio* is presented.

An increase in traffic volume equal to 143.56% of travel time is the same for routes R1 and R3 (Scenario 2). In this case, the number of crashes for routes R2, R3, and R4 is lower than for route R1. The safest route is R3, where a reduction of 88% in the predicted number of crashes, compared with main route R1, is observed.

Table 7. Values of crashes and travel time for all routes (the lowest values are in bold).

| | Route | | | | Ratio | | |
|--|--------------|-------|--------------|-------------|-------|-------------|-------|
| | R1 | R2 | R3 | R4 | R2/R1 | R3/R1 | R4/R1 |
| Observed AADT (scenario 0) | | | | | | | |
| Crashes (SPF) [crash/year] | 53.23 | 58.54 | 23.99 | 37.68 | 1.10 | 0.45 | 0.71 |
| Crash rate [Crash*10 ⁶ /(365*AADT*km)] | 0.17 | 0.16 | 0.13 | 0.10 | - | = | - |
| Travel time + delay [min] | 33.01 | 46.41 | 49.07 | 61.02 | 1.41 | 1.49 | 1.85 |
| Increase in traffic volume to 150% of AADT for R1 (scenario 1) | | | | | | | |
| Crashes (SPF) [crash/year] | 80.41 | 76.90 | 24.87 | 38.56 | 0.96 | 0.31 | 0.48 |
| travel time + delay [min] | 52.48 | 66.12 | 49.07 | 61.02 | 1.25 | 0.94 | 1.16 |
| Increase in traffic volume to 143.56% of AADT for R1 (scenario 2) – the same travel time for R1 ad R3 | | | | | | | |
| Crashes (SPF) [crash/year] | 76.75 | 74.43 | 24.76 | 38.45 | 0.97 | 0.32 | 0.50 |
| Travel time + delay [min] | 49.07 | 62.33 | 49.07 | 61.02 | 1.27 | <u>1.00</u> | 1.24 |
| Increase in traffic volume to value of peak period (in August) for all routes (scenario 3) – (R1 = 137.5%. R2 = 140%. R3 = 158%. R4 = 143%) | | | | | | | |
| Crashes (SPF) [crash/year] | 76.29 | 82.94 | 37.08 | 55.30 | 1.09 | 0.49 | 0.72 |
| Travel time + delay [min] | 57.57 | 75.09 | 63.00 | 74.10 | 1.30 | 1.09 | 1.29 |

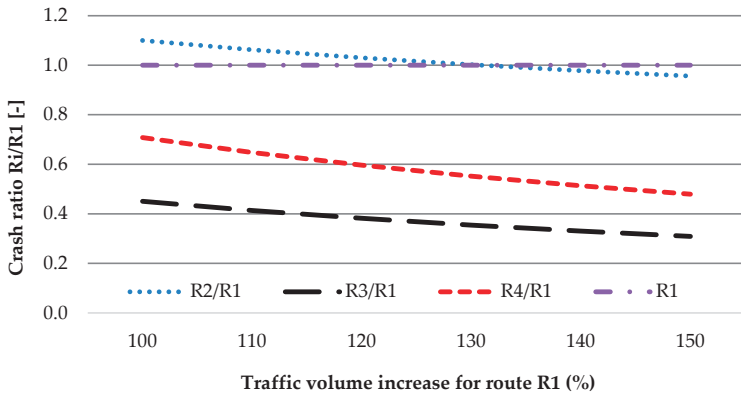
Therefore, the safest route, R3, is very attractive, even in the condition of a high share of rerouting (Scenario 3), which can, in general, cause an increase in crashes (about 50% of crashes compared with the predicted one for route R1) during the peak period (158% of AADT). Despite the benefit to road safety, the benefits to travel time are limited to 6% (Figure 7, Scenario 2) in the case of the same increase in AADT for all routes. For the changing of traffic, as for Scenario 3, the faster route is R1. In other words, this latter condition means that the TTIS is saturated.

Analysis indicates that the best alternative route (for main route R1) in the TTIS is route R3 (Figure 8; Figure 9).

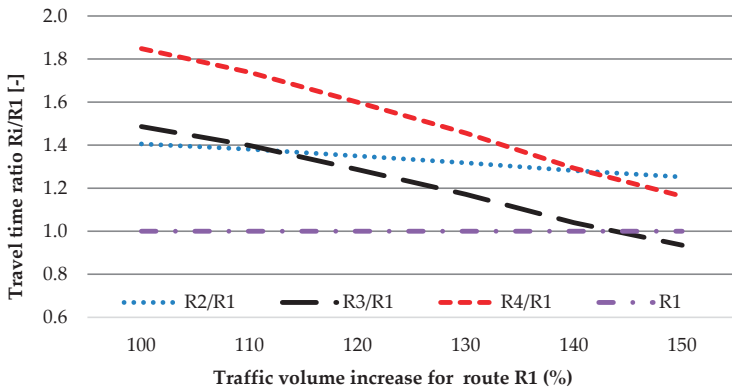
Routes R2 and R4 are not competitive compared with routes R1 and R3, both in terms of delay and safety performance. R2 is not competitive because of the greater value of travel time and predicted crash frequency in comparison to R1. It can be a good alternative route in case of local and temporary traffic interruptions on road sections belonging to other routes due to, e.g., crash occurrences or construction works. Route R4 is too long to be competitive and it is rarely used as an alternative route.

The best alternative for the main route R1 is route R3, mainly due to road safety matters. It results in a lower value of AADT (max AADT for R3 is equal to 8954 veh/day) and greater reserves of capacity. An increase in the number of crashes for R3 is lower compared with the main route R1 and is equal to 80% when considering the same percentage increase in traffic volume. In other words, it is possible to reroute 20% more vehicles to route R3 than to main route R1 to obtain the same road safety level. Therefore, it is possible to reroute more traffic in the system to R3. Travel time in peak traffic is also competitive despite the longer routes.

One of the main problems in the evaluation of the TTIS performance is related to driver choices or, in other words, how drivers use information from VMSs to select routes. Data about the variability of traffic (Figure 6) allows us to compare data on traffic distribution for one year. The peak period in the year is related to the activities of the region, whose function is mainly recreation. Based on those data (for August), the authors assume that differences in the main route R1 are the result of rerouting caused by the TTIS (higher AADT for R3 and R4).



(a)



(b)

Figure 7. Crash (a) and travel time (b) ratio for increase in traffic volume only for route R1.

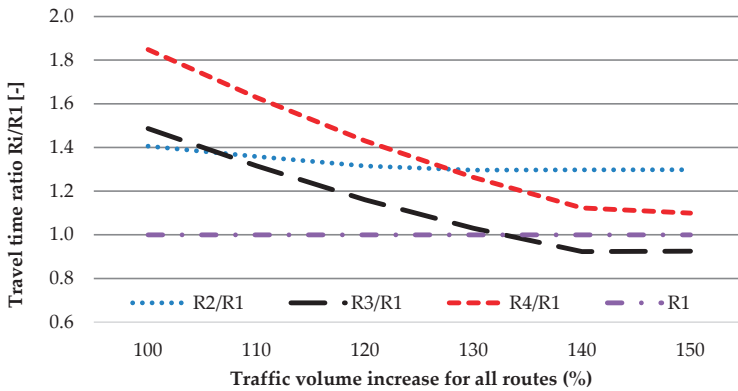


Figure 8. Travel time ratio for increase in traffic volume for all routes.

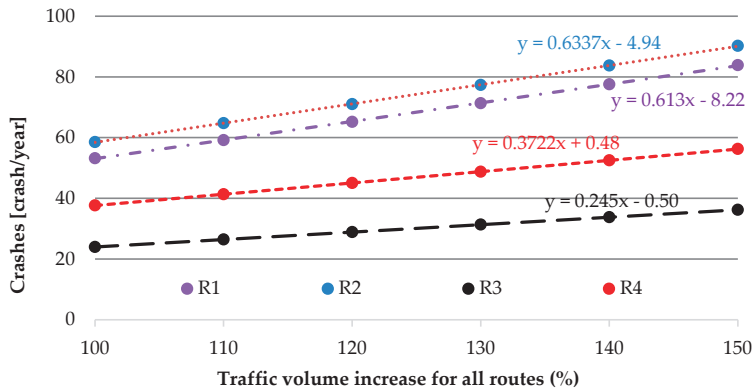


Figure 9. Impact of traffic volume increase for all routes on crashes.

The same data indicate that, during wintertime (January and February), drivers prefer to use national roads (R1 and R2). It can be related to geometrical parameters (lower for regional roads) and the winter maintenance standard of roads (better for national roads).

6. Conclusions

The goal of the research was to develop a methodology for the evaluation of the effects on safety and travel time of the rural Travel Time Information System. For this purpose, a flexible approach was used by calibrating ad hoc SPFs for road sections (national/regional roads and rural/suburban roads), intersections, and by assessing travel time for the same possible traffic scenarios.

The presented methods, through the estimation of the influence of traffic changes in the road network with the TTIS on safety and traffic performance, allowed us to evaluate the threshold values to be used in the the TTIS's control system. This is why, in order to efficiently operate, the TTIS has to be set with threshold values related not only to travel time but also for all the factors which can be directly influenced by the system based on the road network performance, categories, and exposure factors. Looking at the results, the use of the system makes it possible to improve the road safety that is particularly challenged when the network is made up of roads with different standards.

Therefore, given the different risks associated with different road categories, area types, intersection typologies, and the dynamic change in traffic volume (exposure factor) produced by rerouting, it is possible to estimate the TTIS effects on the road safety of the entire road network, by using the SPF models.

The problem of the analyzed system, and common to all ITSs, is that travel time is the only factor in the decision about rerouting traffic. Displayed messages should change from the value of travel time to the recommended way of choosing from multiple criteria, which should also consider road safety.

The ITS, while providing benefits for traffic operation, changes the overall safety performance of the road network. To avoid those effects, the system management should be oriented toward more factors, finding a balance between traffic flow improvement and road safety. The methodology proposed in the paper, with a proper local calibration, can be readily used as a tool for practical applications.

Indirectly, the presented paper indicates the need to develop a navigation system with the selection of routes, taking into account the level of risk in road safety. The presented methodology can be useful to implement this approach in apps to inform drivers about risk. The changes in traffic volume observed by devices in the ITS can be used in the autocalibration procedures of SPFs. It can help to reduce the social cost of road network operation by reducing the number of crashes and victims.

Author Contributions: Conceptualization, M.K. and C.D.; methodology, M.K. and C.D.; formal analysis, C.D.; investigation, All; data curation, M.K. and S.P.; writing—original draft preparation, M.K. and S.P.; writing—review and editing, All; visualization, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mehdizadeh, A.; Cai, M.; Hu, Q.; Alamdar Yazdi, M.A.; Mohabbati-Kalejahi, N.; Vinel, A.; Rigdon, S.E.; Davis, K.C.; Megahed, F.M. A Review of Data Analytic Applications in Road Traffic Safety. Part 1: Descriptive and Predictive Modeling. *Sensors* **2020**, *20*, 1107. [CrossRef] [PubMed]
2. Cafiso, S.; D'Agostino, C.; Kiec, M.; Pogodzinska, S. Application of an Intelligent Transportation System in a Travel Time Information System: Safety Assessment and Management. *Transp. Res. Rec. J. Transp. Res. Board.* **2017**, *2635*, 46–54. [CrossRef]
3. Elvik, R.; Høye, A.; Vaa, T.; Sørensen, M. *Handbook of Road Safety Measure*, 2nd ed.; Emerald Group Bingley: Bingley, UK, 2009.
4. Shi, C.; Chen, B.Y.; Lam, W.H.K.; Li, Q. Heterogeneous Data Fusion Method to Estimate Travel Time Distributions in Congested Road Networks. *Sensors* **2017**, *17*, 2822. [CrossRef] [PubMed]
5. Høye, A.; Sørensen, M.; Elvik, R.; Akhtar, J.; Nævestad, T.; Vaa, T. Evaluation of Variable Message Signs in Trondheim. Available online: <https://www.toi.no/getfile.php/1317731-1308915837/Publikasjoner/T%C3%98I%20rapporter/2011/1153-2011/1153-2011-sum.pdf> (accessed on 23 July 2020).
6. Stoneman, B. The Effects of Dynamic Route Guidance in London. Available online: <http://trl.demo.varistha.co.uk/uploads/trl/documents/RR348.pdf>. (accessed on 23 July 2020).
7. Vaa, T.; Gelau, C.; Penttinen, M.; Spyroupolou, I. ITS and effects on road traffic crashes—State of the art. In Proceedings of the 13th World Congress on ITS, London, UK, 8–12 October 2006.
8. Wang, J.; Rakha, H. Empirical Study of Effect of Dynamic Travel Time Information on Driver Route Choice Behavior. *Sensors* **2020**, *20*, 3257. [CrossRef] [PubMed]
9. Khan, Z.; Koubaa, A.; Farman, H. Smart Route: Internet-of-Vehicles (IoV)-Based Congestion Detection and Avoidance (IoV-Based CDA) Using Rerouting Planning. *Appl. Sci.* **2020**, *10*, 4541. [CrossRef]
10. Maher, M.J.; Hughes, P.C.; Smith, M.J.; Ghali, M.O. Accident- and Travel Time-Minimising Routeing Patterns in Congested Networks. *Traffic Eng. Control.* **1993**, *34*, 414–419.
11. Abdulhai, B.; Look, H. Safety benefits of dynamic route guidance: Boon or boondoggle? In Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, Singapore, 6 September 2002; pp. 544–548. [CrossRef]
12. Chatterjee, K.; McDonald, M. The Network Safety Effects of Dynamic Route Guidance. *ITS J.* **1999**, *4*, 161–185. [CrossRef]
13. Abdulhai, B.; Look, H. Impact of Dynamic and Safety-Conscious Route Guidance on Accident Risk. *J. Transp. Eng.* **2003**, *129*, 369. [CrossRef]
14. Cafiso, S.; D'Agostino, C.; Persaud, B. Investigating the influence of segmentation in estimating safety performance functions for roadway sections. *J. Traffic Transp. Eng.* **2018**, *5*, 129–136. [CrossRef]
15. D'Agostino, C. Investigating transferability and goodness of fit of two different approaches of segmentation and model form for estimating safety performance of Motorways. *Procedia Eng.* **2014**, *84*, 613–623. [CrossRef]
16. Hauer, E. *The Art of Regression Modeling in Road Safety*; Springer Nature: Cham, Switzerland, 2019. [CrossRef]
17. Kulmala, R. Safety at rural three- and four-arm junctions: Development and application of accident prediction models: Dissertation. Ph.D. Thesis, Helsinki University of Technology, Otaniemi, Espoo, Finland, June 1995.
18. Nicholson, A.; Turner, S. Estimating Accidents in a Road Network. In Proceedings of the Roads 96 Conference, Christchurch, New Zealand, 2–6 September 1996.
19. Dunlop, D.D. Regression for Longitudinal Data: A Bridge from Least-Squares Regressions. *Am. Stat.* **1994**, *48*, 299–303. [CrossRef]
20. Cafiso, S.; D'Agostino, C.; Kiec, M. Investigating the Influence of Passing Relief Lane Sections on Safety and Traffic Performance. *J. Transp. Health* **2017**, *7*, 38–47. [CrossRef]
21. McCullagh, P.; Nelder, J.A. *Generalized Linear Models*, 2nd ed.; Chapman & Hall/CRC: London, UK, 1989.
22. Myers, R. *Classical and Modern Regression with Applications*, 2nd ed.; Duxbury Press: Boston, MA, USA, 1990.

23. Cafiso, S.; D'Agostino, C. Safety Performance Function for Motorways using Generalized Estimation Equations. *Procedia Soc. Behav. Sci.* **2012**, *53*, 901–910. [[CrossRef](#)]
24. Lord, D.; Persaud, B. Accident Prediction Models With and Without Trend: Application of the Generalized Estimating Equations (GEE) Procedure. *Transp. Res. Rec. J. Transp. Res. Board.* **2000**, *1717*, 102–108. [[CrossRef](#)]
25. Cafiso, S.; D'Agostino, C. A Stochastic Approach to the Benefit Cost Ratio Analysis of Safety Treatments. *Case Stud. Transp. Policy* **2018**, *8*, 188–196. [[CrossRef](#)]
26. Cafiso, S.; D'Agostino, C.; Kiec, M. Investigating safety performance of the SAFESTAR system for route-based curve treatment. *Reliab. Eng. Syst. Saf.* **2019**, *188*, 125–132. [[CrossRef](#)]
27. Albalade, D.; Fageda, X. Congestion, Road Safety, and the Effectiveness of Public Policies in Urban Areas. *Sustainability* **2019**, *11*, 5092. [[CrossRef](#)]
28. TRB. Highway Capacity Manual Sixth Edition: A Guide for Multimodal Mobility Analysis. *Tr News* **2016**, *86*, 14–18.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Estimation of Traffic Stream Density Using Connected Vehicle Data: Linear and Nonlinear Filtering Approaches

Mohammad A. Aljamal ¹, Hossam M. Abdelghaffar ^{2,3} and Hesham A. Rakha ^{1,*}

¹ Charles E. Via, Jr. Department of Civil and Environmental Engineering, Center for Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA; m7md92@vt.edu

² Department of Computer Engineering and Systems, Engineering Faculty, Mansoura University, Mansoura 35516, Egypt; hossam_wahed@mans.edu.eg or hossamvt@vt.edu

³ Center for Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA

* Correspondence: hrakha@vt.edu

Received: 4 June 2020; Accepted: 17 July 2020; Published: 22 July 2020

Abstract: The paper presents a nonlinear filtering approach to estimate the traffic stream density on signalized approaches based solely on connected vehicle (CV) data. Specifically, a particle filter (PF) is developed to produce reliable traffic density estimates using CV travel-time measurements. Traffic flow continuity is used to derive the state equation, whereas the measurement equation is derived from the hydrodynamic traffic flow relationship. Subsequently, the PF filtering approach is compared to linear estimation approaches; namely, a Kalman filter (KF) and an adaptive KF (AKF). Simulated data are used to evaluate the performance of the three estimation techniques on a signalized approach experiencing oversaturated conditions. Results demonstrate that the three techniques produce accurate estimates—with the KF, surprisingly, being the most accurate of the three techniques. A sensitivity of the estimation techniques to various factors including the CV level of market penetration, the initial conditions, and the number of particles in the PF is also presented. As expected, the study demonstrates that the accuracy of the PF estimation increases as the number of particles increases. Furthermore, the accuracy of the density estimate increases as the level of CV market penetration increases. The results indicate that the KF is least sensitive to the initial vehicle count estimate, while the PF is most sensitive to the initial condition. In conclusion, the study demonstrates that a simple linear estimation approach is best suited for the proposed application.

Keywords: traffic density; connected vehicles; real-time estimation; particle filter; Kalman filter

1. Introduction

Real-time traffic estimation has received increased attention with the introduction of advanced applications and technologies such as intelligent transportation systems (ITSs). Adaptive traffic signal controllers require real-time traffic state estimation to improve intersection performance, as real-time estimation plays a major role in capturing variations in traffic behavior (e.g., nonrecurrent changes). As inputs to traffic signal controllers, traffic state variables (e.g., travel time and traffic density) assist with green time allocation and help to enhance intersection performance by reducing traffic delays, vehicle emissions, and fuel consumption [1,2]. Several estimation techniques have been developed to estimate traffic state variables [3–7]. In some previous studies, the traffic state system has been treated as a linear system [4,7–10]. Other studies have considered the system as nonlinear [3,11–13]. For linear system models, a Kalman filter (KF) has been widely deployed to produce accurate estimates [4,7,10,14] due to its simplicity and applicability in the field. The KF assumes linear system transitions with a

Gaussian distribution for the probability density function (PDF) of the system and measurement noise. For nonlinear system models, an extended KF (EKF) has been utilized in estimation [11,15,16]. The EKF also assumes that the PDF distribution is Gaussian. The EKF is derived by linearizing the system using a Taylor series expansion by calculating the Jacobian expression. However, it was found that use of the EKF approach is only valid if the system is near linearity during the updating time [17], and thus, large errors may result from linearization. In addition, the task of deriving the Jacobian matrices may cause implementation difficulties [18]. A more robust nonlinear approach is a particle filter (PF), which has been frequently employed in the literature to handle nonlinear dynamic problems [12,19,20]. The PF approach is a Monte Carlo sequential solution that deals with nonlinear system transitions without the assumption of the PDF noise distribution [21,22]. In this paper, a PF approach is developed to estimate the traffic stream density along signalized intersection approaches using only connected vehicle (CV) data. Moreover, the paper compares the performance of the PF to the KF and adaptive KF (AKF) approaches.

Traffic density is defined as the number of vehicles per unit length on a specific roadway segment [23]. Estimating the traffic stream density is critical in the development of effective traffic controllers [24]. For instance in the case of freeways, identifying bottleneck locations in the early stages is critical in developing congestion mitigation strategies that include ramp metering, variable speed limits, and traffic routing. For signalized segments, the traffic density measures are crucial for either traffic signal performance [25–27] or traffic signal optimization [28–30]. Hence, traffic density measures must be precisely estimated to represent traffic demands at each signalized intersection approach. Once accurate measurements are obtained, efficient adaptive traffic signal controllers can be developed. However, determining traffic density is not a trivial task and cannot be directly measured in the field since it is a spatial measurement. Consequently, traffic stream density is typically based on estimations.

Previous research has utilized different data sources, such as stationary sensors (e.g., loop detectors), fused data (combining two distinct data sources), and CV data to estimate traffic stream density. Traffic density estimates can be measured using video detection systems, but this is difficult due to the high cost of the infrastructure and the limited visibility of roadway segments [4]. Time-occupancy measurements from loop detectors are used as an alternative data source to estimate the traffic density [31]. However, time-occupancy measurements only represent the temporal density estimates around the location of the detector. A recent study introduced a relationship between time-occupancy and space-occupancy to estimate traffic density by dividing the link into small segments and installing detectors on all of the small segments [32], but the installation cost is high. A more common way of estimating the traffic density is the use of the traffic flow continuity equation (input–output approach), which considers two traffic counting stations, one at the entrance and the other at the end of the link [33]. Vigos et al. [4] proposed a robust linear KF approach with at least three loop detectors to estimate the traffic density along signalized approaches. However, the implementation cost is high. Another study [34] employed two conventional loop detectors to estimate the traffic density using the flow continuity equation. The two loop detectors provide the estimation model with traffic flow and occupancy data. Bhouri et al. [35] proposed a KF approach to estimate the traffic stream density along a freeway segment using both loop detectors and a recorded film [35]. One commonality about the use of fixed sensors is that they are subject to detection failures and thus always produce errors in their data [36,37].

Recent research has fused different data sources to estimate the traffic stream density along certain roadway sections, increasing the accuracy of the estimate over using just one data source [7,14]. Many works have employed the KF approach [14,38,39]. For instance, traffic flow data at the entrance and the exit of the roadway section observed from stationary sensors together with CV data were used to estimate the traffic density [38]. The CV data provided travel-time measurements to correct the prior estimate from the state equation. Another study has utilized fused loop and CV measurements to estimate the traffic density in a freeway section [19]. In that study, the authors derived the estimation

model using the PF estimation approach, considering two sources of measurements: (1) loop detectors, and (2) fusing loop detectors and CVs. They obtained a 30% reduction in the mean absolute percentage error from the fused measurements compared to the measurements from loop detectors, demonstrating that more data sources produce more-accurate outcomes. However, the use of different data sources requires more computational cost in both time and memory as the data include both trivial (data that are not needed) and nontrivial (data that are needed) information.

Limited studies have used CV data as the only source of inputs to estimate the traffic stream density [7,9,10]. These studies developed the linear KF estimator approach. The CV data used were the number of CVs at the entry and at the exit of the tested roadway section, in addition to the travel time experienced for the CVs to traverse the tested section. Moreover, Aljamal et al. [7] demonstrated that treating the estimation interval time as a variable instead of a fixed value is mandatory when dealing with only CV data, as the variable approach always ensures that sufficient information is gathered from the CVs in every estimation interval. This approach enhances the accuracy of the estimation, especially for the scenarios with low CV level of market penetration (LMP) rate. The estimation time interval for this study is therefore defined as the time when an exact number of CVs (i.e., 5 vehicles) reach the end of the tested link.

Several researchers have employed the PF approach to improve traffic stream estimates for different transportation applications, including traffic flow [12,13], travel time [3,20], and traffic speed [40]. In one study, magnetic loop detectors were placed at the boundaries of the tested freeway section to estimate the traffic flow, and a PF estimation approach was developed using traffic flow and speed measurements [12]. In another study, Mihaylova et al. [13] developed two nonlinear approaches, an unscented KF and a PF, to produce real-time traffic flow estimates in a freeway network using data from stationary sensors. They found that the PF approach outperformed the unscented KF. Chen et al. [20] proposed a time series speed equation to estimate traffic speed. They claimed that the traffic system is nonlinear and thus presented two nonlinear approaches, a PF and an ensemble KF, using available speed measurements from loop detectors. They found that the PF approach is more accurate than the ensemble KF. Another study developed a PF estimation approach for travel-time predictions using real-time and historical data [3]. They used the historical data to generate particles as opposed to using a state-transition model. In addition, a comparison between the PF, KF, and k-nearest neighbor estimators found that the PF is the most accurate approach. CV data were employed to estimate the traffic speed and flow using the PF approach [40]. In that study, each link in the network was assumed to have base stations to retrieve and transfer the data. Results found that other data sources (e.g., loop detectors) should be incorporated with CVs to enhance the estimation performance. However, our recent study developed a KF approach, showing that the use of CV data alone is sufficient to obtain accurate results [7].

In summary, the existing literature shows that the PF has been widely used to address nonlinear systems and has been proven to outperform other nonlinear estimation techniques; however, to our best of knowledge in the application of traffic stream density estimation, only a few studies have applied the PF approach using data from stationary sensors and fusing data from different sources. In addition, no comparison between the PF and the linear KF has been reported. Therefore, the PF was adopted in this study. The primary objective of this study is to develop a nonlinear PF estimation approach to estimate the traffic stream density based solely on CV data on signalized approaches. Subsequently, we compare the PF approach to linear estimation approaches—namely, KF and AKF—to identify the best approach for the application of the traffic density estimation, given that no comparison has been reported in the literature between these filtering techniques. Consequently, this research will recommend a specific approach to estimate the traffic stream density. The proposed three approaches are employed to estimate the vehicle counts based solely on CV data. In addition, this study also investigates the sensitivity of the proposed estimation approaches to several factors, such as the LMP rate of the CVs, the initial conditions, and the number of PF particles.

The paper is organized as follows: Section 2 describes the problem formulation and the estimation approaches. Section 3 discusses the findings from applying the estimation approaches. Section 4 includes the conclusions of the study and the proposed future work.

2. Problem Formulation and Estimation Approaches

First, Section 2.1 formulates the research problem using a state-space model. Then, three different estimation approaches are described: the PF (Section 2.2.1), the KF (Section 2.2.2) and the AKF (Section 2.2.3).

2.1. State-Space Model

The state-space model is represented by a state equation and a measurement equation. The state equation describes how the system behaves and provides a prior knowledge of the estimation. The measurement equation is used to help correct and improve the prior estimation. In this study, the goal is to estimate the number of vehicles on signalized links using only CV data, as depicted in Figure 1, where CVs are the vehicles that have the connection icon (e.g., the first vehicle on the left). The only information that is needed in practice is as follows: (1) the traffic flow of CVs observed at the tested link's entrance and exit. (2) the travel time of each CV. Vehicle-to-Infrastructure (V2I) communication can provide this information to the traffic signal controller.

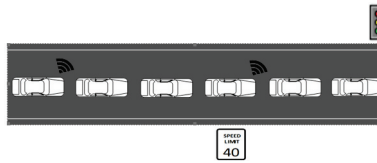


Figure 1. Tested link section includes connected vehicles (CVs) and non-CVs.

The model is formulated using the derived state-space equations in [7]. The state equation, Equation (1), is based on the continuity equation of traffic flow; whereas the measurement equation, Equation (2), is based on the traffic flow hydrodynamic relationship, based on measurements of the average travel time of CVs. In Equation (1), the number of vehicles is computed by continuously adding the difference of the number of vehicles that enter and exit the tested section to the cumulative number of vehicles traveling along the section previously computed.

$$N(t) = N(t - \Delta t) + u(t) \quad (1)$$

$$TT(t) = H(t) \times N(t) \quad (2)$$

where $N(t)$ is the number of vehicles traversing the link at time t , $N(t - \Delta t)$ is the number of vehicles traversing the link in the preceding time interval, and $u(t)$ is the system inputs, as described in Equation (3).

$$u(t) = \frac{\Delta t [q^{in}(t) - q^{out}(t)]}{\max(\rho_{actual}, \rho_{min})} \quad (3)$$

where q^{in} and q^{out} represent the flow of CVs entering and exiting the link, respectively, during Δt . ρ is the CVs' LMP, defined as the ratio of CV count to total vehicle count. In the state equation, the ρ variable is set to be the maximum number of the actual ρ (ρ_{actual}) and a predefined minimum value of ρ (ρ_{min}). ρ_{actual} can be obtained from historical data. ρ_{min} is introduced to avoid producing large errors in the state equation since a single ρ value is used to approximate the two ρ values (upstream and downstream of the tested link) [7]. In this study, ρ_{min} is set to be equal to 0.5; more details about the system state representation can be found in [7]. It should be noted that the ρ variable is the main noise source in the system, and thus, there is an urgent need to develop the measurement equation to

fix these errors. In Equation (2), TT is the average vehicle travel time, $H(t)$ is a vector that transforms the vehicle counts to travel times. $H(t)$ is derived from the hydrodynamic relationship between the macroscopic traffic parameters (flow, density, and space-mean speed), as presented in Equation (4).

$$H(t) = \frac{1}{\bar{q}(t)} = \frac{2 \times \rho_{actual}}{q^{in}(t) + q^{out}(t)} \quad (4)$$

2.2. Estimation Approaches

As mentioned earlier, the KF and the AKF are considered linear estimators that can efficiently handle linear state-space systems. However, in the proposed state-space equations, we suspect some nonlinearity coming from the ρ variable, which raises the question, would a nonlinear filter improve the estimation performance? For this purpose, this study develops a nonlinear PF approach to estimate the vehicle counts along the signalized link. This section presents the formulation of the three approaches used to estimate the vehicle counts using only CV data along signalized approaches. The three techniques are the proposed PF, the KF [7], and the AKF [10].

2.2.1. The PF Approach

The PF approach is used to solve nonlinear state-space systems with no form restrictions on the initial state and noise distributions. For instance, the PF can deal with any arbitrary PDF distribution [21]. The PF approach is used to estimate the posterior PDF of the state vehicle count variable (N) given some measurements of CV travel times (TT) by assigning k number of particles (samples). Each particle has a certain relative weight (w). When a new measurement is received, the particles' locations and weights are updated. It should be noted that the particles with low relative weight values are replaced with new particles (resampling) so that the system keeps only the important particles. The estimates are then calculated using the average value of the remaining particles. The following steps are used to implement the proposed PF approach:

1. Initialization: $t = 0$; where t is the time interval.

- (a) $\hat{N}^+(0)$, R , V , and k ,
where $\hat{N}^+(0)$ is the initial vehicle count estimate; R is the measurement's covariance error; and V is the variance of the initial vehicle count estimate, which is used to randomly generate the initial particles' locations around $\hat{N}^+(0)$.
- (b) Generate k particles' locations randomly, from 1 to K , from the initial prior Gaussian distribution $P(N_0)$.

$$N^k(0) \sim P(N_0) \quad (5)$$

2. For $t = 1 : T$.

- (a) Update the locations ($N^k(t)$), measurements ($TT^k(t)$), and weights ($w^k(t)$) of the particles.

$$N^k(t) = N^k(t - \Delta t) + u(t) \quad (6)$$

$$TT^k(t) = H(t) \times N^k(t) \quad (7)$$

$$w^k(t) = \frac{1}{\sqrt{2\pi R}} e^{-(TT - TT^k(t))^2 / 2R} \quad (8)$$

where TT is the observed measurement from the CVs. The weights are then normalized using the following equation, $\hat{w}^k(t) = w^k(t) / \sum_{k=1}^K w^k(t)$.

- (b) Replace the low-weighted particles with new particles (resampling [21]). After a few iterations in the PF process, the weight will focus on a few particles only and most particles will have insignificant weights, resulting in sample degeneracy [41]. The resampling process is therefore used to tackle the degeneracy problem. It should be noted that the highly weighted particles are used to compute the PF posterior estimate.
- (c) Compute the PF posterior estimate: The PF posterior estimate is computed as the average value of the remaining particles (particles with high weights), as shown in Equation (9).

$$\hat{N}^+(t) = \frac{1}{K} \sum_{k=1}^K N^k(t) \tag{9}$$

- (d) Next time step ($t + \Delta t$): When 5 new CVs traverse the link, return to step 2a.

2.2.2. The KF Approach

The KF approach is a linear quadratic estimator. It has been proven to be the best for estimating linear systems with Gaussian noise [42]. The KF estimation approach can be solved using the following steps:

1. Initialization: $t = 0$; where t is the time interval.

- (a) $\hat{N}^+(0)$, R , and $\hat{P}^+(0)$,
where $\hat{P}^+(0)$ is the initial posterior error covariance estimate for the state system.

2. For $t = 1 : T$.

- (a) Prior estimates:

$$\hat{N}^-(t) = \hat{N}^+(t - \Delta t) + u(t) \tag{10}$$

$$\hat{T}T(t) = H(t) \times \hat{N}^-(t) \tag{11}$$

$$\hat{P}^-(t) = \hat{P}^+(t - \Delta t) \tag{12}$$

where \hat{N}^- is an estimate of a priori vehicle count, $\hat{T}T$ is the estimated average travel time, and \hat{P}^- is the a priori covariance estimate for the state system.

- (b) Correction: The correction uses the prior estimate and the new measurement (i.e., the CV average travel time) to compute the Kalman gain (G).

$$G(t) = \hat{P}^-(t)H(t)^T [H(t)\hat{P}^-(t)H(t)^T + R]^{-1} \tag{13}$$

- (c) Posterior state estimates:

$$\hat{N}^+(t) = \hat{N}^-(t) + G(t) [TT(t) - \hat{T}T(t)] \tag{14}$$

$$\hat{P}^+(t) = \hat{P}^-(t) \times [1 - H(t)G(t)] \tag{15}$$

where \hat{N}^+ is the posterior vehicle count estimate, and \hat{P}^+ is the posterior error covariance estimate.

- (d) Next time step ($t + \Delta t$): When 5 new CVs traverse the link, return to step 2a.

2.2.3. The AKF Approach

The AKF approach is presented to estimate the total number of vehicles, using real-time noise error estimates in the state and measurement systems (i.e., mean and variance values). It should be noted that the KF and the AKF approaches use the same equations, but the AKF approach dynamically estimates the noise statistical parameters every estimation step. The vehicle count estimates can be obtained using the following steps:

1. Initialization: $t = 0$; where t is the time interval.

- (a) $\hat{N}^+(0)$, $m(0)$, and $\hat{P}^+(0)$,
 where $m(0)$ is the mean of the noise for the state system.

2. For $t = 1 : T$

(a) Prior estimates:

$$\hat{N}^-(t) = \hat{N}^+(t - \Delta t) + u(t) + m(t - \Delta t) \tag{16}$$

$$\hat{P}^-(t) = \hat{P}^+(t - \Delta t) + M(t - \Delta t) \tag{17}$$

(b) Estimation of noise statistics for the measurement system:

$$\hat{T}T(t) = H(t) \times \hat{N}^-(t) \tag{18}$$

$$r = \frac{1}{n} \sum_{t=1}^n [TT(t) - \hat{T}T(t)] \tag{19}$$

$$R = \frac{1}{n-1} \sum_{t=1}^n [(r(t) - r) \cdot (r(t) - r)^T - (\frac{n-1}{n})H(t)\hat{P}^-(t)H^T(t)] \tag{20}$$

where r and R are the mean and covariance of the measurement noise, respectively, and n is the number of state noise samples.

(c) Correction:

$$G(t) = \hat{P}^-(t)H(t)^T [H(t)\hat{P}^-(t)H(t)^T + R(t)]^{-1} \tag{21}$$

(d) Posterior state estimates:

$$\hat{N}^+(t) = \hat{N}^-(t) + G(t) [TT(t) - \hat{T}T(t) - r(t)] \tag{22}$$

$$\hat{P}^+(t) = \hat{P}^-(t) \times [1 - H(t)G(t)] \tag{23}$$

(e) Estimation of noise statistics for the state system:

$$m = \frac{1}{n} \sum_{t=1}^n [\hat{N}^+(t) - \hat{N}^+(t - \Delta t) - u(t) + m(t - \Delta t)] \tag{24}$$

$$M = \frac{1}{n-1} \sum_{t=1}^n [(m(t) - m) \cdot (m(t) - m)^T - (\frac{n-1}{n})\hat{P}^+(t - \Delta t) - \hat{P}^+(t)] \tag{25}$$

where m and M are the mean and covariance of the state noise, respectively.

(f) Next time step ($t + \Delta t$): When 5 new CVs traverse the link, return to step 2a.

3. Results and Discussion

This section evaluates and compares the three estimation approaches. The simulated data were generated for a signalized link under an oversaturation condition in which the traffic demand exceeds the link capacity. The free-flow speed is 40 km/h; the saturation flow rate is 1800 veh/h/lane, resulting in a traffic capacity of 855 veh/h given the cycle length and traffic signal's green times; the speed-at-capacity is 32 km/h; and the jam density is 160 veh/km/lane. The traffic signal is operated at a cycle length of 120 s and a phase split of 50:50. The amber and all-red intervals are 3 s. To test the accuracy of the estimation approaches, the INTEGRATION microscopic traffic assignment and simulation software was used [43,44]. The relative root mean square error (RRMSE), presented in Equation (26), was used to evaluate the proposed estimation approaches.

$$\text{RRMSE}(\%) = 100 \frac{\sqrt{S \sum_{s=1}^S [\hat{N}^+(s) - N(s)]^2}}{\sum_{s=1}^S N(s)} \quad (26)$$

where $\hat{N}^+(s)$ represents the estimated count of vehicles, $N(s)$ represents the actual count of vehicles, and S is the overall number of estimations.

3.1. Performance of Estimation Approaches

The simulations were conducted with the same predefined initial conditions to obtain a fair comparison. The initial conditions are described in Table 1. It should be noted that each estimator requires specific initial variables. For instance, $\hat{N}^+(0)$, R , and $\hat{P}^+(0)$ are required for the KF approach. For all estimation approaches, the first estimate begins with an erroneous initial estimate of vehicle count ($\hat{N}^+(0) = 5$ veh), whereas the actual vehicle count is zero [4,7].

Table 1. Initial conditions for the Kalman filter (KF), adaptive KF (AKF), and particle filter (PF) approaches.

| Initial Conditions | KF | AKF | PF |
|-----------------------------------|----|-----|-----|
| $\hat{N}^+(0)$ (veh) | 5 | 5 | 5 |
| R (s^2) | 20 | – | 20 |
| V (veh^2) | – | – | 5 |
| k (# of part.) | – | – | 200 |
| $\hat{P}^+(0)$ (veh^2) | 5 | 5 | – |
| m (veh) | – | 5 | – |

The three estimation approaches were evaluated using different CV LMPs, including 1%, 3%, 5%, 8%, 10%, 15%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%. For each LMP scenario, 100 random samples from the full data set were created using a Monte Carlo simulation. Table 2 presents the RRMSE values of the KF, AKF, and the PF approaches. The table indicates that estimation errors decrease with increasing LMP for all estimation approaches. The table also demonstrates that the KF outperforms the AKF and the PF approaches. For instance, for the scenario of 1% LMP, the vehicle count estimates were off by 30%, 48% and 64% using KF, AKF and PF, respectively.

Table 2. Relative root mean square error (RRMSE) of KF, AKF, and PF approaches for different levels of market penetration rate (LMPs).

| LMPs % | RRMSE (%) | | |
|--------|-----------|-----|----|
| | KF | AKF | PF |
| 1 | 30 | 48 | 64 |
| 3 | 25 | 34 | 60 |
| 5 | 23 | 32 | 56 |
| 8 | 23 | 28 | 52 |
| 10 | 19 | 24 | 48 |
| 15 | 19 | 24 | 42 |
| 20 | 18 | 23 | 40 |
| 30 | 18 | 19 | 30 |
| 40 | 18 | 18 | 22 |
| 50 | 18 | 17 | 18 |
| 60 | 14 | 16 | 15 |
| 70 | 12 | 17 | 12 |
| 80 | 9 | 17 | 9 |
| 90 | 6 | 17 | 7 |

The PF approach produces high RRMSE values at low LMPs ($LMP < 40\%$), while for the high-LMP scenarios, the PF produces RRMSE values close to the values obtained from the KF. Moreover, the AKF approach produces high errors, especially at very low LMPs ($LMP < 10\%$) and high LMPs ($LMP \geq 70\%$). This demonstrates that the real-time estimates of the statistical noise values obtained from the AKF are not needed for the high-LMP scenarios, and the user may proceed with predefined statistical values due to low errors in the vehicle count estimates (low error in the ρ value). It was found that the high RRMSE error values produced from the AKF and PF approaches are mainly caused from assigning an inappropriate initial vehicle count estimate, as discussed in the next section.

Figure 2 presents the KF, AKF, and PF estimation outcomes with regard to the actual values at different LMPs (i.e., 10% to 90% with an increase of 10%). In each subfigure, three plots are generated to display the estimation approaches' outcomes with regard to the actual values; the top one displays the PF outcomes, the middle one presents the KF outcomes, and the bottom one displays the AKF outcomes. The actual curve is represented by the dotted curve. In conclusion, the KF approach is recommended, as it produces the most accurate estimates in addition to its simplicity and applicability in the field. The next section will discuss the impact of the initial conditions on the performance of the various estimation approaches.

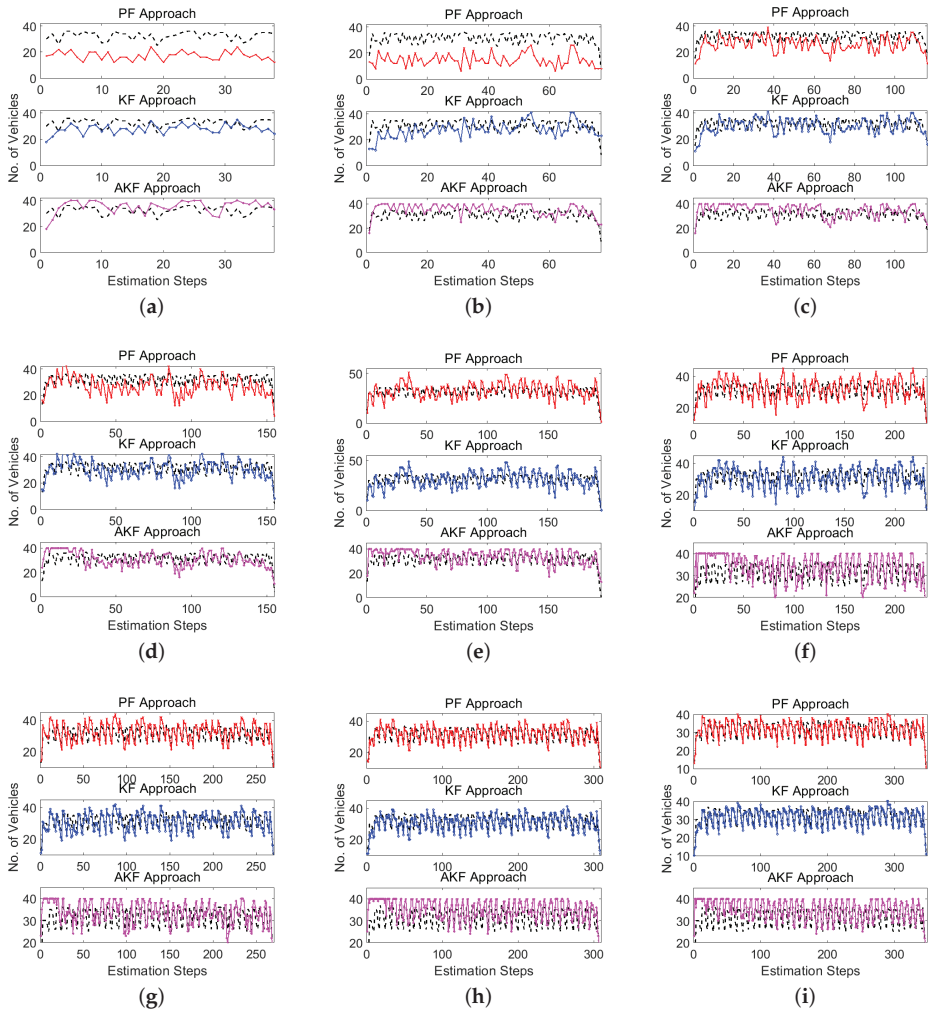


Figure 2. Actual and estimated vehicle counts at different LMP scenarios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, (e) 50%, (f) 60%, (g) 70%, (h) 80% and (i) 90%.

3.2. Impact of Initial Conditions

This section examines the effect of the choice of the initial conditions on the performance of the estimators, such as the initial vehicle count estimate $\hat{N}^+(0)$ and the k number of particles in the PF approach. First, different $\hat{N}^+(0)$ values were tested, from 0 to 25 at increments of 5, at different LMP scenarios, as presented in Tables 3 and 4. Table 3 presents the RRMSE values when the $\hat{N}^+(0)$ is set to equal 0, 5, and 10 vehicles. Table 4 displays the RRMSE for the $\hat{N}^+(0)$ values of 15, 20, and 25 vehicles. The tables demonstrate that the RRMSE values are sensitive to the changes of the $\hat{N}^+(0)$ values. The tables also show that the PF is the most sensitive estimator to $\hat{N}^+(0)$ for all LMP scenarios. For instance, for the scenario of 1% LMP, the RRMSE is 81% when the simulation starts with 0 veh, while the RRMSE is 17% when $\hat{N}^+(0)$ is equal to 25. Therefore, starting the simulations with an appropriate initial estimate close to the truth value significantly improves the estimation accuracy since this helps the PF to quickly converge. In addition, the AKF seems to be sensitive to the $\hat{N}^+(0)$

with low LMP scenarios (LMP $\leq 10\%$), while the choice of $\hat{N}^+(0)$ has a slight effect on the estimation accuracy for the scenarios with medium and high LMPs. For instance, for the scenario of 1% LMP, the RRMSE is 71% when the simulation starts with 0 veh, while the RRMSE is 21% when $\hat{N}^+(0)$ is equal to 25. Lastly, the tables show that the KF is the least-sensitive estimator to the $\hat{N}^+(0)$ value. Figure 3 summarizes the RRMSE values for nine LMP scenarios presented in Tables 3 and 4.

Table 3. RRMSE values for the KF, AKF, and PF approaches using different initial vehicle count estimates (i.e., 0, 5 and 10) for different LMPs.

| LMPs % | $\hat{N}^+(0) = 0$ | | | $\hat{N}^+(0) = 5$ | | | $\hat{N}^+(0) = 10$ | | |
|--------|--------------------|-----|----|--------------------|-----|----|---------------------|-----|----|
| | KF | AKF | PF | KF | AKF | PF | KF | AKF | PF |
| 1 | 34 | 71 | 81 | 30 | 48 | 64 | 27 | 36 | 51 |
| 3 | 28 | 49 | 78 | 25 | 34 | 60 | 23 | 26 | 47 |
| 5 | 26 | 45 | 73 | 23 | 32 | 56 | 23 | 27 | 44 |
| 8 | 24 | 33 | 69 | 23 | 28 | 52 | 23 | 27 | 41 |
| 10 | 19 | 33 | 62 | 19 | 24 | 48 | 20 | 24 | 37 |
| 15 | 21 | 29 | 55 | 19 | 24 | 42 | 20 | 23 | 37 |
| 20 | 20 | 24 | 47 | 18 | 23 | 40 | 19 | 23 | 35 |
| 30 | 19 | 21 | 34 | 18 | 19 | 30 | 19 | 19 | 27 |
| 40 | 18 | 20 | 24 | 18 | 18 | 22 | 19 | 18 | 19 |
| 50 | 19 | 17 | 18 | 18 | 17 | 18 | 17 | 17 | 22 |
| 60 | 14 | 17 | 14 | 14 | 16 | 15 | 15 | 16 | 19 |
| 70 | 12 | 18 | 12 | 12 | 17 | 12 | 12 | 17 | 17 |
| 80 | 9 | 18 | 9 | 9 | 17 | 9 | 9 | 17 | 15 |
| 90 | 6 | 17 | 6 | 6 | 17 | 7 | 7 | 17 | 14 |

Table 4. RRMSE values for the KF, AKF, and PF approaches using different initial vehicle count estimates (i.e., 15, 20 and 25) for different LMPs.

| LMPs % | $\hat{N}^+(0) = 15$ | | | $\hat{N}^+(0) = 20$ | | | $\hat{N}^+(0) = 25$ | | |
|--------|---------------------|-----|----|---------------------|-----|----|---------------------|-----|----|
| | KF | AKF | PF | KF | AKF | PF | KF | AKF | PF |
| 1 | 23 | 32 | 36 | 20 | 23 | 24 | 19 | 21 | 17 |
| 3 | 22 | 26 | 33 | 20 | 25 | 24 | 20 | 24 | 19 |
| 5 | 21 | 25 | 31 | 20 | 23 | 26 | 19 | 24 | 20 |
| 8 | 21 | 27 | 33 | 22 | 26 | 26 | 21 | 26 | 23 |
| 10 | 20 | 24 | 30 | 20 | 24 | 27 | 19 | 26 | 22 |
| 15 | 19 | 23 | 30 | 19 | 24 | 26 | 19 | 23 | 18 |
| 20 | 19 | 23 | 30 | 19 | 23 | 30 | 19 | 23 | 16 |
| 30 | 19 | 19 | 21 | 19 | 19 | 21 | 19 | 19 | 26 |
| 40 | 19 | 18 | 20 | 19 | 18 | 20 | 19 | 18 | 44 |
| 50 | 18 | 17 | 33 | 18 | 17 | 47 | 18 | 17 | 33 |
| 60 | 15 | 16 | 32 | 15 | 16 | 32 | 15 | 16 | 32 |
| 70 | 12 | 17 | 30 | 12 | 17 | 30 | 12 | 17 | 30 |
| 80 | 9 | 17 | 30 | 9 | 17 | 30 | 9 | 17 | 30 |
| 90 | 7 | 17 | 29 | 7 | 17 | 44 | 7 | 17 | 29 |

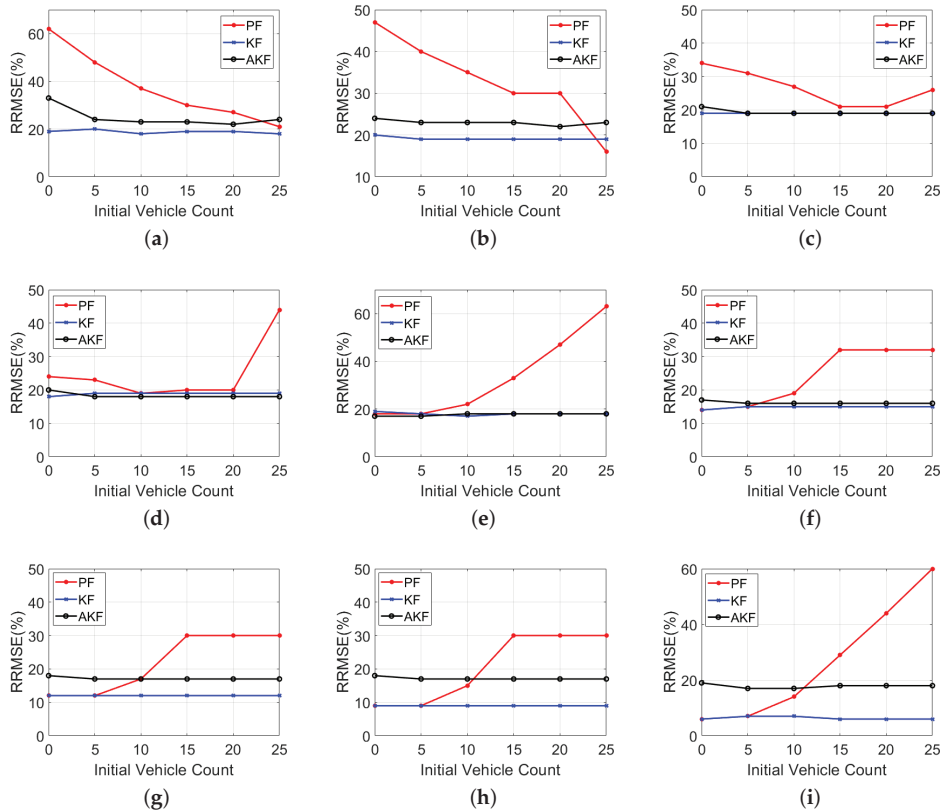


Figure 3. RRMSE values using various initial vehicle count estimates at different LMP scenarios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, (e) 50%, (f) 60%, (g) 70%, (h) 80% and (i) 90%.

This study also examined the choice of the number of particles, k , on the PF performance (i.e., $k = 10, 100, 200, 1000$, and 2000), as presented in Table 5. The findings show that the estimation accuracy increases as the number of particles increases, especially at low LMPs. However, increasing the number of particles is associated with additional computational time. The PF is implemented in MATLAB R2019a on a Dell PC with 8.0 GB RAM. The computation time ranges between 0.2 and 1.6 s, with 10 particles for various LMPs; 1.1 and 3.0 s with 100 particles; 1.3 and 6.8 sec with 200 particles; 1.3 and 73 s with 1000 particles; 4 and 256 s with 2000 particles. The results in Table 5 show that the use of 1000 and 2000 particles slightly reduces the RRMSE values compared to the use of 200 particles; however, this comes at a very high computational cost. Therefore, the use of 200 particles is recommended in the PF approach.

Table 5. RRMSE values using different number of particles in the PF for different LMPs.

| LMPs % | RRMSE (%) | | | | |
|--------|-----------|-----------|-----------|------------|------------|
| | $k = 10$ | $k = 100$ | $k = 200$ | $k = 1000$ | $k = 2000$ |
| 1 | 72 | 66 | 64 | 61 | 59 |
| 3 | 69 | 62 | 60 | 57 | 56 |
| 5 | 66 | 59 | 56 | 53 | 52 |
| 8 | 60 | 54 | 52 | 48 | 47 |
| 10 | 56 | 50 | 48 | 46 | 44 |
| 15 | 48 | 44 | 42 | 40 | 40 |
| 20 | 44 | 41 | 40 | 38 | 36 |
| 30 | 34 | 30 | 30 | 30 | 30 |
| 40 | 22 | 22 | 22 | 22 | 22 |
| 50 | 19 | 18 | 18 | 18 | 17 |
| 60 | 16 | 15 | 15 | 14 | 14 |
| 70 | 13 | 12 | 12 | 12 | 11 |
| 80 | 11 | 9 | 9 | 9 | 9 |
| 90 | 9 | 7 | 7 | 6 | 6 |

4. Summary and Conclusions

The paper developed a nonlinear PF estimation approach to estimate the number of vehicles approaching a traffic signal based solely on CV data, with the aim of improving the estimation accuracy of linear state-of-the-art estimation approaches. This study introduced two linear approaches, KF and AKF, as benchmarks, to be compared with the proposed nonlinear PF approach. The results show that the KF produces the least error and accurately estimates the vehicle counts compared with the AKF and PF approaches. Consequently, to address the research problem appropriately, it is recommended to deploy the linear KF approach rather than the more complex AKF and PF approaches because of its simplicity and high-performance accuracy. In addition, the study investigated the sensitivity of the developed approaches to different factors, including the LMP of CVs, the initial vehicle count estimates, and the number of particles used in the PF approach. The results indicate that the estimation errors decrease as the LMP increases. Furthermore, the paper investigated the effect of the choice of the number of particles on the performance of the PF and showed that the PF estimation accuracy increases as the number of particles increases. However, this comes at the expense of significantly longer computational times. This can significantly impact the performance of the PF, requiring longer time to converge. The results demonstrate that the KF approach is the least sensitive to the initial vehicle count estimate, while the PF approach is the most sensitive to the initial vehicle count estimate and thus is the most suitable for the proposed application. Proposed future work entails integrating the KF approach with an adaptive traffic signal controller to quantify the impact of inaccuracies of the traffic stream density on the traffic signal controller performance.

Author Contributions: The work described in this article is the collaborative development of all authors. Conceptualization, M.A.A., H.M.A. and H.A.R.; methodology, M.A.A., H.M.A. and H.A.R.; software, M.A.A., H.M.A. and H.A.R.; validation, M.A.A., H.M.A. and H.A.R.; formal analysis, M.A.A., H.M.A. and H.A.R.; investigation, M.A.A., H.M.A. and H.A.R.; writing—review and editing, M.A.A., H.M.A. and H.A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research effort was funded by the Urban Mobility and Equity Center (UMEC).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feng, Y.; Head, K.L.; Khoshmaghmagham, S.; Zamanipour, M. A real-time adaptive signal control in a connected vehicle environment. *Transp. Res. Part C Emerg. Technol.* **2015**, *55*, 460–473. [\[CrossRef\]](#)

2. Abdelghaffar, H.M.; Rakha, H.A. Development and Testing of a Novel Game Theoretic De-Centralized Traffic Signal Controller. *IEEE Trans. Intell. Transp. Syst.* **2019**. [[CrossRef](#)]
3. Chen, H.; Rakha, H.A. Real-time travel time prediction using particle filtering with a non-explicit state-transition model. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 112–126. [[CrossRef](#)]
4. Vigos, G.; Papageorgiou, M.; Wang, Y. Real-time estimation of vehicle-count within signalized links. *Transp. Res. Part C Emerg. Technol.* **2008**, *16*, 18–35. [[CrossRef](#)]
5. Mihaylova, L.; Hegyi, A.; Gning, A.; Boel, R.K. Parallelized particle and Gaussian sum particle filters for large-scale freeway traffic systems. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 36–48. [[CrossRef](#)]
6. Pan, T.; Sumalee, A.; Zhong, R.X.; Indra-Payoong, N. Short-term traffic state prediction based on temporal-spatial correlation. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1242–1254. [[CrossRef](#)]
7. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Real-time Estimation of Vehicle Counts on Signalized Intersection Approaches Using Probe Vehicle Data. *IEEE Trans. Intell. Transp. Syst.* **2020**. [[CrossRef](#)]
8. Chu, L.; Oh, S.; Recker, W. Adaptive Kalman filter based freeway travel time estimation. In Proceedings of the 84th TRB Annual Meeting, Washington, DC, USA, 9–13 January 2005; pp. 1–21.
9. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Kalman filter-based vehicle count estimation approach using probe data: A multi-lane road case study. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 4374–4379. [[CrossRef](#)]
10. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Developing a Neural-Kalman Filtering Approach for Estimating Traffic Stream Density Using Probe Vehicle Data. *Sensors* **2019**, *19*, 4325, doi:10.3390/s19194325. [[CrossRef](#)]
11. Wang, Y.; Papageorgiou, M. Real-time freeway traffic state estimation based on extended Kalman filter: A general approach. *Transp. Res. Part B Methodol.* **2005**, *39*, 141–167. [[CrossRef](#)]
12. Mihaylova, L.; Boel, R. A particle filter for freeway traffic estimation. In Proceedings of the 43rd IEEE Conference on Decision and Control (CDC), Nassau, Bahamas, 14–17 December 2004; pp. 2106–2111.
13. Mihaylova, L.; Boel, R.; Hegyi, A. Freeway traffic estimation within particle filtering framework. *Automatica* **2007**, *43*, 290–300. [[CrossRef](#)]
14. Anand, R.A.; Vanajakshi, L.; Subramanian, S.C. Traffic density estimation under heterogeneous traffic conditions using data fusion. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Baden, Germany, 5–9 June 2011; pp. 31–36.
15. Abdelghaffar, H.M.; Woolsey, C.A.; Rakha, H.A. Comparison of Three Approaches to Atmospheric Source Localization. *J. Aerosp. Inf. Syst.* **2017**, *14*, 40–52. [[CrossRef](#)]
16. Hegyi, A.; Girimonte, D.; Babuska, R.; De Schutter, B. A comparison of filter configurations for freeway traffic state estimation. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 17–20 September 2006; pp. 1029–1034.
17. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In Proceedings of Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 21–25 April 1997; pp. 182–193.
18. Zhai, Y.; Yeary, M. Implementing particle filters with Metropolis-Hastings algorithms. In Proceedings of the Region 5 Conference: Annual Technical and Leadership Workshop, Norman, OK, USA, 2 April 2004; pp. 149–152.
19. Wright, M.; Horowitz, R. Fusing loop and GPS probe measurements to estimate freeway density. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3577–3590. [[CrossRef](#)]
20. Chen, H.; Rakha, H.A.; Sadek, S. Real-time freeway traffic state prediction: A particle filter approach. In Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 626–631.
21. Liu, J.S.; Chen, R. Sequential Monte Carlo methods for dynamic systems. *J. Am. Stat. Assoc.* **1998**, *93*, 1032–1044. [[CrossRef](#)]
22. Ristic, B.; Arulampalam, S.; Gordon, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*; Artech House: Norwood, MA, USA, 2003.
23. Roess, R.P.; Prassas, E.S.; McShane, W.R. *Traffic Engineering*; Pearson Education: Hoboken, NJ, USA, 2004.
24. Abdelghaffar, H.M.; Elouni, M.; Bichiou, Y.; Rakha, H.A. Development of a Connected Vehicle Dynamic Freeway Variable Speed Controller. *IEEE Access* **2020**, *8*, 99219–99226. [[CrossRef](#)]
25. Cronje, W. Analysis of existing formulas for delay, overflow, and stops. In Proceedings of the 62nd Annual Meeting of the Transportation Research Board, Washington, DC, USA, 17–21 January 1983; pp. 89–93.

26. Calle-Laguna, A.J.; Du, J.; Rakha, H.A. Computing optimum traffic signal cycle length considering vehicle delay and fuel consumption. *Transp. Res. Interdisciplin. Perspect.* **2019**, *3*, 100021. [CrossRef]
27. Balke, K.N.; Charara, H.A.; Parker, R. *Development of a Traffic Signal Performance Measurement System (TSPMS)*; Technical Report; Texas A and M University System: College Station, TX, USA, May 2005.
28. Roess, R.P.; Prassas, E. S.; McShane, W. R. *Traffic Engineering*, 15th ed.; Pearson Education: Hoboken, NJ, USA, 2019.
29. Gazis, D.C. Optimum control of a system of oversaturated intersections. *Oper. Res.* **1964**, *12*, 815–831. [CrossRef]
30. Abdelghaffar, H.M.; Rakha, H.A. A novel decentralized game-theoretic adaptive traffic signal controller: Large-scale testing. *Sensors* **2019**, *19*, 2282. [CrossRef]
31. Cheung, S.Y.; Coleri, S.; Dundar, B.; Ganesh, S.; Tan, C.W.; Varaiya, P. Traffic measurement and vehicle classification with single magnetic sensor. *Transp. Res. Rec.* **2005**, *1917*, 173–181. [CrossRef]
32. Qian, G.; Lee, J.; Chung, E. Algorithm for queue estimation with loop detector of time occupancy in off-ramps on signalized motorways. *Transp. Res. Rec.* **2012**, *2278*, 50–56. [CrossRef]
33. Gerlough, D.L.; Huber, M.J. Traffic flow theory. In Proceedings of the Annual Meeting of the Transportation Research Board, Washington, DC, USA, 19–23 January 1976.
34. Kurkjian, A.; Gershwin, S.B.; Houpt, P.K.; Willsky, A.S.; Chow, E.; Greene, C. Estimation of roadway traffic density on freeways using presence detector data. *Transp. Sci.* **1980**, *14*, 232–261. [CrossRef]
35. Bhouri, N.; Salem, H.H.; Papageorgiou, M.; Blossville, J.M. Estimation of traffic density on motorways. In Proceedings of the IFAC/IFIP/IFORS International Symposium (AIPAC'89), Nancy, France, 3–5 July 1989; pp. 579–583.
36. Mimbela, L.E.Y.; Klein, L.A. Summary of Vehicle Detection and Surveillance Technologies Used in Intelligent Transportation Systems, Technical Report; 2000. Available online: <https://www.fhwa.dot.gov/ohim/tvtw/vdstits.pdf> (accessed on 4 June 2020).
37. Lee, J.; Hernandez, M.; Stoschek, A. Camera System for a Vehicle and Method for Controlling a Camera System. US Patent 8,218,007, 10 July 2012.
38. Anand, A.; Ramadurai, G.; Vanajakshi, L. Data fusion-based traffic density estimation and prediction. *Int. J. Intell. Transp. Syst. Res.* **2014**, *18*, 367–378. [CrossRef]
39. van Erp, P.B.; Knoop, V.L.; Hoogendoorn, S.P. Estimating the vehicle accumulation: Data-fusion of loop-detector flow and floating car speed data. In Proceedings of the 97th TRB Annual Meeting, Washington, DC, USA, 7–11 January 2018.
40. Cheng, P.; Qiu, Z.; Ran, B. Particle filter based traffic state estimation using cell phone network data. In Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 17–20 September 2006; pp. 1047–1052.
41. Li, T.; Sattar, T.P.; Sun, S. Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters. *Signal Process.* **2012**, *92*, 1637–1645. [CrossRef]
42. Maybeck, P.S. The Kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*; Cox, I.J., Wilfong, G.T., Eds.; Springer: New York, NY, USA, 1990; pp. 194–204.
43. Rakha, H.A. *INTEGRATION Release 2.40 for Windows: User's Guide-Volume II: Advanced Model Features*; Technical report; Center for Sustainable Mobility, Virginia Tech Transportation Institute: Blacksburg, VA, USA, June 2020. [CrossRef]
44. Aljamal, M.A.; Rakha, H.A.; Du, J.; El-Shawarby, I. Comparison of microscopic and mesoscopic traffic modeling tools for evacuation analysis. In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2321–2326.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Empirical Study of Effect of Dynamic Travel Time Information on Driver Route Choice Behavior

Jinghui Wang and Hesham Rakha *

Center for Sustainable Mobility, Virginia Tech Transportation Institute, 3500 Transportation Research Drive, Blacksburg, VA 24061, USA; jwang86@vt.edu

* Correspondence: hrakha@vt.edu

Received: 22 May 2020; Accepted: 5 June 2020; Published: 8 June 2020

Abstract: The objective of this paper is to study the effect of travel time information on day-to-day driver route choice behavior. A real-world experimental study is designed to have participants repeatedly choose between two alternative routes for five origin-destination pairs over multiple days after providing them with dynamically updated travel time information (average travel time and travel time variability). The results demonstrate that historical travel time information enhances behavioral rationality by 10% on average and reduces inertial tendencies to increase risk seeking in the gain domain. Furthermore, expected travel time information is demonstrated to be more effective than travel time variability information in enhancing rational behavior when drivers have limited experiences. After drivers gain sufficient knowledge of routes, however, the difference in behavior associated with the two information types becomes insignificant. The results also demonstrate that, when drivers lack experience, the faster less reliable route is more attractive than the slower more reliable route. However, with cumulative experiences, drivers become more willing to take the more reliable route given that they are reluctant to become risk seekers once experience is gained. Furthermore, the effect of information on driver behavior differs significantly by participant and trip, which is, to a large extent, dependent on personal traits and trip characteristics.

Keywords: route choice behavior; real world experiment; Intelligent Transportation Systems (ITS); advanced traveler information systems (ATIS)

1. Introduction

Advanced traveler information systems (ATISs), which are an integral component of Intelligent Transportation Systems (ITSs), are designed to provide real-time information that enables drivers to choose rationally from among alternative routes. The effectiveness of ATIS is dependent on the drivers response to received information. Incorporating information into the modeling practice may enhance the accuracy of route choice models by adding realistic behavioral mechanisms and thus improve the effectiveness of ITSs. Accordingly, it is essential to capture the behavioral generalization of informed drivers in order to enhance ATIS design.

From a modeling perspective, traditional transportation research attempts to replicate driver route choice behavior assuming that individuals are capable of accurately perceiving route performance and attempt to maximize their expected utility. Mathematically, such assumptions make it cost-effective and technically simpler to model traveler behavior. Most attempts at route choice modeling are discrete choice models that are econometrically derived from random utility theory. Since the 1970s, transportation researchers have studied the decisions associated with route choice modeling. In the past forty years, innovations in discrete choice models have progressed in three stages, namely: Multinomial Logit Modeling [1], Nested Logit Modeling [2] and Mixed Logit Modeling (Ben-Akiva et al. unpublished manuscript, 1996). Each enhancement attempted to direct the logit

model towards more flexible model structures. Despite previous achievements, the ability of these models to capture realistic route choice behavior has been increasingly challenged due to insights from the psychology field. Human choice behavior often produces *imperfect* dynamical systems that can be controlled using techniques described in [3]. Through a range of empirical research, drivers were detected to be not omniscient, as expected in traditional models, in precisely perceiving the actual route performance. Bounded rationality was initially introduced by Simon [4] to explicitly account for the fact that human beings are incapable of identifying the best route among multiple alternatives due to limitations in knowledge, cognition and information acquisition. Tawfik and Rakha [5] verified Simon's theory using a real world experiment, demonstrating that drivers generally only had a 50% accuracy in perceiving route information (e.g., travel time, travel distance, speed). Even though travelers occasionally have correct perception of route performance, they may not be willing to switch to the perceived better route; rather, they stick to the habitual choice until its performance is not satisfying. In other words, travelers are not necessarily utility maximizers [6]. Satisfying psychology triggers individuals' behavioral mechanisms in seeking a satisfactory solution instead of the optimal one. Irrational behaviors deviate travelers from the best route and are not easily predictable by traditional models due to limitations in model assumptions.

Route information provides an explicit description of the actual performance of the choice sets, which has the potential to improve travelers knowledge and direct them towards the objectively optimal decision. Accordingly, route information is expected to facilitate travelers to make more logical choices (choose faster routes in this study). The accuracy of traditional discrete choice models may probably be improved by integrating information effects into the modeling practice. An explicit generalization of the effect of information on route choice behavior is thus studied.

The proposed research attempts to provide valuable insights in addressing a number of important questions, namely: does route information enable drivers to behave more rationally? How does the information affect behavioral mechanisms from person to person as well as from trip to trip? What is the difference in behavioral effects between information types? This study is a follow-up experiment of [5]. The results of the two experiments are compared and provide significant implications to the behavioral effect of real-time information. The major contribution of this study is to design a real world route choice experiment and study realistic route choice behavior of informed drivers and their day-to-day behavioral variations. This study differs from most of the studies in the literature that have investigated driver route choice behavior in a hypothetical environment such as simulation and questionnaire that is unable to completely reflect reality. The paper also comprehensively presents the heterogeneity of drivers' responses to the provided route information, considering the diversities in driver's age, gender, and personal traits, trip characteristics, and temporal variation. The findings of this study are critical and insightful to the modeling of route choice behavior and personalized ATIS design.

2. Literature Review

Empirical research found that the factors considered by travelers in making route choice decisions were not unitary [6]. Numerous attributes were found to be important considerations, including travel time, trip distance, average speed, and the number of traffic signals along the route. Nonetheless, previous attempts at identifying the attributions of route choice identify travel time as the most important factor even though travelers may also consider other factors. In accordance with Tawfik and Rakha's study, 70% of drivers' route choices was successfully explained by travel time followed by average speed and distance traveled [5,7]. Consequently, travel time information is provided to the test participants in this study.

As captured in the "hot stove" effect [8], individuals were not inclined to select options associated with high variability, although these might actually provide larger benefits. Considering uncertainty, people do not have perfect knowledge of the gains that could be accrued and the loss associated with risking changing habitual choices. Prospect Theory [9] explicitly and thoroughly describes this

psychological behavior that risk-seeking behavior would likely exhibit in the loss domain rather than in the gain domain. In relation to route choice, Katsikopoulos et al. [10] verified the results of Prospect Theory through a simulated experiment in which participants were provided with the information of travel time variability, indicating that risk aversion emerged in the gain domain (alternative route is faster but riskier) while risk seeking emerged in the loss domain (alternative route is slower but riskier). Accordingly, drivers repeatedly make illogical choices due to the risk aversion in the gain domain. Information is expected to reduce the uncertainty and enhance rational behavior partially by leading travelers to risk seeking in the gain domain. Katsikopoulos et al. [10,11] revealed that the provided information supported choice rationality and reduced inertia.

The effect of travel time information on route choice behavior has been incrementally studied both from a theoretical and practical standpoint. Early studies, such as Lida et al. [12] and Yang et al. [13], pioneered the investigation of the information effects on drivers route choice behavior, both of which conducted studies in the simulation environment, with the number of participants 40 and 20, respectively. Ben et al. [14] thoroughly investigated the combined effects of information and driving experience on route choice behavior using a simulated experiment in which a total of 49 participants were recruited. The results provided evidence to suggest that the expected benefit of information is achieved only if drivers lacked long-term experience. Based on this study, a discrete choice model with Mixed Logit specifications was developed to accurately describe the respondents' learning process under the provision of real-time information [15]. Further, Ben et al. [15] also demonstrated that information provided on average travel time resulted in different responses compared to information on travel time variability, which remains to be verified. Using a simulation- and a stated preference-based approach, numerous attempts were made to econometrically address the various behavioral mechanisms of drivers' route choice with real-time information. The studied behavioral mechanisms involved logical choice [14,15], inertia choice [11,16], switching behavior [17–19], habit and learning [20,21], and others [22–27]. Specifically, Karthik et al. [16] designed an inertia behavior simulation study and demonstrated that user experiences decreased inertia behavior in day-to-day variation. The travel time information was demonstrated by many studies to effectively move route choice towards rationality ([14,15,17,19,25,26,28–32]), however, the effect of information strongly depends on other factors, such as personal traits, trip characteristics, and other decision considerations. From the personal trait perspective, Jou et al. [17] concluded that elderly travelers would be less likely to switch due to the habitual and risk-averse effects, and male travelers would be more likely to switch to the best route. Also, trip characteristics and traveler preferences were proved by Polydoropoulou et al. [18] to significantly affect route switching and compliance with information. In summary, to the authors' best of knowledge, existing studies have typically lacked realism (either based on simulation or stated preferences approaches) and have not characterized the effect of information details of trip characteristics, such as directness of the route, number of intersections, conflicts with non-motorized traffic on driver route choice behavior. This study attempts to address this void.

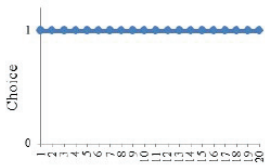
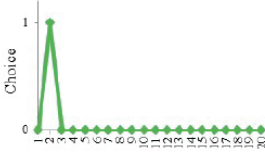
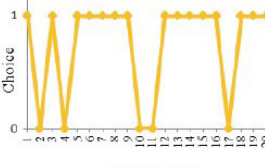
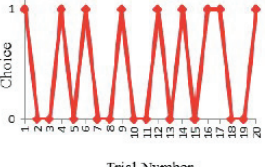
Although previous attempts provided econometric and empirical generalizations, most were based on simulation and stated preference approaches. In the simulator surroundings, however, respondents make decisions in a digital and virtual environment. Stated preference is an investigative approach in which respondents are given questionnaires to make choices hypothetically. Both approaches are performed under fictitious conditions and may not accurately capture actual choice behavior. Consequently, an in-field case study is needed to address the driver route choice behavior. To the author's best of knowledge, this study, as a follow-up test of Tawfik and Rakha's experiment (in which information was not available), is the first attempt at addressing this need using dynamic travel time information, which differs from the previous real-world experiments (e.g., [33,34]) that conducted experiments for a short time period (e.g., several days) and did not capture the day-to-day variation of route choice behavior using the learning mechanism that accounts for information effects. For example, Papinski et al. [33] had 31 participants involved in a real-world

route choice study with GPS as the data collection tool. The study provided valuable insights into the use of GPS trajectory data for route choice analysis, yet was only conducted for two days, thus not capturing day-to-day variations and dynamics of learning behavior.

Drivers’ responses to information may differ based on personal characteristics, demographics, preferences and choice situations [35,36]. Nonetheless, few studies so far have attempted to quantitatively investigate such discrepancy. Tawfik et al. [21] developed a latent class choice model by classifying personal traits and choice situations into four behavioral groups as illustrated in Table 1. The results demonstrated that the model outperformed traditional hierarchical models in predicting realistic behavior. However, Tawfik et al.’s study did not incorporate the effect of information in the modeling practice. Accordingly, this study attempts to investigate the information effect considering different participants and choice situation characteristics in order to capture preliminary insights for modeling in the future horizon.

In general, given the incomplete picture of the behavioral aspects of route-choice decision making, more attempts are justified. The proposed research is thus initiated by a real world case study to provide a better understanding of underlying behavioral effects of travel time information on route choice decisions.

Table 1. Four identified behavioral driver types [21].

| Behavior Type | Typical Behavior | Type Description |
|---------------|--|--|
| 1 |  <p>A line graph with 'Choice' on the y-axis (0 to 1) and 'Trial Number' on the x-axis (1 to 20). The data points are all at the value of 1, forming a horizontal line.</p> | A driver starts by arbitrarily selecting a route, is apparently satisfied with the experience, and continues making the same choice for the entire 20 trials. |
| 2 |  <p>A line graph with 'Choice' on the y-axis (0 to 1) and 'Trial Number' on the x-axis (1 to 20). The choice is 1 at trial 1 and 0 for all subsequent trials from 2 to 20.</p> | A driver starts by arbitrarily selecting a route, is apparently not satisfied with the experience, tries the other route, and decides that the first route was better. The driver makes a choice after trying both routes and does not change afterward. |
| 3 |  <p>A line graph with 'Choice' on the y-axis (0 to 1) and 'Trial Number' on the x-axis (1 to 20). The choice alternates between 1 and 0 in a regular pattern: 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0.</p> | A driver switches between the two alternative routes over the duration of the experiment. The driver, however, drives on one route more than the other route. This reflects his/her preference for the selected route. |
| 4 |  <p>A line graph with 'Choice' on the y-axis (0 to 1) and 'Trial Number' on the x-axis (1 to 20). The choice alternates between 1 and 0 in a regular pattern: 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0.</p> | A driver switches between the two alternative routes over the duration of the experiment. The driver drives both routes with approximately equal percentages. This reflects a lack of preference towards any of the alternatives. |

3. Experimental Design

As aforementioned, Tawfik et al. identified four route choice patterns observed in a real world experiment. This experiment attempts to quantify the influence of route information on traveler route choice behavior by comparing the choice patterns between Tawfik et al.'s experiment and the experiment conducted in this study. Occasionally, drivers prefer a route they frequently choose instead of switching to the actually faster route; or may deviate from the habitual route to the alternative route, which is on average worse, only because the performance of the usually-taken route becomes bad on a random day. These irrational behaviors may probably be caused by a lack of precise information. The study attempts to address a number of questions: Will travel time information make drivers behave more rationally? Will the effect of information be different among individuals? What type of information will be most effective?

A total of 20 participants were recruited within two age groups including 18-33 and 55-75. These two age groups were selected because the authors wanted to investigate the impact of drivers' age on the information effectiveness in changing choice behavior. The big difference of drivers' age in the two age groups may more easily distinguish the difference of information effect attributed to drivers' age. Each of them was required to accomplish three sectors of the experiment: a pre-run questionnaire, on-road test and a post-run questionnaire. The pre-run questionnaire was conducted before the beginning of the on-road test, which gathered the participants' demographics, driving experiences, preferences, habits, information usage and the perception of route performance. Noticeably, each participant was demonstrated to have little knowledge of the route performance according to the results of pre-run questionnaire. The on-road test was conducted around the areas in Blacksburg and Christiansburg, VA for the morning, noon and evening peak from October 2013 to April 2014. The participants were asked to drive as if (When drivers were doing the test, they were asked to drive from one predefined origin to the destination during every trip, and they actually did not commute during the test. However, the researchers wanted to emulate the trip as a commute trip on which travel time may probably be the first consideration by the drivers. So "as if" here means that drivers were asked to behave like in a commute.) they were commuting in order to ensure that travel time was an important consideration when they were to make choices. Each participant was asked to drive 11 trials, 5 of which provided participants with strict information (average travel time) and 5 provided with range information (travel time variability). The last one trial was not provided with any information, aiming to see how well information impacted drivers. It should be noted that the information was provided one time with average travel time and one time with travel time variability in order to eliminate the bias on each of the information types. The average travel time information provided to each trial was estimated by averaging the experienced travel time of three previous trials (The experienced travel time was recorded by GPS during the testing; three previous trials were selected to be averaged because the trials before has little impact on the decision based on the literature [5]; information used for the first trial was obtained from the experiment in [5].) and travel time variability was estimated using the average value and standard deviation ($average\ travel\ time \pm 2 * standard\ deviation$), so that the information could be dynamically updated each day to enhance the reliability estimate. It is worth noting that the provided travel time information was collected using GPS during the real-world experiment rather than from on-road or in-vehicle sensors, but the experimental design methodology of this study is also applicable to sensor-based information. For each trial, there were five O-D trips, each of which had two alternative routes, one route was on average faster in travel time than the other. The characteristics of each route were specified in Table 2. The participants' task was to repeatedly make choices between the two alternatives on each trip. Statistically, 55 choice observations were collected for each participant, 100 observations by each trial and 220 on each trip. Upon the completion of 11 trials of the on-road test, the post-run questionnaire was thereafter conducted, whereby the participants were asked whether the provided information was beneficial. The accuracy of travel time perception would be compared between the two questionnaires in order to have a knowledge of whether the participants' perception was improved as a result of providing them with information.

The logical choice rate—the proportion of times in which the faster route is chosen as a function of time (trial number), participant and trip, respectively—was selected as the indicator of the positive role of information in facilitating rational behavior. The inertial choice rate—the proportion of participants remaining on their habitual but slower route—served to evaluate whether the information contributed to enhancing participant attitudes of risk seeking in the gain domain. The on-road data collected by Tawfik was applied to estimate the choice rates specified as “without information” group. Tawfik’s experiment was conducted on the same trips in Blacksburg and Christiansburg in 2012, which was also a day-to-day commuting test in which participants were asked to repeatedly make choices between the two alternative routes on each trip. The difference between the two experiments was that the proposed study provided participants with travel time information. For more details of Tawfik’s study, see [5].

Table 2. Route characteristics of each O-D trip.

| Trip No. | Route No. | Ave. Travel Time | No. of Intersections | | No. of Left Turns | Route Description |
|----------|-----------|------------------|----------------------|--------------|-------------------|--|
| | | | Signalized | Unsignalized | | |
| 1 | 1 | 9.2 | 10 | 3 | 3 | Mostly a high speed (65 mile/hour) freeway |
| | 2 | 9.3 | 5 | 4 | 4 | High speed (45 mile/hour) urban highway |
| 2 | 3 | 15.8 | 5 | 2 | 3 | Mostly a shorter, low speed (30 mile/hour) back road with a lot of curves |
| | 4 | 18.2 | 2 | 2 | 2 | Mostly a longer, high speed (55 mile/hour) rural highway |
| 3 | 5 | 8.6 | 5 | 3 | 3 | A longer high speed (65 mile/hour) freeway followed by a low speed (25 mile/hour) urban road |
| | 6 | 9.4 | 8 | 3 | 2 | A shorter urban route (40 and 35 mile/hour) |
| 4 | 7 | 10.4 | 5 | 3 | 4 | A short urban route that passes through campus (25 and 35 mile/hour) |
| | 8 | 10.3 | 6 | 2 | 2 | Primarily a long high speed (65 mile/hour) freeway and low speed (25 mile/hour) urban roads |
| 5 | 9 | 10.5 | 8 | 4 | 4 | A long urban road that passes through town (35 mile/hour) |
| | 10 | 8.5 | 3 | 1 | 3 | A short low speed (25 and 35 mile/hour) rural road that passes by a small airport, and more direct |

4. Results Analysis

By comparing the perceived travel time of the pre-run questionnaire to the actual travel time collected during the on-road tests, it was demonstrated that the accuracy of participants’ perception of travel time ranged from 5% to 55% for all five trips, with an average accuracy of only 38%. Consequently, it would be safe to conclude that the participants had limited knowledge of the route performance prior to the start of the experiment. Based on the results of participants’ perception in the post-run questionnaire, the average accuracy increased from 38% to 62% with an increase of 24%. Consequently, it would be interesting to see whether participants behave more rationally with higher perception accuracy.

Figure 1 presents the proportions of logical- and inertial- choices as a function of time, identified as trial number. As expected, the logical choice rates are on average around 10% higher in the “with-information” group than “without-information” group, especially for the first two trials in which the enhancement is up to 15%. This demonstrates that the positive effect of information becomes more evident when travelers have limited knowledge of route performance. Although there are some oscillations at some of the trials, in general, the logical rates between the two groups are getting closer from the beginning to the end. The inertial choice rates are basically lower with the provision of information, implying that it is more likely for travelers to risk switching to the faster route when they

are informed. However, regardless of being informed or not being informed, the inertial behavior is not reduced in day-to-day variation, which is different from the results in the simulation study [16]. This may be attributed to the habit or other decision considerations.

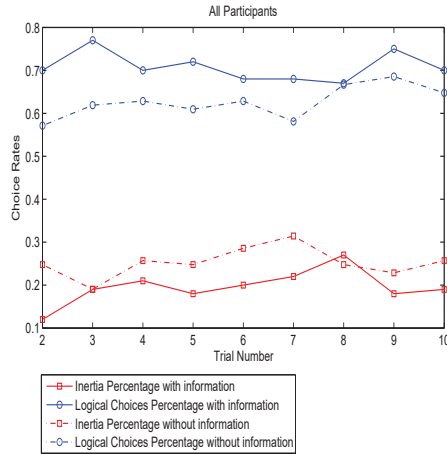


Figure 1. Logical- and inertial- choice rates over trials.

In reality, the behavioral effect of information varies from person to person. One may probably have more confidence in his/her experiences than the acquired information; or travel time is not his/her top consideration. Accordingly, the insights gained from previous analyses are needed. Nine of the participants in this study attended Tawfik and Rakha's experiment. The choice results of these participants were specifically compared between the two experiments in order to see how the effect of information differentiated individually and how well they learned from the information. Figure 2 compares the behavioral types (introduced in Table 1, which was proposed by [21]) for each of the nine participants between with- and without-information. Only 10 trials were compared because the participants were informed for 10 trials only. The degree of the fluctuation of each line gives an explicit generalization of participants' behavioral aggressiveness. The more fluctuated in the lines, the more aggressively the participants behave. In general, the information significantly changes behavioral types either from risk-seeking to risk-aversion or vice versa. Some of the participants exhibited a high preference for one route when information was not provided and switched frequently when they were informed; whereas some switched more without information and maintained a single route when informed. Overall, the effect of information significantly differs at an individual level.

Figure 3 summarizes the behavioral tendency of participants. According to Figure 3a, participants 1, 2, 3, 5, 8 basically moved their choices towards rationality with the assistance of information, whereas participants 4, 6, 7, 9 behaved more irrationally when they were informed. In Figure 3b, participants 6, 7, 9 instead have higher inertial rates with the provision of information, implying that they behaved even more risk averse when they were provided with information.

Based on the results of the post-run questionnaire, participants 6, 7 and 9 mentioned that travel time information had little impact on their route choices. Specifically, participant 6 preferred rural roads due to his preference on route scenery, although travel time was important to him as well. Participant 7 held the point that, instead of travel time, the number of intersections was the overriding factor she considered for route choice decisions. Participant 9 preferred to stick to her current route without any route-switching, which is the first type of the typical behavior shown in Table 1. Noticeably, participant 4 had both logical and inertial rates decreased with the provision of travel time information. That was because travel time was not the only consideration to this participant. Based on the results of

the questionnaire, “avoid traffic lights” was the other equally important factor to him, which highly impacted his choice behavior. Occasionally, participant 4 switched to the slower route instead in order to avoid traffic lights even though he was informed the alternative route was better in terms of travel time, which increased the proportion of compromising behavior (the other type of illogical choice other than inertial choice) and decreased the logical choice rate. In general, travel time may have little effectiveness in enabling drivers to behave logically when drivers do not take travel time as their foremost factor in planning their routes. Additionally, participants 4, 6, 7, 9 are all senior persons from the age group of 55–75 year old. This implies that elder drivers are preferable to make choices based on their preferences or habits rather than received information, which confirms the results of [17].

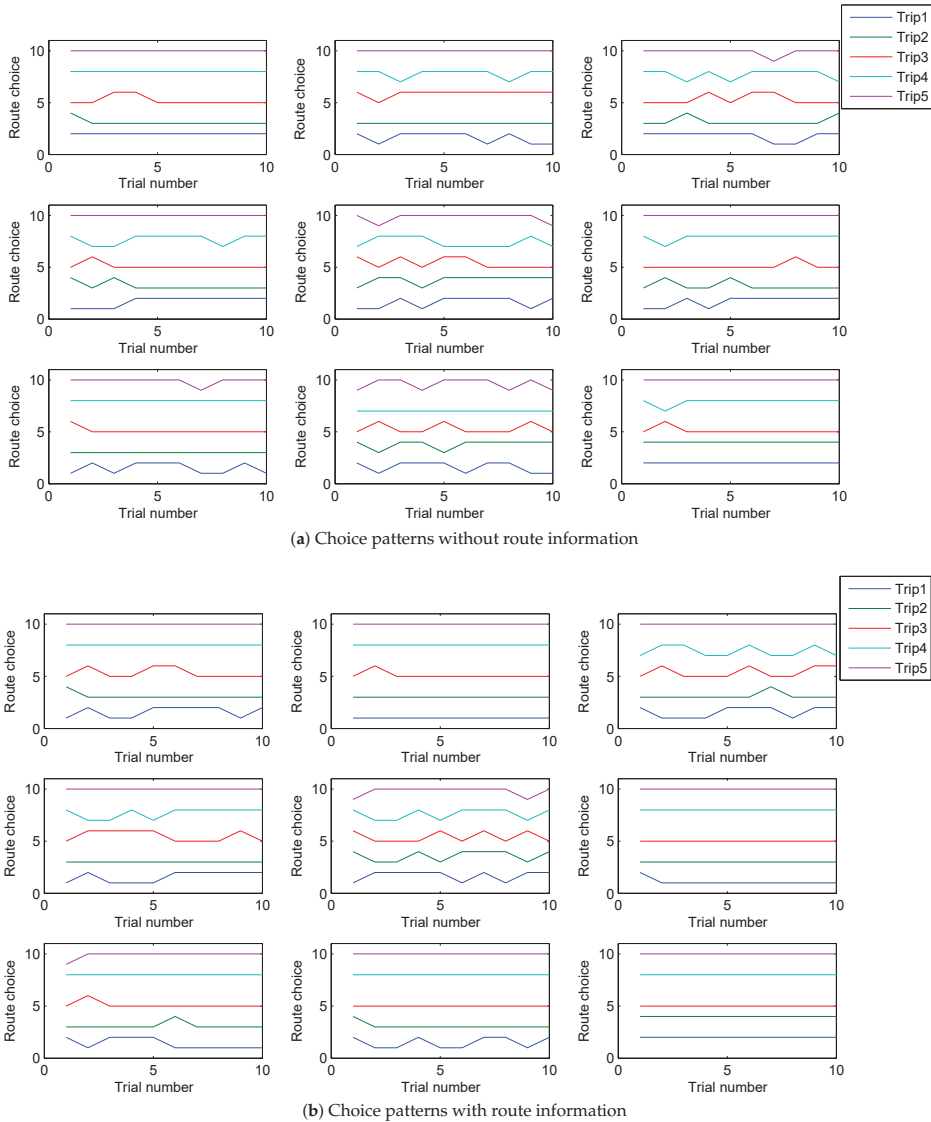


Figure 2. Participants choice patterns without vs. with route information.

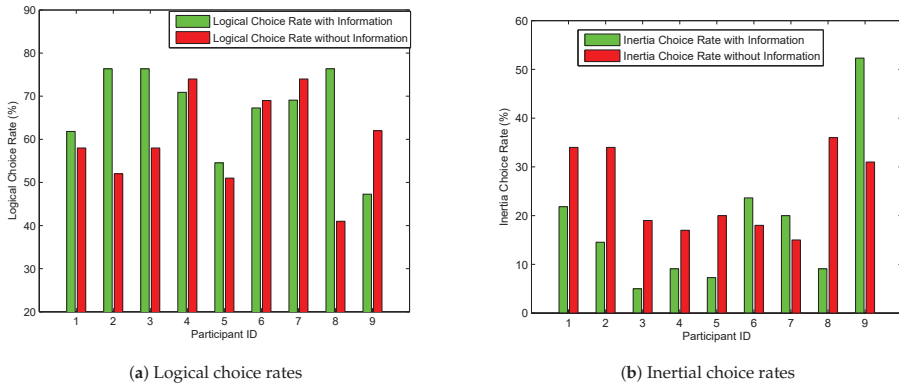


Figure 3. Logical- and inertial- choice rates over participants.

In addition to individual traits, trip characteristics may also affect the positive role of information. To study such effects, the choice rates were aggregated by trips. As illustrated in Figure 4, information enhances behavioral rationality only for the first three trips. For trip 4, logical rates decrease while inertial rates increase when information is provided. On trip 5, the choice rates do not change significantly between with- and without- information. According to the route characteristics addressed in Table 2, route 7 and route 8 (on trip 4) are almost identical in travel time, whereas many participants pointed out that they were reluctant to take route 7 even though it occasionally took less travel time since they did not want to risk being caught on campus by pedestrian flows. The provided information was considered to be less reliable for this trip. Interestingly, travel time is very close as well between the two routes on trip 1; however, the effect of information appears to be very positive. That is because there is no distinct advantage for one route over the other on this trip. Although route 1 is on a highway system with a 20 km/h higher speed limit than route 2, there are five more signalized intersections on it. The provided travel time information for this trip was considered reliable by participants. For trip 5, route 10 distinctively outperforms route 9 in terms of travel time, directness, less traffic and fewer intersections. The authors of [5] clearly indicated that drivers were able to precisely perceive the route performance and to make correct decisions on this trip without any assistance of information. Overall, information provides little benefit if one route visibly outperforms the other.

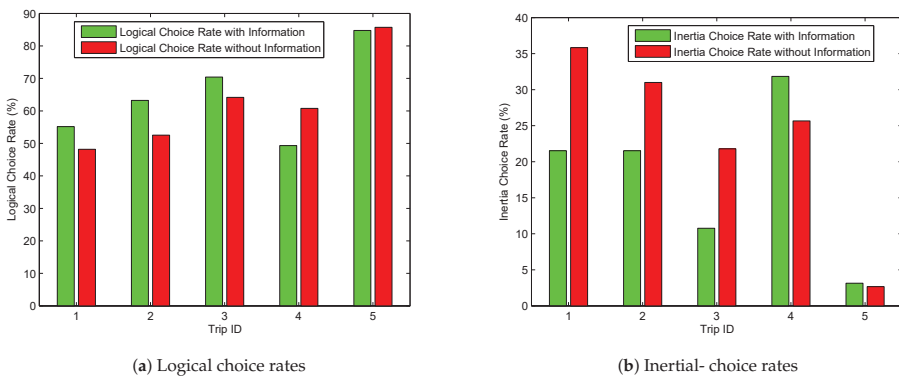


Figure 4. Logical- and inertial- choice rates over trips.

Figures 5 and 6 provide a broad view of the effect of different information types on route choice behavior. In Figure 5, the comparative analysis was performed between strict information (average travel time) and range information (variability). According to Figure 5a, strict information results in higher logical rates with lower inertial rates for the first trial, demonstrating that strict information is more effective than range information when drivers lack experience. For the following trials, however, there is no significant distinctiveness between the two scenarios. This may be attributed to the fact that the effect of information type tends to be identical after drivers gain experience. As illustrated in Figure 5b, strict information results in higher logical rates and lower inertial rates on average. Nonetheless, to some of the participants, range information performs better, implying that the responses to different information types, to a large extent, are dependent on individual traits, although strict information overall performs better in this study.

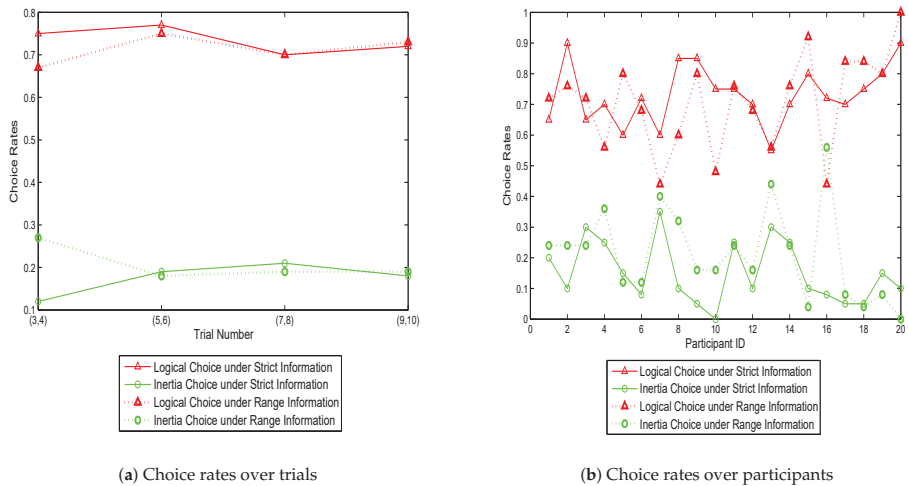


Figure 5. Choice rates with strict information vs. with range information.

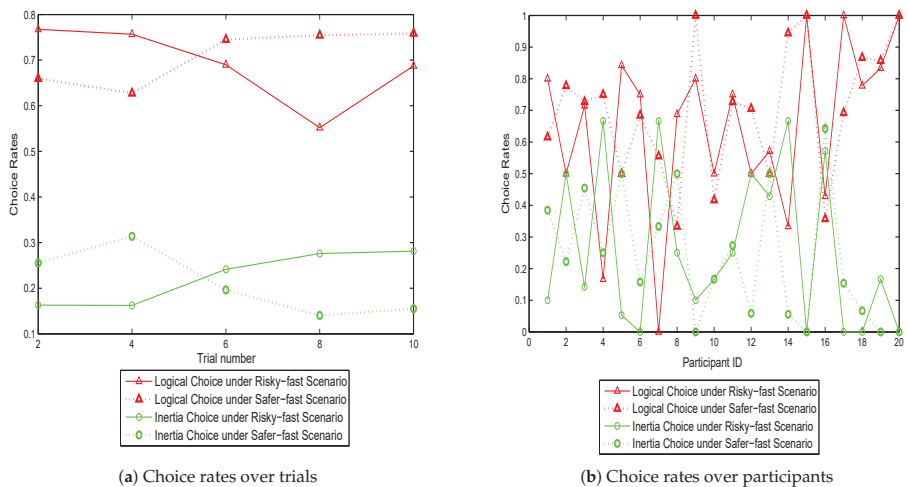


Figure 6. Choice rates with risky-fast scenario vs. safer-fast scenario.

Figure 6 presents the effect of different range information scenarios. As illustrated in Figure 6a, “Risky-fast” scenario refers to the faster route (lower average travel time) with higher variability while “safer-fast” represents the faster route with lower variability. Interestingly, the risky-fast scenario appears to have higher logical rates and lower inertial rates in the first two trials, whereas the positive effect decreases in the following three trials. This implies that, when drivers have limited knowledge of route performance, the faster route with high variability is more attractive and subject to make drivers take risk in the gain domain. Once drivers gather experience, however, they are reluctant to risk seeking in the gain domain under higher uncertainty; instead, the safer-fast route becomes preferable. This confirms the result of [10,11,14,15]. Figure 6b demonstrates that there is no consensus between participants on which scenario is more effective. Some of the participants have higher logical rates and lower inertial rates for the risky-fast scenario while some exhibit the opposite pattern.

5. Conclusions

This study empirically investigated the effect of dynamic travel time information on day-to-day commuter route choice behavior by designing and running a real world experiment on 20 participants over 11 weekdays for 5 O-D pairs considering two routes for each O-D pair. Consequently, the test entailed a total of 1100 route choice decisions ($20 \times 11 \times 5$). The experiment confirms some of the results obtained from previous simulation studies, demonstrating that, in general, real-time information significantly enhances behavioral rationality especially when drivers lack long-term experience. Simultaneously, inertial choice rates decrease with information provision, demonstrating that drivers are more willing to risk switching to faster routes when they have more information about these routes. Nonetheless, the positive role of information is, to a large extent, dependent upon the individual’s age, preferences, and route characteristics. The results demonstrate that travel time information may not have positive impacts on driver route choice behavior if they value other factors in making their decisions, such as route scenery, habit, number of intersections and traffic signals. The results also reveal that the effect of information on driver behavior is less evident for elder drivers, which is consistent with [17]. In addition to personal traits, route characteristics are found to be another important factor influencing the effectiveness of information. Specifically, information may not add value if one route is significantly better than the other given that drivers would be able to identify the optimum route on their own through their experiences.

The effect of the type of route information provided to the travelers on their route choice behavior was also studied. The conclusions are consistent with the results of simulation studies, demonstrating that, when drivers have limited experiences, information on expected travel times is in general more effective than information on travel time variability in enhancing rational behavior. After drivers gain sufficient knowledge of the alternative routes, however, the benefit of providing strict information appears to diminish. The results also demonstrate that drivers prefer to take the faster less reliable route as opposed to the slower more reliable route when they lack historical experience. However, as drivers accumulate experience, they become more willing to take the more reliable route, demonstrating that they become less risk seeking in the gain domain at higher uncertainty once experience is gained. In addition, the effect of information types on route choice behavior significantly differs from person to person. Which type of information is most effective to what group of travelers remains to be investigated in future research.

The experiment also demonstrates that, regardless of being informed or not being informed, the drivers’ inertial behavior does not reduce in day-to-day variation, which is different from the results obtained in an earlier simulation study [16]. This may be attributed to habitual behavior or the fact that more decision considerations are accounted for in actual driving.

Finally, it should be noted that given the small participant sample size, these conclusions serve as a first attempt at understanding driver route choice behavior empirically. Further research is needed to validate these findings on a bigger sample of drivers and for different confounding factors.

Author Contributions: The authors confirm contribution to the paper as follows; study conception and design: H.R. and J.W.; data collection: J.W.; analysis and interpretation of results: J.W. and H.R.; draft manuscript preparation: J.W. and H.R. All authors reviewed the results and approved the final version of the manuscript.

Funding: This effort was jointly funded by the Mid-Atlantic University Transportation Center (MAUTC), the University Mobility and Equity Center (UMEC), and through the Office of Energy Efficiency and Renewable Energy (EERE), Vehicle Technologies Office, Energy Efficient Mobility Systems Program under award number DE-EE0008209.

Acknowledgments: The authors acknowledge all personnel who assisted with the data collection.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Daganzo, C.F.; Sheffi, Y. On stochastic models of traffic assignment. *Transp. Sci.* **1977**, *11*, 253–274. [[CrossRef](#)]
2. Ben-Akiva, M.E.; Lerman, S.R. *Discrete Choice Analysis: Theory and Application to Travel Demand*; MIT press: Cambridge, MA, USA, 1985.
3. Bucolo, M.; Buscarino, A.; Famoso, C.; Fortuna, L.; Frasca, M. Control of imperfect dynamical systems. *Nonlinear Dyn.* **2019**, *98*, 2989–2999. [[CrossRef](#)]
4. Simon, H.A. *Models of Bounded Rationality: Empirically Grounded Economic Reason*; MIT press: Cambridge, MA, USA, 1982.
5. Tawfik, A.M.; Rakha, H. A real-world route choice experiment to investigate drivers perceptions and choices. In Proceedings of the Transportation Research Board 91st Annual Meeting, Washington, DC, USA, 22–26 January 2012; pp. 12–3927.
6. Vreeswijk, J.; Thomas, T.; van Berkum, E.; van Arem, B. Drivers' Perception of Route Alternatives as Indicator for the Indifference Band. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2383*, 10–17. [[CrossRef](#)]
7. Tawfik, A.M.; Rakha, H.A.; Miller, S.D. An experimental exploration of route choice: Identifying drivers choices and choice patterns, and capturing network evolution. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 1005–1012.
8. Denrell, J.; March, J.G. Adaptation as information restriction: The hot stove effect. *Organ. Sci.* **2001**, *12*, 523–538. [[CrossRef](#)]
9. Tversky, A.; Kahneman, D. Prospect theory: An analysis of decision under risk. *Econometrica* **1979**, *47*, 263–291.
10. Katsikopoulos, K.V.; Duse-Anthony, Y.; Fisher, D.L.; Duffy, S.A. Risk attitude reversals in drivers' route choice when range of travel time information is provided. *Hum. Factors: J. Hum. Factors Ergon. Soc.* **2002**, *44*, 466–473. [[CrossRef](#)]
11. Katsikopoulos, K.V.; Duse-Anthony, Y.; Fisher, D.L.; Duffy, S.A. The framing of drivers' route choices when travel time information is provided under varying degrees of cognitive load. *Hum. Factors: J. Hum. Factors Ergon. Soc.* **2000**, *42*, 470–481. [[CrossRef](#)]
12. Iida, Y.; Akiyama, T.; Uchida, T. Experimental analysis of dynamic route choice behavior. *Transp. Res. Part Methodol.* **1992**, *26*, 17–32. [[CrossRef](#)]
13. Yang, H.; Kitamura, R.; Jovanis, P.P.; Vaughn, K.M.; Abdel-Aty, M.A.; Reddy, P.D. *Exploration of Driver Route Choice with Advanced Traveler Information Using Neural Network*; California Partners for Advanced Transportation Technology (PATH): Berkeley, CA, USA, 1993.
14. Ben-Elia, E.; Erev, I.; Shiftan, Y. The combined effect of information and experience on drivers' route-choice behavior. *Transportation* **2008**, *35*, 165–177. [[CrossRef](#)]
15. Ben-Elia, E.; Shiftan, Y. Which road do I take? A learning-based model of route-choice behavior with real-time information. *Transp. Res. Part Policy Pract.* **2010**, *44*, 249–264. [[CrossRef](#)]
16. Srinivasan, K.K.; Mahmassani, H.S. Modeling inertia and compliance mechanisms in route choice behavior under real-time information. *Transp. Res. Rec. J. Transp. Res. Board* **2000**, *1725*, 45–53. [[CrossRef](#)]
17. Jou, R.C.; Lam, S.H.; Liu, Y.H.; Chen, K.H. Route switching behavior on freeways with the provision of different types of real-time traffic information. *Transp. Res. Part Policy Pract.* **2005**, *39*, 445–461. [[CrossRef](#)]
18. Polydoropoulou, A.; Ben-Akiva, M.; Kaysi, I. Influence of Traffic Information on Drivers' Route Choice Behavior. *Transp. Res. Rec. J. Transp. Res. Board* **1994**, *1453*, 56–65.

19. Srinivasan, K.K.; Mahmassani, H.S. Analyzing heterogeneity and unobserved structural effects in route-switching behavior under ATIS: a dynamic kernel logit formulation. *Transp. Res. Part Methodol.* **2003**, *37*, 793–814. [\[CrossRef\]](#)
20. Bogers, E.A.; Viti, F.; Hoogendoorn, S.P. Joint modeling of advanced travel information service, habit, and learning impacts on route choice by laboratory simulator experiments. *Transp. Res. Rec. J. Transp. Res. Board* **2005**, *1926*, 189–197. [\[CrossRef\]](#)
21. Tawfik, A.M.; Rakha, H.A. Latent Class Choice Model of Heterogeneous Drivers' Route Choice Behavior Based on Learning in a Real-World Experiment. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2334*, 84–94. [\[CrossRef\]](#)
22. Lee, J.H.; Cho, S.H.; Kim, D.K.; Lee, C. Valuation of Travel Time Reliability Accommodating Heterogeneity of Route Choice Behaviors. *Transp. Res. Rec.* **2016**, *2565*, 86–93. [\[CrossRef\]](#)
23. Kou, W.; Chen, X.; Yu, L.; Qi, Y.; Wang, Y. Urban commuters' valuation of travel time reliability based on stated preference survey: A case study of Beijing. *Transp. Res. Part Policy Pract.* **2017**, *95*, 372–380. [\[CrossRef\]](#)
24. Moghaddam, Z.R.; Jeihani, M. The Effect of Travel Time Information, Reliability, and Level of Service on Driver Behavior Using a Driving Simulator. *Procedia Comput. Sci.* **2017**, *109*, 34–41. [\[CrossRef\]](#)
25. Moghaddam, Z.R.; Jeihani, M.; Peeta, S.; Banerjee, S. Comprehending the roles of traveler perception of travel time reliability on route choice behavior. *Travel Behav. Soc.* **2019**, *16*, 13–22. [\[CrossRef\]](#)
26. Dai, R.; Lu, Y.; Ding, C.; Lu, G.; Wang, Y. A simulation-based approach to investigate the driver route choice behavior under the connected vehicle environment. *Transp. Res. Part Traffic Psychol. Behav.* **2019**, *65*, 548–563. [\[CrossRef\]](#)
27. Su, X.; Sperli, G.; Moscato, V.; Picariello, A.; Esposito, C.; Choi, C. An edge intelligence empowered recommender system enabling cultural heritage applications. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4266–4275. [\[CrossRef\]](#)
28. Mahmassani, H.S.; Jou, R.C. Transferring insights into commuter behavior dynamics from laboratory experiments to field surveys. *Transp. Res. Part Policy Pract.* **2000**, *34*, 243–260. [\[CrossRef\]](#)
29. Avineri, E.; Prashker, J.N. The impact of travel time information on travelers' learning under uncertainty. *Transportation* **2006**, *33*, 393–408. [\[CrossRef\]](#)
30. Dia, H. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transp. Res. Part Emerg. Technol.* **2002**, *10*, 331–349. [\[CrossRef\]](#)
31. Srinivasan, K.K.; Mahmassani, H.S. Role of congestion and information in trip-makers' dynamic decision processes: Experimental investigation. *Transp. Res. Rec. J. Transp. Res. Board* **1999**, *1676*, 44–52. [\[CrossRef\]](#)
32. Ramaekers, K.; Reumers, S.; Wets, G.; Cools, M. Modelling route choice decisions of car travellers using combined GPS and diary data. *Netw. Spat. Econ.* **2013**, *13*, 351–372. [\[CrossRef\]](#)
33. Papinski, D.; Scott, D.M.; Doherty, S.T. Exploring the route choice decision-making process: A comparison of planned and observed routes obtained using person-based GPS. *Transp. Res. Part Traffic Psychol. Behav.* **2009**, *12*, 347–358. [\[CrossRef\]](#)
34. Li, H.; Guensler, R.; Ogle, J. Analysis of morning commute route choice patterns using global positioning system-based vehicle activity data. *Transp. Res. Rec. J. Transp. Res. Board* **2005**, *1926*, 162–170. [\[CrossRef\]](#)
35. Abdel-Aty, M.A.; Kitamura, R.; Jovanis, P.P. Using stated preference data for studying the effect of advanced traffic information on drivers' route choice. *Transp. Res. Part Emerg. Technol.* **1997**, *5*, 39–50. [\[CrossRef\]](#)
36. Parkany, E.; Gallagher, R.; Viveiros, P. Are attitudes important in travel choice? *Transp. Res. Rec. J. Transp. Res. Board* **2004**, *1894*, 127–139. [\[CrossRef\]](#)





Article

Vehicle Trajectory Prediction Method Based on License Plate Information Obtained from Video-Imaging Detectors in Urban Road Environment

Zheng Zhang ¹, Haiqing Liu ^{1,*}, Laxmisha Rai ² and Siyi Zhang ¹

¹ College of Transportation, Shandong University of Science and Technology, Qingdao 266000, China; zheng_zhang163@163.com (Z.Z.); ZSY033186ZSY@163.com (S.Z.)

² College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266000, China; laxmisha@ieee.org

* Correspondence: hqliu@sdust.edu.cn; Tel.: +86-1586-554-3096

Received: 29 January 2020; Accepted: 24 February 2020; Published: 25 February 2020

Abstract: The vehicle license plate data obtained from video-imaging detectors contains a huge volume of information of vehicle trip rules and driving behavior characteristics. In this paper, a real-time vehicle trajectory prediction method is proposed based on historical trip rules extracted from vehicle license plate data in an urban road environment. Using the driving status information at intersections, the vehicle trip chain is acquired on the basis of the topologic graph of the road network and channelization of intersections. In order to obtain an integral and continuous trip chain in cases where data is missing in the original vehicle license plate, a trip chain compensation method based on the Dijkstra algorithm is presented. Moreover, the turning state transition matrix which is used to describe the turning probability of a vehicle when it passes a certain intersection is calculated by a massive volume of historical trip chain data. Finally, a *k*-step vehicle trajectory prediction model is proposed to obtain the maximum possibility of downstream intersections. The overall method is thoroughly tested and demonstrated in a realistic road traffic scenario with actual vehicle license plate data. The results show that vehicles can reach an average accuracy of 0.72 for one-step prediction when there are only 200 historical training data samples. The proposed method presents significant performance in trajectory prediction.

Keywords: vehicle trajectory prediction; license plate data; trip chain; turning state transit

1. Introduction

With economic development and the continuous expansion of the scale of cities, the number of vehicles increases sharply, inducing a frequent occurrence of road traffic offenses. In the urban traffic system, the supervision of abnormal vehicles, such as fake license plates, suspected of cases of illegal operation and other anomalies, have been viewed with high precaution by the traffic administration since they seriously threaten normal traffic order and safety. Intelligence analysis about these abnormal vehicles makes broader sense for assisting effective vehicle monitoring and management. The intelligence information includes all the holographic states of a vehicle, for example, the basic data of the vehicle and driver, the trajectory, the origin destination (OD) characteristics, aggregation with other vehicles, and others. It is really an important research topic on the real-time intelligence information analysis method by current devices deployed onboard or roadside.

In many intelligence information types mentioned above, the trajectory of the vehicle contains abundant spatial and temporal distribution features. Using a geographic information system (GIS), the trajectory can be accurately projected on an electronic map to give an intuitive presentation of the vehicle's movement. Moreover, through the analysis of massive trajectory data, the trip rules of

vehicles, such as the OD, and the travel preferences of the driver can also be obtained. These are very valuable for intelligent macro traffic management. In some microscopic traffic control applications, the vehicle trajectory is also meaningful. For example, the single trajectory prediction can be used to provide support for capturing illegal vehicles [1,2], while multiple trajectories prediction can be used to evaluate the short-time traffic volume for designing a proper traffic signal plan [3–8].

Using the electronic surveillance cameras deployed on roads, the driving status of a vehicle such as the time, license plate number, speed, lane and direction can all be acquired. On the basis of the road network topology, a vehicle passes through a series of intersections and the trajectory of the vehicle can be obtained easily from the driving status data collected by a video-imaging detector at the intersections. Compared with the methods based on location-based-services (LBS) [9–12], wireless fidelity (WiFi) probes [13–15] and cellular signaling [16–19], vehicle trajectory extracted from vehicle license plate data has certain advantages in terms of wide adaptability, accuracy and visualization effects. Fully using the driving status acquired by the video-imaging detectors, this paper studies the vehicle trajectory prediction scheme based on the latent travelling rules extracted from massive vehicle license plate data. For a certain vehicle, the turning characteristic at an intersection is described by a turning probability transition matrix in which the probability is calculated according to statistics of historical trip chains from the vehicle license plate. The experimental study shows that the proposed method presents a better performance in a short-term trajectory prediction.

The rest of the paper is organized as follows. Section 2 presents related work. In Section 3, trip chain building and compensating methods based on vehicle license plates are presented. In Section 4, the turning state transition matrix based on historical trip chains is proposed for trajectory prediction and the one/ k -step trajectory prediction models are described. The experimental study and the evaluation results are presented in Section 5. In Section 5, we conclude the paper and provide directions for future work.

2. Related Work

Vehicle trajectory is important intelligence information for both urban macroscopic traffic management and microscopic traffic control. Generally, vehicle trajectory uses a vehicle's or driver's location and identity as major data foundations. According to different vehicle or driver locations and identity acquisition types in an urban traffic environment, the current trajectory-building methods can be classified into the following categories: LBS-based methods, WiFi probe-based methods, cellular signaling-based methods and video-imaging-based methods.

The LBS-based method mainly uses global positioning system (GPS) floating vehicle data to track the target vehicle. The movement of the vehicle is detected continuously in time and the trajectory can be presented visually combined with the application of an electronic map. In [9], based on the vehicles' historical trips, the relationship between different road segments are built by transforming the road network model into a matrix, and the driving regularity of vehicles is analyzed for the design of the algorithm of vehicle route prediction. In [10], the historical vehicle GPS data is used to match the current trajectories and infer future possible destinations. The method predicts the trajectory by a systematic procedure for describing the features of the similarity of trajectories and destinations. The method also takes the station correlations and the user historical destinations into account. The positive prediction rate of the proposed method can reach 92% under the condition that the test trip has been completed over 70%. In [11], authors propose a vehicle trajectory prediction method based on the hidden Markov model. The relevant parameters are analyzed by historical vehicle GPS data, and the Viterbi algorithm is used to seek the double layers hidden states sequences corresponding to the recent driven trajectory. The future vehicle trajectory is predicted by a novel algorithm based on the hidden Markov model of double-layer hidden states. In [12], using the GPS data, a vehicle trajectory prediction method based on a variable-order Markov model is proposed. Kernel smoothing which combines sequence analysis with the Markov statistic is used for model building. The method presents a higher performance in prediction accuracy. However, these methods are only suitable for some special commercial vehicles

which are equipped with GPS or other onboard positioning devices, and not applicable for most private vehicles.

By installing certain WiFi probe devices [13] at intersections or roads, the WiFi probe-based method generates the trajectory by detecting the passing time and the media access control (MAC) addresses of electronic terminals with WiFi-connecting function, such as the onboard unit and driver's mobile phone. In [14], authors use a WiFi probe as the data collector to scan the mobile devices within a certain range in a certain period of time to obtain the MAC address, reference distance, time stamp and other information of mobile phone. Furthermore, a customer flow prediction model based on seasonal autoregressive integrated moving average (SARIMA) model and BP neural network model is built. In [15], an urban mobility trajectory analysis model based on large-scale WiFi probe request data is built. Unique entries per access point and per hour of WiFi data are aggregated to approximate local population counts by type of user. In the model, spatial network analysis is used to apply the results to the road and pedestrian sidewalk network to identify usage intensity levels and trajectories for individual street segments. The research demonstrates the significant potential in the use of WiFi probe request data for understanding mobility patterns. Similar in [16], authors design a user feature space in which frequent trajectory patterns are used to represent each user as a feature vector based on the anonymized WiFi scan lists. As per the popularity of electronic terminals, the application of WiFi probe is more adaptable than the LBS-based method. However, the identity of the target is denoted by the MAC addresses of the electronic terminal. It is not directly relevant to the vehicle itself. Besides, in order to achieve an urban-wide trajectory analysis, several WiFi probe devices should be built, inducing high expense for construction.

As with the WiFi probe scheme, the cellular signaling-based method also uses the location and identity of electronic terminals to generate the trajectory. By contrast with the former, the location is detected by the mobile station and the identity is generally the mobile user or the MAC of the mobile terminal. For example, in [17], the authors propose a data-driven method for dynamic people-flow prediction based on cellular probe data. The grid-based data transformation and data integration module is proposed to integrate multiple data sources for human daily trajectory generation. Moreover, a dynamic people-flow prediction model based on random forest is also presented. The experimental results show that the proposed method can provide prediction precision of 76.8% and 70% for outbound and inbound people, which is better than the single-feature model. In [18], the authors introduce a mobility modeling method based on real traffic data collected from 4G cellular networks, including data collection, trajectory construction, data noise removal, data storage and analysis. The experiments discover the user's mobility features, changing of city hotspots, and mobility patterns. However, locating using a mobile cellular network and mobile base station is inaccurate outdoors without supplementary GPS or WiFi devices. Hence, the precision of these methods is relatively low and they are only suitable for microscopic traffic and population evolution analysis.

Fully using the driving status data collected by electronic surveillance cameras on roads, the vehicle trajectory is acquired by time detecting and license plate number series. In [19], an offline method for historical OD pattern estimation based on automatic license plate recognition data is proposed. A particle filter is used to estimate the probability of a vehicle trajectory from all possible candidate trajectories combined with the time geography theory. In this method, the path flow estimation process is conducted through dividing the reconstructed complete trajectories of all detected vehicles into multiple trips. The proposed method is verified and the results show that the MAPEs of the OD estimation are lower than 19%. In [20], a vehicle trajectory extraction algorithm based on license plate recognition data is proposed. The license plate and timestamps are used for the establishment of trip chain. Aiming at the data loss problem when detecting a vehicle license plate, the K shortest path algorithm and gray relational analysis are further used for trip chain compensation. The research focuses on extracting the vehicle trajectory and the prediction of future driving state is not studied. In [21], authors propose a vehicle trajectory reconstruction method based on license plate data. In the method, travel time threshold is used to obtain a single travel chain and the similarity of the ideal

solution and depth first search method are used to build a vehicle trajectory reconstruction model. It can effectively solve the problem of incomplete license plate number data. However, the related research mainly focuses on the macroscopic trajectory modeling and OD analysis and is seldom concerned with the microscopic real-time vehicle trajectory prediction.

3. Methods

3.1. Trip Chain Building Based on Vehicle License Plate Information Obtained from Video-Imaging Detectors

In this section, we introduce the original data collected by the video-imaging detectors, and establish the corresponding mathematical model based on the actual road network. Meanwhile, using the Dijkstra algorithm, the missing data is supplemented and the travel chain is divided according to the time-cost matrixes.

3.1.1. Preparations for the Trip Chain Building

The whole urban road network consists of intersections and sections. Based on graph theory, the whole road network can be represented by a binary group composed of nodes and edges, as shown in Equation (1).

$$G = (V, E) \quad (1)$$

In the binary group, V denotes the set of intersections and E denotes the set of sections. V and E are expressed by Equations (2) and (3) respectively.

$$V = \{v_1, \dots, v_i, \dots, v_M\} \quad (2)$$

$$E = \{< v_i, v_j > \mid i, j \in M\} \quad (3)$$

where M is the number of intersections in the road network. In Equation (3), $< v_i, v_j >$ denotes that there is a road section between the i -th and the j -th intersection, i.e., the two intersections are directly connected.

Considering the directions of the sections and the distance between two adjacent intersections, we use the distance and the travelling time to represent the weights of the edge, and the cost matrixes are shown by Equations (4) and (5).

$$Dis = \begin{bmatrix} d_{1,1} & \cdots & d_{1,j} & \cdots & d_{1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{i,1} & \cdots & d_{i,j} & \cdots & d_{i,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{M,1} & \cdots & d_{M,j} & \cdots & d_{M,M} \end{bmatrix} \quad (4)$$

$$Tra = \begin{bmatrix} tr_{1,1} & \cdots & tr_{1,j} & \cdots & tr_{1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ tr_{i,1} & \cdots & tr_{i,j} & \cdots & tr_{i,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ tr_{M,1} & \cdots & tr_{M,j} & \cdots & tr_{M,M} \end{bmatrix} \quad (5)$$

For the cost matrixes, $d_{i,j}$ and $tr_{i,j}$ denote the distance and travelling time, respectively, from the i -th intersection to the j -th intersection when they are directly connected as given in Equation (3). If the i -th intersection to the j -th intersection are not directly connected or $i = j$, $d_{i,j}$ and $tr_{i,j}$ are assigned ∞ . In this paper, the travelling time is calculated by vehicle license plate data collected by video-imaging detectors referring to [22–24].

Electronic video detectors deployed at the intersection can collect the driving states of a passing vehicle, including the vehicle license plate, detecting time, lane number, vehicle type, body color and others. Moreover, each video detector has basic installation information, for example, the position (longitude and latitude) where the device located, the unique ID of the device, direction of the intersection which it detects, correlations about the intersections and lane. When a single vehicle is on a trip, it will be detected by a series of video detectors on the road and a set of driving states will be formed, expressed as:

$$Ts = \{S_i\}, i = 1, \dots, N \quad (6)$$

where N is the number of samples during the whole trip. Each sample in the series is presented by Equation (7).

$$S_i = (t_i, u_i, g_i, v_i, h_i, l_i, v'_i) \quad (7)$$

In Equation (7), the meaning of each field is explained as follows:

t_i is the detection time.

u_i is the unique ID of the video detector.

g_i is the position where the video detector is located. It is expressed by the longitude and latitude.

v_i is the intersection where the video detector locates.

h_i is the approach direction information of the intersection. In this paper, the direction code is numbered clockwise from a certain approach.

l_i is the lane information of the approach. In this paper, the lane code is numbered from inside lane to outside.

v'_i is the downstream intersection of the current lane. It is acquired by the connectivity of adjacent intersections and channelization.

3.1.2. Trip Chain Optimization and Division Based on Vehicle License Plate

In Equation (6), when all the samples are sorted over time by detection data, the series represents a whole trip chain in the sampling time period. In this section, the whole trip chain is firstly optimized and verified. Furthermore, it is divided into sub-trip chains based on the time interval feature of adjacent samples.

In actual applications, some intersections are not installed with video devices or those installed devices may be damaged. Even though the devices work normally, there are still missing detections or errors in detection of the vehicle license plate with a certain probability caused by the poor lighting condition, the performance of license plates recognition algorithm, and other reasons. Hence, the trip chain acquired by the original data of vehicle license plate is not consecutive in general. For some adjacent samples, the two intersections where video devices are located are not directly connected in the road network graph, as shown in Figure 1.

For any two adjacent samples S_i and S_{i+1} in Ts , when there is an undetected intersection between them, the values in the cost matrix presented in Equation (4) or (5) should be equal to ∞ , that is:

$$d_{v_i, v_{i+1}} = \infty \text{ or } tr_{v_i, v_{i+1}} = \infty \quad (8)$$

In order to obtain a complete trip chain for further vehicle driving behavior analysis, the data of the undetected intersection should be compensated when there are missing detections between S_i and S_{i+1} . Suppose that the vehicle drives following the shortest path, the Dijkstra algorithm is used to compensate the trip chain where the two intersections v_i and v_{i+1} are taken as the origin and destination, respectively. In the road network graph, the compensating intersections series is described by Equation (9) and the situation is shown by Figure 2.

$$V^{i,i+1} = \{v_1^{i,i+1}, \dots, v_k^{i,i+1}, \dots, v_{N_c}^{i,i+1}\} \quad (9)$$

where $v_k^{i,i+1}$ denotes the k -th intersection between v_i and v_{i+1} in the trip chain. N_c is the total number of compensating intersections.

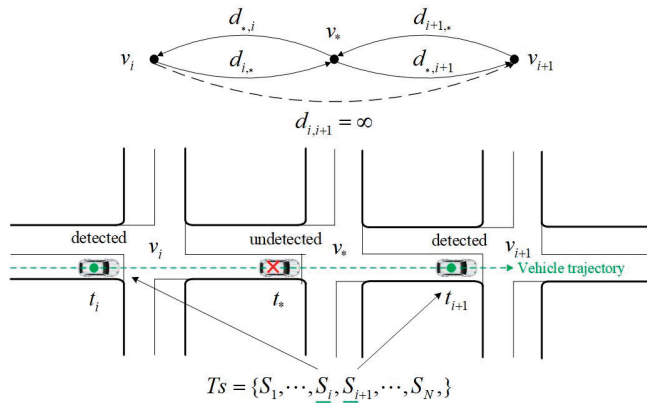
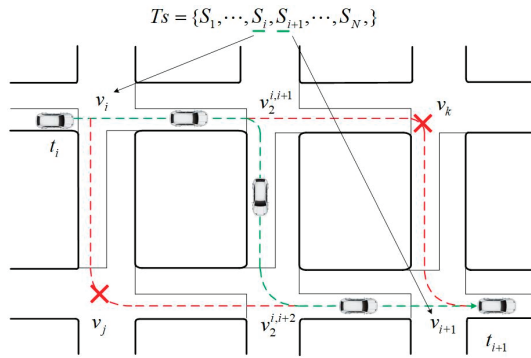


Figure 1. Presentation of missing data and the topology.



Green routes are shorter than red routes Thus: $V^{i,i+1} = \{v_1^{i,i+1}, v_2^{i,i+1}\}$

Figure 2. Presentation of intersection series after compensation.

After obtaining the compensating intersections, fields such as position, approach direction, lane information and downstream intersection can all be acquired based on the connectivity of adjacent intersections in the road network graph and channelization in actual scenario.

Considering the calculation error and randomness of the vehicle driving features, we take $\lambda_{i,j} \bullet tr_{i,j}$ as the upper limit of travel time from intersection i to intersection j , where $\lambda_{i,j}$ is the amplification coefficient. If the actual travelling time for a single vehicle is bigger than $\lambda_{i,j} \bullet tr_{i,j}$, it is identified that the vehicle stops between the i -th and j -th intersection, and the trip chain should be cut off at that place. Referring to this principle, the effectiveness of the compensating nodes is judged as follows:

When the actual travelling time is bigger than sum of upper thresholds of the compensating sections between v_i and v_{i+1} , as described by Equation (10):

$$\lambda_{v_k^{i,i+1}, v_{k+1}^{i,i+1}} \sum_{k=1}^{N_c} tr_{v_k^{i,i+1}, v_{k+1}^{i,i+1}} < t_{i+1} - t_i \tag{10}$$

The compensating intersections series presented in Equation (9) is ineffective. Under this condition, v_i is set as the destination for the former trip chain, and v_{i+1} is set as a new origin for a new trip chain.

Otherwise,

$$\lambda_{i,i+1} \sum_{k=1}^{N_c} tr_{v_k^{i,i+1}, v_{k+1}^{i,i+1}} \geq t_{i+1} - t_i \tag{11}$$

The compensating intersections series presented in Equation (9) is effective and the detection time is calculated by Equation (12):

$$t_{v_k^{i,i+1}} = t_i + \sum_{j=1}^k \frac{tr_{v_j^{i,i+1}}}{\sum_{l=1}^{N_c} tr_{v_l^{i,i+1}}} (t_{i+1} - t_i) \tag{12}$$

By the aforementioned operations, all necessary fields for the compensating samples of a trip chain can be acquired. Since the compensation strategy is proposed based on the assumption that the vehicle drives following the shortest path, there may be some departures with the actual trajectory. To confirm that the compensating samples are accurate enough, we further propose a verification and optimization scheme based on the turning state and downstream intersection. After compensation, the new trip chain can be presented by Equation (13).

$$Ts = \{S_1, \dots, S_i, S_1^{i,i+1}, \dots, S_k^{i,i+1}, \dots, S_{N_c}^{i,i+1}, S_{i+1}, \dots, S_N\}_{1 \times (N+N_c)} \tag{13}$$

In Equation (13), the next sample of S_i is $S_1^{i,i+1}$. If the downstream intersection v_i^j in S_i is not in accordance with the $v_1^{i,i+1}$ in $S_1^{i,i+1}$, the acquired N_c samples are incorrect and should be re-compensated. The re-compensation algorithm flowchart is presented in Figure 3.

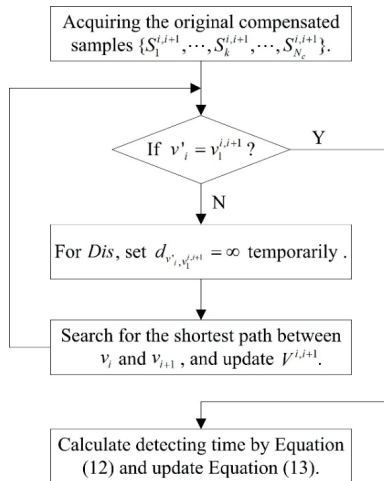


Figure 3. Flowchart of re-compensation algorithm.

For simplicity, the whole trip chain presented by Equation (13) is further expressed by a general form, as shown in Equation (14).

$$Ts = \{S_1, \dots, S_i, \dots, S_{N+N_c}\} \tag{14}$$

In the actual scenario, the whole trip consists of one or more sub-trip chains, where each sub-trip chain denotes a complete trip from the origin to destination. The detection time interval of any adjacent samples in Equation (14) denotes the travelling time of the vehicle in the section between v_i and v_{i+1} . When the time interval is bigger than a certain threshold, it implies that the vehicle finishes the trip

at some place between v_i and v_{i+1} . Under this condition, v_i and v_{i+1} belong to different sub-trip chains. Similarly, we take $\lambda_{i,i+1} \cdot tr_{i,i+1}$ as the threshold to divide the trip chain. For the series shown in Equation (14), the detecting time interval of adjacent samples is calculated in order.

$$t_{i+1} - t_i > \lambda_{i,i+1} \cdot tr_{i,i+1} \tag{15}$$

The trip chain is divided into two sub-trip chains, in which v_i is the destination of the former sub-trip chain, and v_{i+1} is the origin of the following sub-trip chain, as shown in Equation (16):

$$Ts = \begin{cases} Ts(1) \\ Ts(2) \end{cases} \tag{16}$$

where,

$$Ts(1) = \{S_1, \dots, S_i\} \tag{17}$$

$$Ts(2) = \{S_{i+1}, \dots, S_{N+N_e}\} \tag{18}$$

Because of the large data coverage time range, the number of vehicle trips is often greater than two. Therefore, according to the above method, the travel chain can be divided into T sub-trip chains, as shown by Equation (19).

$$Ts = \begin{cases} Ts(1) \\ \dots \\ Ts(j) \\ \dots \\ Ts(N_T) \end{cases} \tag{19}$$

where N_T is the number of sub-trip chains of the vehicle in the sampling time period.

3.2. Vehicle Trajectory Prediction Model Based on Turning State Transition Matrix

The series of intersections in the trip chain contains the turning information when a vehicle passes a certain intersection. The turning state transition matrix denotes the probability matrix for which direction a vehicle may take. Considering that the series of intersections for the j -th sub-trip chain are denoted by Equation (20):

$$V^j = \{v_1^j, \dots, v_{i'}^j, \dots, v_{N_j}^j\} \tag{20}$$

Referring to the series presented in Equation (20), it is easy to acquire the downstream intersection of each node in the j -th sub-trip chain when the vehicle is driving on the road. In the k -th intersection, assuming that there are N_a approaches and N_e exits with the associated downstream intersections denoted as $\{v'_k(1), \dots, v'_k(N_e)\}$. The turning state of the case vehicle at a certain intersection can be described by Equation (21).

$$B^j = \begin{bmatrix} b_{a_1, e_1}^j & \dots & b_{a_1, e_{N_e}}^j \\ \vdots & \ddots & \vdots \\ b_{a_{N_a}, e_1}^j & \dots & b_{a_{N_a}, e_{N_e}}^j \end{bmatrix}_{N_a \times N_e} \tag{21}$$

In Equation (21), b_{a_m, e_n} denotes the turning relationship when a vehicle drives passing the intersection. When the vehicle enters the intersection from the m -th approach and leaves from the n -th exit, then,

$$b_{a_m, e_n} = 1 \tag{22}$$

Otherwise,

$$b_{a_m, e_n} = 0 \tag{23}$$

For the j -th sub-trip chain acquired by Equation (19), the turning relationship can be obtained by the series of intersections and the turning state of the case vehicle (Equation (21)) is established,

denoted as B^j . In extended time, the case vehicle passes the same intersection for many times. Hence, the total turning state of the vehicle at a certain intersection can be calculated by the sum of all the turning state matrixes. For a case vehicle, suppose that there are N_T sub-trip chains passing through the i -th intersection, the total turning states of the vehicle can be calculated by Equation (24):

$$B_i = \sum_{j=1}^{N_T} B^j = \begin{bmatrix} \sum b_{a_1, e_1}^j & \cdots & \sum b_{a_1, e_{N_e}}^j \\ \vdots & \ddots & \vdots \\ \sum b_{a_{N_a}, e_1}^j & \cdots & \sum b_{a_{N_a}, e_{N_e}}^j \end{bmatrix}_{N_a \times N_e} \quad (24)$$

In addition, the turning state transition matrix is acquired by Equation (25):

$$Pr_i = B_i / N_T = \begin{bmatrix} \sum b_{a_1, e_1}^j / N_T & \cdots & \sum b_{a_1, e_{N_e}}^j / N_T \\ \vdots & \ddots & \vdots \\ \sum b_{a_{N_a}, e_1}^j / N_T & \cdots & \sum b_{a_{N_a}, e_{N_e}}^j / N_T \end{bmatrix}_{N_a \times N_e} \quad (25)$$

In Equation (25), $\sum b_{a_m, e_n}^j / N_T$ denotes the probability when the vehicle chooses the n -th exits from the m -th approach. For each row of the matrix shown in Equation (25), the following constraints should be satisfied:

$$\sum_{n=1}^{N_e} \sum b_{a_m, e_n}^j / N_T = 1 \quad (26)$$

Equation (26) implies that, in case the vehicle drives in an intersection from a certain approach, it must go out from one of the exits. However, for some intersections, there may be no effective trip chain, i.e, the case vehicle does not pass the intersection during experimentation. When this happens, the turning state probability for each exit is assigned an equal average probability value, as shown in Equation (27):

$$\sum b_{a_m, e_n}^j / N_T = 1 / N_e \quad (27)$$

When a vehicle enters the intersection from the m -th approach, it will go to the n -th downstream intersection from the n -th exit at a probability of $\sum b_{a_m, e_n}^j / N_T$. Hence, the one-step prediction probability of the vehicle for the next intersection v'_i is calculated by Equation (28).

$$P_{i+1} = O_i \cdot Pr_i = [o_{a_1}^i, \dots, o_{a_{N_a}}^i]_{1 \times N_a} \cdot \begin{bmatrix} \sum b_{a_1, e_1}^j / N_T & \cdots & \sum b_{a_1, e_{N_e}}^j / N_T \\ \vdots & \ddots & \vdots \\ \sum b_{a_{N_a}, e_1}^j / N_T & \cdots & \sum b_{a_{N_a}, e_{N_e}}^j / N_T \end{bmatrix}_{N_a \times N_e} \quad (28)$$

In Equation (28), O_i is the original state of the vehicle. When the vehicle is originally detected at the m -th approach:

$$o_{a_m}^i = 1 \quad (29)$$

For other approaches,

$$o_{a_1, \dots, a_{m-1}, a_{m+1}, \dots, a_{N_a}}^i = 0 \quad (30)$$

From Equation (28), the turning probabilities of the vehicle to the downstream intersections are acquired, expressed by Equation (31):

$$P_{i+1} = [p_{e_1}^{i+1}, \dots, p_{e_{N_e}}^{i+1}] \quad (31)$$

Based on the analysis above, the one-step predicted intersection which the vehicle will pass through is the downstream intersection corresponding to the maximum probability $\max\{p_{e_n}^{i+1}\}$. The one-step prediction method can be described by Figure 4.

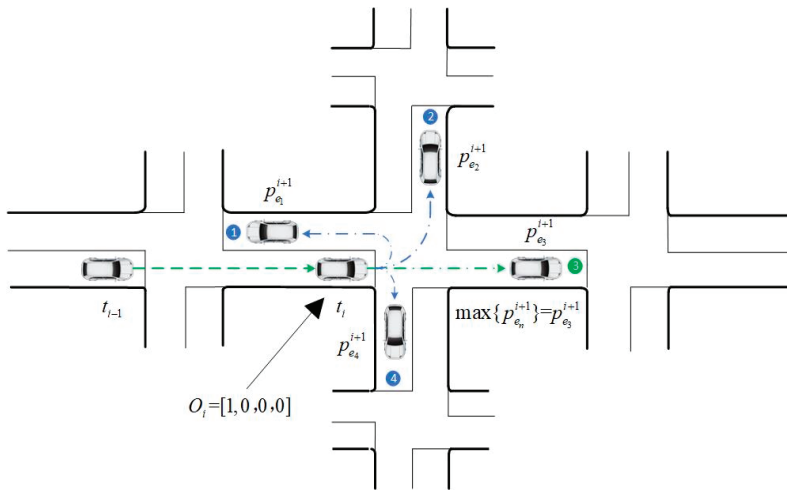


Figure 4. One step trajectory prediction.

The k -step prediction probability of the vehicle for the next k intersections is calculated by Equation (32).

$$P_{i+k} = O_{i+k-1} \cdot Pr_{i+k-1} \tag{32}$$

In Equation (32), O_{i+k-1} is the original probability state of v_{i+k-1} . Referring to the directly connected relationship with the upstream intersection v_{i+k-2} , assuming that the vehicle comes from the m -th approach of v_{i+k-1} , o_{a_m} is valued as the turning probability based on the upstream intersection v_{i+k-2} . For other approaches,

$$o_{a_1, \dots, a_{m-1}, a_{m+1}, \dots, a_{N_a}} = 0 \tag{33}$$

Similarly, the k -step predicted intersection is the k -th downstream intersection corresponding to the maximum probability $\max\{P_{e_n}^{i+k}\}$.

The k -step prediction method can be described by Figure 5.

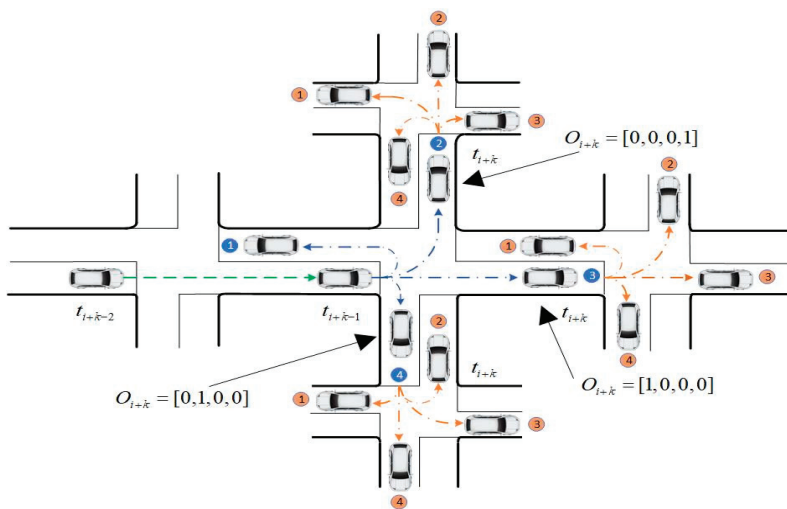
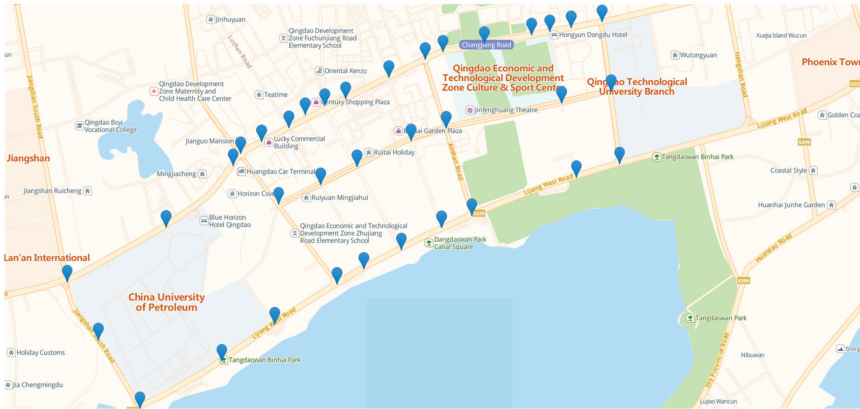


Figure 5. K-step trajectory prediction.

4. Experiments and Discussion

In this section, a regional road network in Qingdao, China, is selected for the case study. In the network, there are 27 intersections, 40 sections and 35 positions deploying with video-imaging cameras, as shown in Figure 6.



(a)



(b)

Figure 6. The positions of video-imaging cameras and road network topology in the case study. (a): Distribution of video-imaging cameras. (b): Road topology.

The original vehicle license plate data sample is acquired from the video-imaging detectors in actual traffic scenario. The proposed method is evaluated based on actual historical video-imaging data for the duration of one month.

4.1. Results of Trip Chain Building and Compensation

In the proposed method, travel time threshold between adjacent intersections is highly necessary for dividing the trip chain into different sub-trip chains. Since the traffic states are different at different time periods per day, the threshold should be calibrated according to the traffic variation. In this section,

we take the morning and evening rush hours as examples to present the amplification coefficient $\lambda_{i,j}$ calibration progress.

Considering the section in Figure 6b as an example, the samples at 7:00–9:00 AM for morning rush hours and 17:00–18:00 for evening rush hours in the original dataset in one month are extracted and analyzed, and the statistics of travelling time for all vehicles are presented in Figure 7.

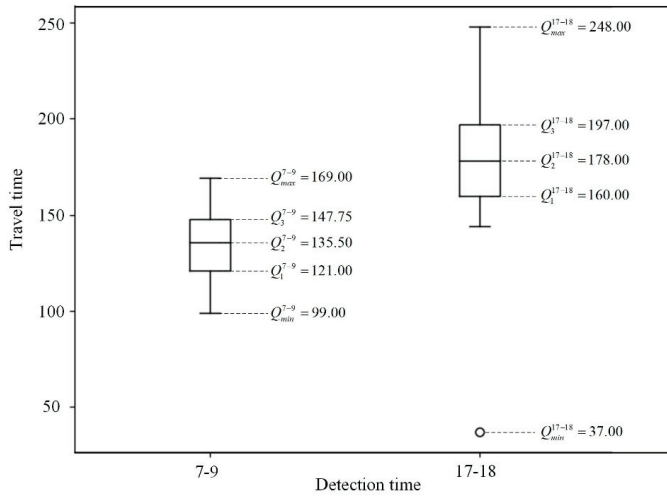


Figure 7. An example of travelling time value distribution of the case section.

From Figure 7, it is evident that the traffic flow of the case section has typical tidal feature since the travelling time values in evening rush hours are much higher than the mornings. However, the travel time values are clustered in major regions referring to different time periods. If the threshold is set too small, some normal sub-trip chains will be over-segmented and much useful information will be lost for the establishment of the turning state transition matrix. If the threshold is too big, two sub-trip chains will be considered as one, inducing a misjudgment of the vehicle travelling state at that the joint points. In order to avoid this, the amplification coefficient $\lambda_{i,j}$ is calculated by the ratio of the upper value and the average travelling time in Equation (5) after excluding data outliers, as shown in Equation (34).

$$\lambda_{i,j} = Q_{max} / Tr_{i,j} \tag{34}$$

After acquiring the trip chains of a target vehicle, missing data points are compensated by the method proposed in Section 3. In order to evaluate the performance of the proposed trip chain compensation method, part of consecutive sampling nodes are selected and removed artificially from a whole trip chain. In this paper, at most 5 consecutive sampling nodes are compensated for. Based on the original trip chain shown in Equation (6), 1 to 5 consecutive sampling nodes are removed respectively to obtain sample sequences for compensation, as shown in Equation (35).

$$Ts_{con} = \{S_1, \dots, S_{i-1}, S_{i+n_{con}}, \dots, S_N\}, i = 2, 3, \dots, N - n_{con} \tag{35}$$

In Equation (35), The total number of cases is $N - n_{con}$, and the number of consecutive nodes for compensation is n_{con} .

Using the method proposed in Section 3.2, the removed nodes in Equation (35) are compensated. To assess the performance of the compensation method quantitatively, the compensation accuracy is proposed. It is calculated by the ratio of the number of correct nodes after compensation to the total number of nodes for compensation. The compensating accuracy under different cases is shown in Figure 8.

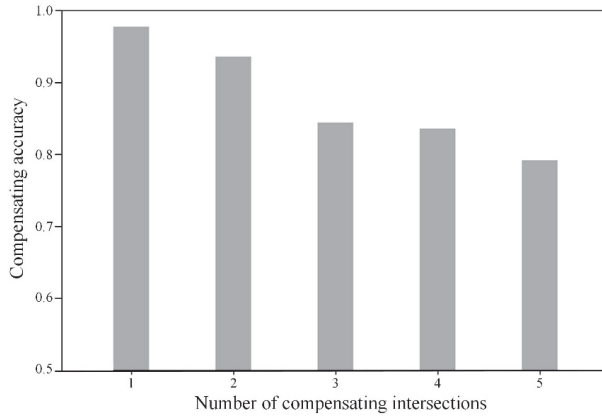


Figure 8. Compensating accuracy under different cases.

From Figure 8, it is evident that the proposed method presents a significant performance in the compensating missing nodes. All the cases are with a high accuracy of more than 80%. Moreover, the accuracy presents a declining trend with the increase of the number of nodes for compensation. This is because when several consecutive sampling nodes are missing, there will be more possible trajectories for the vehicle in the undetected region. The Dijkstra method may not perform very well in a large and complex road network.

4.2. Trajectory Prediction Results and Analysis

Among all vehicles in the original data sample, a section of the vehicles are selected for the verification of the performance of trajectory prediction. In this paper, all the trip chains of the case vehicles are acquired from the original data. Moreover, part of the historical trip chains are used for training the turning state transition matrix and the remaining trip chains are used for testing the accuracy of the trajectory prediction results. For one to four-step prediction, the results of 10 case vehicles are presented in Figure 9.

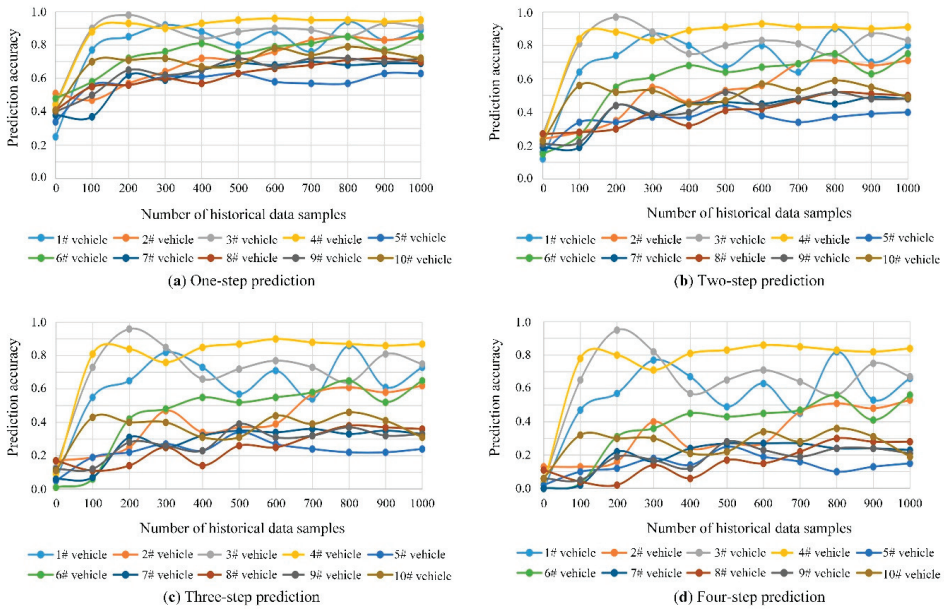


Figure 9. One to four-step trajectory prediction results of case vehicles.

In Figure 9, it is evident that, the accuracy varies significantly among different vehicles. As shown in Figure 9, 1#, 3# and 4# vehicles present a much higher prediction accuracy than others for one to four-step trajectory. This presentation is mainly caused by the regularity of vehicle driving characteristics. For vehicle trajectories that are relatively regular, such as the trajectories created by the commuters to and from work in each working day, the accuracy presents much high and stable values, while for the random travelling trajectories, such as the trajectories from taxis, the accuracy is relatively low. For example, the 5# vehicle presents a low prediction accuracy and large fluctuation with the gradual increase of training data. In order to show the results more clearly, the average prediction accuracy for testing vehicles together with the fitting results are further presented in Figure 10. According to the variation of the accuracy values, the logarithmic function is applied for the fitting, as shown in Equation (36).

$$y = a \ln(x + b) + c \quad (36)$$

In Figure 10, with the increase of the amount of training data, the accuracy presents a rising trend. More training data contains more information about the trip chains so that the turning state transition matrix can describe the travelling characteristics more accurately. In the case analysis, vehicles can reach an average accuracy of 0.72 for one-step prediction on the basis that there are more than 200 training data samples. Hence, the proposed method presents a better performance in trajectory prediction. Moreover, the accuracy presents an overall downward trend with the increase of number of prediction steps. The maximum accuracy is about 0.80, 0.63, 0.51 and 0.43 for one-step, two-step, three-step and for four-step trajectory prediction, respectively. The reason is that there are more cases for the vehicle to choose the following intersections with the increase of the number of prediction steps. As the trajectory becomes more unpredictable, the accuracy declines.

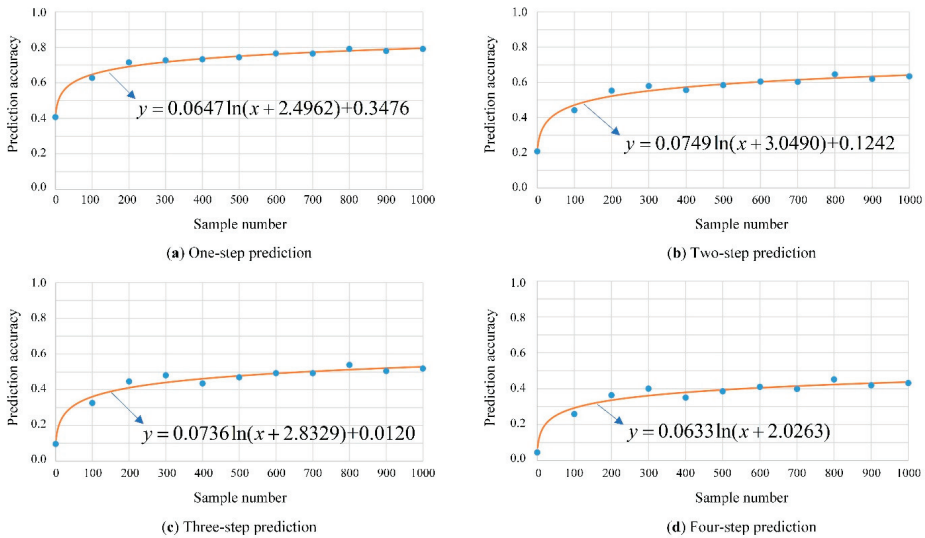


Figure 10. Average prediction accuracy and the fitting results for different prediction steps.

5. Conclusions and Future Work

This paper proposes a vehicle trajectory prediction algorithm based on license plate data collected from video-imaging detectors. In order to obtain more complete vehicle travel information, we use the Dijkstra algorithm for data compensation. The driving characteristics are described by the turning state transition matrix which is acquired by the historical trip chains based on the time series of license plate data. Based on the turning state transition matrix, we make a multi-step prediction for specific vehicles. The experimental results show that, although the performance of trajectory prediction for different vehicles varies significantly, the proposed vehicle trajectory prediction algorithm has high average accuracy at the expense of a simple calculation, especially for one-step prediction. Compared with the traditional schemes, the proposed method fully exploits the potential value of existing data and without any extra investment needed. This is really beneficial for urban traffic feature analysis and traffic management.

In this paper, the vehicle license plate data obtained from video-imaging detectors is the unique input of the proposed method. A high-quality license plate data set is the prerequisite for the implementation of the method. Some subtle errors in the original data, such as timestamp error, detector positioning error and others, should be eliminated. Hence in actual applications, a sophisticated data pre-processing scheme is indispensable.

Future research mainly focuses on two aspects. Firstly, the proposed method can be verified using a license plate data set of 10 vehicles in one month. In order to acquire more precise conclusions, the data sample should be further expanded. Secondly, according to the general understanding, the driving characteristics of a section of vehicles in an urban environment is time-sensitive to some extent. Hence, an analysis of the sensitivity of historical data to the prediction accuracy will be carried out.

Author Contributions: Conceptualization, H.L. and Z.Z.; methodology, H.L. and Z.Z.; software, Z.Z.; validation, H.L., Z.Z. and S.Z.; formal analysis, H.L.; investigation, H.L.; resources, H.L.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, L.R., H.L. and Z.Z.; visualization, Z.Z.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shandong Provincial Natural Science Foundation, grant number ZR2019QF017 and Basic Research Plan on Application of Qingdao Science and Technology, grant number 19-6-2-3-cg.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Li, M.X.; Mao, J.L.; Qi, X.D.; Yuan, P.S.; Jin, C.Q. Cloned Vehicle Behavior Analysis Framework. In Proceedings of the APWeb and WAIM Joint International Conference on Web and Big Data, Macau, China, 23–25 July 2018; pp. 223–231.
- Shen, M.S.; Liu, D.R.; Shann, S.H. Outlier detection from vehicle trajectories to discover roaming events. *Inf. Sci.* **2015**, *294*, 242–254. [[CrossRef](#)]
- Kong, X.J.; Xu, Z.Z.; Shen, G.J.; Wang, J.Z.; Yang, Q.Y.; Zhang, B.S. Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Gener. Comput. Syst.* **2016**, *61*, 97–107. [[CrossRef](#)]
- Zhan, X.Y.; Zheng, Y.; Yi, X.W.; Ukkusuri, S.V. Citywide traffic volume estimation using trajectory data. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 272–285. [[CrossRef](#)]
- Yu, J.; Lu, P.Z.; Han, J.M.; Lu, J.F. Detecting Regularities of Traffic Signal Timing Using GPS Trajectories. *IEEE Trans. Inf. Syst.* **2018**, *101*, 956–963. [[CrossRef](#)]
- Yu, J.; Lu, P.Z. Learning traffic signal phase and timing information from low-sampling rate taxi GPS trajectories. *Knowl. Based Syst.* **2016**, *110*, 275–292. [[CrossRef](#)]
- Yang, Q.; Yu, J.; Han, J.M. Traffic Signals Timing Cycle Length Learning: Using Taxi Gps Trajectories. In Proceedings of the 2018 International Conference on Machine Learning and Cybernetics (ICMLC), Chengdu, China, 15–18 July 2018; pp. 13–18.
- Zhao, Y.; Zheng, J.F.; Wong, W.; Wang, X.M.; Meng, Y.; Liu, H.X. Various methods for queue length and traffic volume estimation using probe vehicle trajectories. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 70–91. [[CrossRef](#)]
- Ye, N.; Wang, Z.Q.; Malekian, R.; Zhang, Y.Y.; Wang, R.C. A method of vehicle route prediction based on social network analysis. *J. Sens.* **2015**, *2015*, 1–9. [[CrossRef](#)]
- Wang, L.; Zhong, Y.G.; Ma, W.J. GPS-data-driven dynamic destination prediction for on-demand one-way carsharing system. *Int. Intell. Transp. Syst.* **2018**, *12*, 1291–1299. [[CrossRef](#)]
- Ye, N.; Zhang, Y.Y.; Wang, R.C.; Malekian, R. Vehicle trajectory prediction based on Hidden Markov Model. *Ksii. Trans. Internet Inf. Syst.* **2016**, *10*, 3150–3170.
- Wang, X.; Jiang, X.H.; Chen, L.F.; Wu, Y. KVLMM: A Trajectory Prediction Method Based on a Variable-Order Markov Model With Kernel Smoothing. *IEEE Access* **2018**, *6*, 25200–25208. [[CrossRef](#)]
- Oliveira, L.; Schneider, D.; Souza, J.D.; Shen, W.M. Mobile Device Detection Through WiFi Probe Request Analysis. *IEEE Access* **2019**, *7*, 98579–98588. [[CrossRef](#)]
- Wu, Y.Y.; Gu, S.H.; Yu, T.; Xu, X.L. APS-PBW: The Analysis and Prediction System of Customer Flow Data Based on WIFI Probes. In Proceedings of the International Conference on Knowledge Science, Changchun, China, 17–19 August 2018; pp. 477–488.
- Traunmueller, M.W.; Johnson, N.; Malik, A.; Kontokosta, C.E. Digital footprints: Using WiFi probe and locational data to analyze human mobility trajectories in cities. *Comput. Environ. Urban Syst.* **2018**, *72*, 4–12. [[CrossRef](#)]
- Zhao, S.; Zhao, Z.; Zhao, Y.; Huang, R.; Li, S.; Pan, G. Discovering People’s Life Patterns from Anonymized WiFi Scanlists. In Proceedings of the 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, Bali, Indonesia, 9–12 December 2014; pp. 276–283.
- Chen, X.X.; Wan, X.; Ding, F.; Li, Q.; McCarthy, C.; Cheng, Y.; Ran, B. Data-Driven Prediction System of Dynamic People-Flow in Large Urban Network Using Cellular Probe Data. *J. Adv. Transp.* **2019**, *2019*, 1–12.
- Zhao, Z.B.; Zhang, P.; Huang, H.Z.; Zhang, X. User mobility modeling based on mobile traffic data collected in real cellular networks. In Proceedings of the 2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, QLD, Australia, 13–15 December 2017; pp. 1–6.
- Rao, W.M.; Wu, Y.J.; Xia, J.X.; Ou, J.S.; Kluger, R. Origin-destination pattern estimation based on trajectory reconstruction using automatic license plate recognition data. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 29–46. [[CrossRef](#)]

20. Ruan, S.B.; Wang, F.J.; Ma, D.F.; Jin, S.; Wang, D.H. Vehicle trajectory extraction algorithm based on license plate recognition data. *J. Zhejiang Univ.* **2018**, *52*, 836–844.
21. Yu, H.Y.; Yang, S.; Wu, Z.H. Vehicle trajectory reconstruction from automatic license plate reader data. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1–13. [[CrossRef](#)]
22. Ma, D.F.; Luo, X.Q.; Jin, S.; Guo, W.W.; Wang, D.H. Estimating maximum queue length for traffic lane groups using travel times from video-imaging data. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 123–134. [[CrossRef](#)]
23. Luo, X.Q.; Ma, D.F.; Jin, S.; Guo, W.W.; Wang, D.H. Queue Length Estimation for Signalized Intersections Using License Plate Recognition Data. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 209–220. [[CrossRef](#)]
24. Bozyiğit, A.; Alankuş, G.; Nasiboğlu, E. Public transport route planning, Modified dijkstra’s algorithm. In Proceedings of the International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017; pp. 502–505.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Using Vehicle Interior Noise Classification for Monitoring Urban Rail Transit Infrastructure

Yifeng Wang ¹, Ping Wang ¹, Qihang Wang ¹, Zhengxing Chen ¹ and Qing He ^{1,2,3,*}

- ¹ Key Laboratory of High-Speed Railway Engineering of the Ministry of Education, School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, China; yfw@my.swjtu.edu.cn (Y.W.); wping@swjtu.edu.cn (P.W.); qihangwang@my.swjtu.edu.cn (Q.W.); chenzhengxing@my.swjtu.edu.cn (Z.C.)
 - ² Department of Industrial and Systems Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA
 - ³ Department of Civil, Structural and Environmental Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA
- * Correspondence: qinghe@buffalo.edu

Received: 5 January 2020; Accepted: 15 February 2020; Published: 18 February 2020

Abstract: This study developed a multi-classification model for vehicle interior noise from the subway system, collected on smartphones. The proposed model has the potential to be used to analyze the causes of abnormal noise using statistical methods and evaluate the effect of rail maintenance work. To this end, first, we developed a multi-source data (audio, acceleration, and angle rate) collection framework via smartphone built-in sensors. Then, considering the Shannon entropy, a 1-second window was selected to segment the time-series signals. This study extracted 45 features from the time- and frequency-domains to establish the classifier. Next, we investigated the effects of balancing the training dataset with the Synthetic Minority Oversampling Technique (SMOTE). By comparing and analyzing the classification results of importance-based and mutual information-based feature selection methods, the study employed a feature set consisting of the top 10 features by importance score. Comparisons with other classifiers indicated that the proposed XGBoost-based classifier runs fast while maintaining good accuracy. Finally, case studies were provided to extend the applications of this classifier to the analysis of abnormal vehicle interior noise events and evaluate the effects of rail grinding.

Keywords: urban rail transit interior noise; smartphone sensing; XGBoost classifier; railway maintenance

1. Introduction

By the end of 2018, the total operating mileage of urban rail transit (URT) in China exceeded 5700 km, including 4350 km of subway lines, and it is expected to double in the next 3 to 5 years [1]. With the rapid extension of the URT network, the current maintenance mode relies on humans, and it is challenging to ensure the safe and stable operation of trains. Therefore, intelligent URT maintenance work should be promoted for higher efficiency.

As one of the most prevalent kinds of URT, subways are increasingly essential in people's daily lives. However, abnormal vibration and noise significantly affect passengers' riding experience. Moreover, these abnormalities provide information about wheel-rail interactions and degradation of the track structures. Generally, train-induced noise can be categorized as external or interior noises [2]. Vehicle interior noise which is pertinent to this study mainly consists of noise from electrical equipment, aerodynamic noise, and wheel-rail noises [3]. Usually, the aerodynamic noise is dominant when the train speed exceeds 250 km/h, and electrical equipment noise dominates for speeds slower than 35 km/h [4]. As the subway trains usually run at 30–80 km/h, the wheel-rail noise is the main component

of vehicle interior noise [5]. The wheel-rail interaction significantly influences the wheel-rail noise. Therefore, we assumed that there exists a mapping relationship between vehicle interior noises and wheel-rail interactions. This mapping relationship provides an approach to monitor track conditions through vehicle interior noise. Moreover, it would be convenient to develop a simple onboard interior noise monitoring system that contributes to the safety and reliability of the railway system.

Regarding vehicle interior noise, past studies have mainly focused on the generation mechanism, transmission characteristics, and control strategies [6–10]. Typical study topics, such as noise characteristics analysis [11], sound quality evaluation [12], and noise level prediction [13], can be attributed to the above research fields. However, because the vehicle-track coupling system consists of a large number of components, the interior noise is affected by numerous factors, such as track slab [14], rail roughness, wheel out-of-roundness [9], and car body structure [15]. These factors may interact with each other and influence the characteristics of vehicle interior noise. Therefore, researchers generally choose one or two factors, such as rail fastener stiffness [7] and wheel polygonal wear [9], to perform their analysis at a lower complexity.

Among related studies, the prediction of vehicle interior noise is one of the most prevalent topics because it benefits the design and construction of track-vehicle systems at the early stages. Methods such as the boundary element method (BEM) [16], finite element method (FEM) [17], and statistical energy analysis method (SEAM) [15] are commonly used in this. However, their effectiveness relies significantly on the selected boundary conditions and model parameters. Thus, these numerical models are generally applied for specific problems. Moreover, the results of field tests are also often used for model verification. Despite the effectiveness of the method combining analytical models, numerical simulation, and field tests in the study of vehicle interior noise, the difficulty to obtain model parameters limits its application. Moreover, field tests may also interfere with daily operations. Overall, these studies do not make the best use of data collected during the daily operation and maintenance of the railway system.

In this context, the railway transportation industry is at the forefront of implementing analytics and big data [18]. Machine learning (ML) and artificial intelligence (AI) are two concepts at the leading edge of information technology, both of which contribute to big data technology. In recent years, the implementation of ML in the railway industry has been widely studied, for example in the prediction of passenger flow [19], delay events [20], and railway operation disruptions [10]. Moreover, many cases have been reported for railway infrastructure management and maintenance, including the detection and diagnosis of defects [21–23], prediction of failure events [24,25], and forecast of remaining useful life of devices [26]. These studies indicate that ML technologies have a promising prospect in promoting intelligent railway maintenance, thus ensuring the safety of the railway transit system.

As for data on vehicle interior noise, users require automatic methods to segment, label, and store the increasing amount of acoustic data from monitoring systems. The major challenge in this field is the automatic classification of audio [27]. Recent studies on the classification of traffic noise have been conducted, for example, to identify the type of vehicle through roadside noise [28,29] and evaluate passengers' subjective experience by categorizing the cabin's interior noise [30]. However, compared with traffic noise, the factors influencing vehicle interior noise of subway trains are considerably more complicated.

For collecting track conditions, the railway industry has employed various dedicated devices, such as track inspection vehicles [31] and visual inspection systems [32]. Although these devices perform well in detecting track conditions, the expensive cost and the interference for regular operation limit their usage in urban rail transit systems. There are also some on-board devices being developed to monitor track conditions using in-service vehicles [33–35]. However, the installation of these devices may change the design characteristics of cars and cause potential safety issues. As of now, these novel on-board monitoring devices have not been widely used. As an integrated platform, a smartphone can achieve data collection, storage, and transmission individually. Besides, the smartphone is mature, cost-effective, and easy to use, promoting its application in various fields. Studies using the embedded

accelerometers of smartphones to monitor road conditions and evaluating the ride quality have been reported [36,37]. These research works inspired the authors to investigate the feasibility of using smartphones to collect multi-source data about subway vehicles.

According to the above literature review, current studies about vehicle interior noise mainly focus on its generation mechanism and influencing factors through analytical models, numerical simulations, and field tests. To the best of our knowledge, only a few studies have analyzed vehicle interior noise using data-driven methods. Therefore, this study aims to advance data mining of vehicle interior noise for decision making in rail maintenance, such as for rail grinding. In this context, there are two significant challenges. First, despite sensing technologies being well developed now, it is still difficult to establish an onboard data collection framework that is easy to deploy, cost-efficient, and reliable. Moreover, the simultaneous collection of dynamic responses from the car body and interior noise is essential because these two datasets are connected to each other. Second, due to the complexity of vehicle interior noise, the extraction of useful features and correct labeling of noise classes remain challenging.

The goal of this study is to mine useful information from the vast amount of interior noise data using ML methods. To pursue this goal, onboard smartphone data were collected, including dynamic responses and noises. Further, a series of analyses were performed to classify the noises and clarify the influencing factors. The novel contributions of this paper are summarized as follows:

1. A smartphone-based onboard data collection framework for vehicle interior noise and dynamic responses of the car body was established.
2. The theory of Shannon entropy was considered when selecting the optimal window size for segmenting the multi-source time-series signals.
3. A multi-classification model for subway vehicle interior noise was established based on the XGBoost algorithm. The generation of a set of 45 features and performing feature selection based on different methods were also included.
4. Case studies were conducted to extend the application scenario for the analysis of abnormal noise causes and evaluating the effect of rail grinding.

This paper is organized as follows. Section 2 briefly illustrates the research methodology. Section 3 introduces the data utilized in this study and its collection framework. Section 4 describes the modeling approaches, including data segmentation and time windows, and establishes the multi-classification model with the Extreme Gradient Boosting (XGBoost) method. Furthermore, Section 5 presents the analysis results and discussions. Finally, in Section 6, conclusions are drawn according to the relevant analysis.

2. Research Methodology

The research methodology of this study is shown in Figure 1. First, we developed an Android app that leverages built-in sensors of onboard smartphones to collect vehicle interior noise and the corresponding dynamic responses of the car body. Second, time windows were used to segment the multi-source signals and establish the corresponding relationship between the audio and other signals. This method was significantly effective in overcoming the difficulty brought by the different sampling frequencies of a variety of sensors. Third, features were generated and selected from the time- and frequency-domains. Fourth, an automatic classification model for train interior noise was developed using XGBoost, a tree-based method. Finally, the proposed model was validated based on field experiments on the subway line.

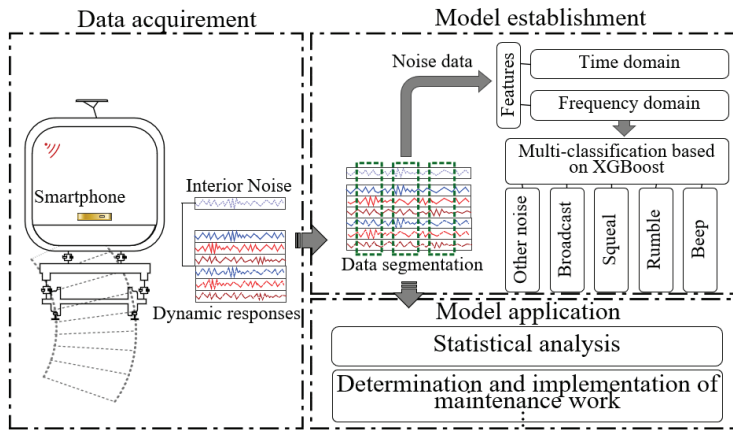


Figure 1. Research methodology of this study.

3. Data Collection and Description

Figure 2 shows the field test setup for data collection using Android smartphones (Huawei Honor FRD-AL00). During the test, the smartphone was placed on the cabin floor, right above the bogie to sense the response from the wheel-rail contact interface. In a parallel study, we verified that the differences between smartphone sensors and high-precision industry accelerators are acceptable, especially in the vertical direction [36]. Thus, the dynamic response signals can be considered a good record of the movement state of the car body. An app was developed to save and transmit the data to our cloud server. In the field test, three sensors were used, namely the microphone, accelerometer, and gyroscope. Moreover, considering the performance of these sensors and the characteristics of the signals, the sampling frequency of the accelerometer and gyroscope were set to 100 Hz, and that of the microphone to 22,050 Hz.

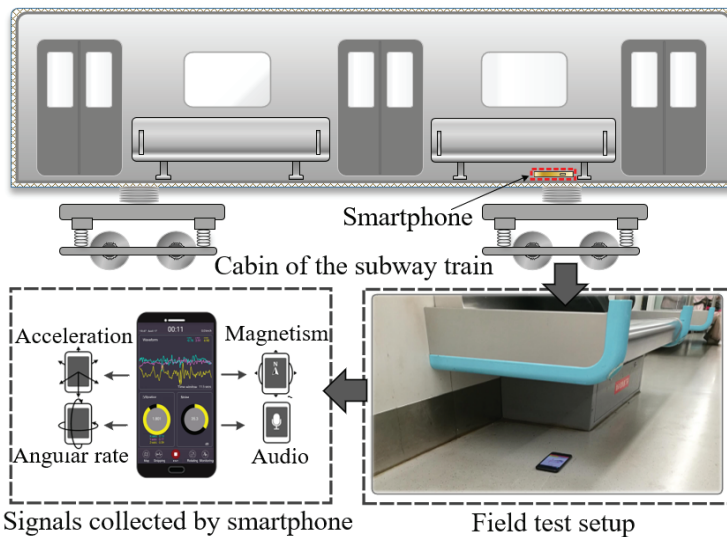


Figure 2. Data collection with the smartphone.

In this study, all tests were performed on Line 7 of the Chengdu Metro, China, which is a loop subway line. Its layout is shown in Figure 3a. This line covers 38.61 km and 31 stations, and it started operations in December 2017. The trains run along the outer and inner loop, with a maximum speed of 80 km/h. Because this is a loop line, it contains a large number of curve sections (166 curves). The radius distribution of these curves is presented in Figure 3b. It is challenging to maintain the track structures in good conditions due to the high number of curves, and the squeal that typically occurs along the curves is one of the most significant problems.

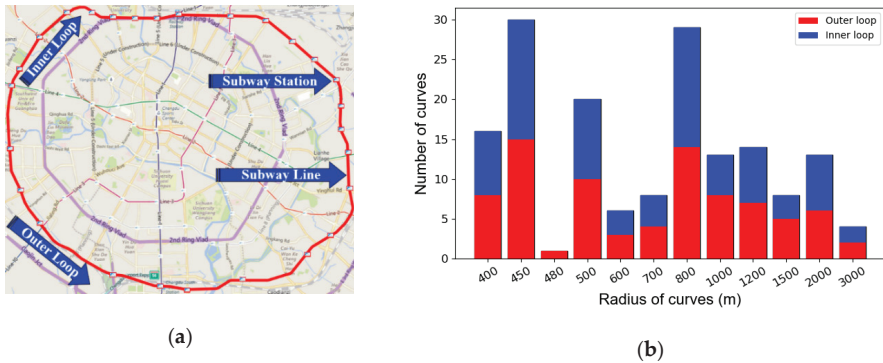


Figure 3. Line 7 of the Chengdu Metro, China: (a) Overview; (b) Radius of curves.

The data used in this study were collected on 2 August 2019, and 1 October 2019, before and after rail grinding. There were more abnormal events in the dataset before rail grinding. The data from August was used to train and test the multi-classification model, and to justify the need for rail grinding. The data measured on both days were compared. When training the model, we manually labeled the audio sequence into five groups, including 'Other noises', 'Broadcast', 'Squeal', 'Rumble', and 'Beep'. Here, 'Broadcast' refers to the official broadcast by the subway system or passengers' voices. 'Squeal' is an intense noise generated by the relative movement between wheel and rail. 'Rumble' refers to a low heavy sound when the train passes a specific area. 'Beep' is the alarm sound when a door is opened or closed. 'Other noises' refers to a sound which cannot be categorized into the above four classes. The time-frequency characteristics of these five classes of noise are presented in Figure 4.

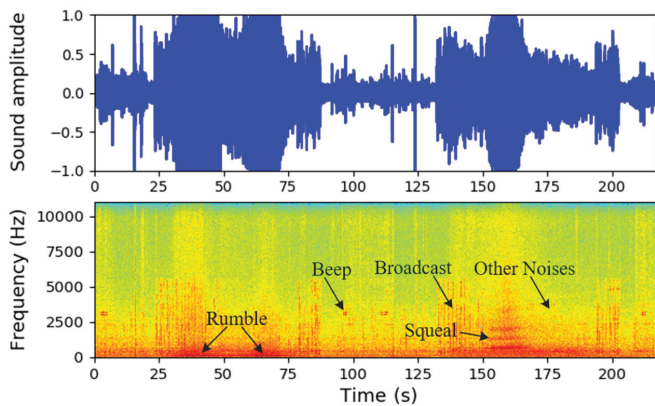


Figure 4. Data collection with the smartphone.

4. Model Approach

4.1. Data Segmentation and Time Window

Differences in sensor sampling frequencies make it difficult to identify the corresponding relationship among the multi-source signals. In this context, data segmentation is a typical method to preprocess continuous data and capture embedded features. This approach has been frequently implemented in activity recognition, such as in speech [38] and human activity [39] recognition. Therefore, we adopted the moving time-window method to segment the signals in our study. During data segmentation, there were two crucial parameters to be determined the size of the time window and the overlap between two adjacent windows. To avoid the duplication of data interference with statistical analysis, the overlap parameter was set to 0. That is, there was no overlap between two adjacent windows. Although the window method is normally used in data segmentation, there is no clear consensus on which window size should be employed [39]. The characteristics of vehicle interior noise are different from other audio signals. Therefore, we cannot use the window sizes used in speech recognition as a reference. Generally, small windows allow for on-point activity detection with a few resources and low energy costs. In contrast, large windows are usually considered to identify complex activities. To obtain the optimal window size for vehicle interior noise multi-classification, we leveraged the Shannon entropy and the actual requirements when labeling the training data manually.

We assumed that under the optimal window size, the system carries more information than under other situations [40]. The Shannon entropy is a method commonly used to describe the average information of a system, and it can be written as:

$$H = - \sum_{i=1}^m p(x_i) \log_2 p(x_i), \quad (1)$$

where x_i denotes the i th event; m represents the total number of events; and $p(x_i)$ is the probability when $x = x_i$ and $\sum_{i=1}^m p(x_i) = 1$. To obtain the optimal window size, the vehicle interior noise signal was first divided into a series of segment sequences according to different window sizes. The standard deviation of each segment was calculated to describe the state of the segment. Consequently, standard deviation sequences corresponding to different window sizes were available. It was then assumed that all values of standard deviation fall within the range of $(0, A]$, where A is the maximum standard deviation under different window sizes. After that, this interval was equally divided into m sub-intervals, where the i th sub-interval can be written as $(a_i, a_{i+1}]$, $a_1 = 0$, and $a_{m+1} = A$. Thus, the optimization model for time window size can be described as:

$$\max H(n) = - \sum_{i=1}^m p_i(n) \log_2 p_i(n), \quad (2)$$

where n is the time window size, and $p_i(n)$ is the probability of standard deviation values to fall into the range of $(a_i, a_{i+1}]$ when the time window size is n . In this study, the optimal time window size was obtained from an extensive number of samples. The size of the windows ranged from 0.1 to 64 s, and the total number of samples was 200. For a higher classification accuracy, more attention should be paid to small windows. To obtain those samples, logarithm interpolation was used. For all samples, the next sample is always $10^{(\log_{10} 64 - \log_{10} 0.1)/200}$ times the previous one. By calculating the Shannon entropy considering all 200 sizes, we obtained the maximum entropy and its corresponding window size.

4.2. Data Balance Using the Synthetic Minority Oversampling Technique (SMOTE)

The pie chart in Figure 5a shows the proportion of the five categories of vehicle interior noise studied in this work. The most frequent event is 'Broadcast', which accounts for 67.56% of all vehicle

interior noise events. ‘Other noises’ is the next most frequent event, at approximately 22%. ‘Beep’, ‘Squeal’, and ‘Rumble’ represent smaller percentages of the vehicle interior noise events, at 4.99%, 2.79%, and 2.66%, respectively. These results indicate that there is a severe class imbalance, which could significantly undermine most standard classification learning algorithms [41].

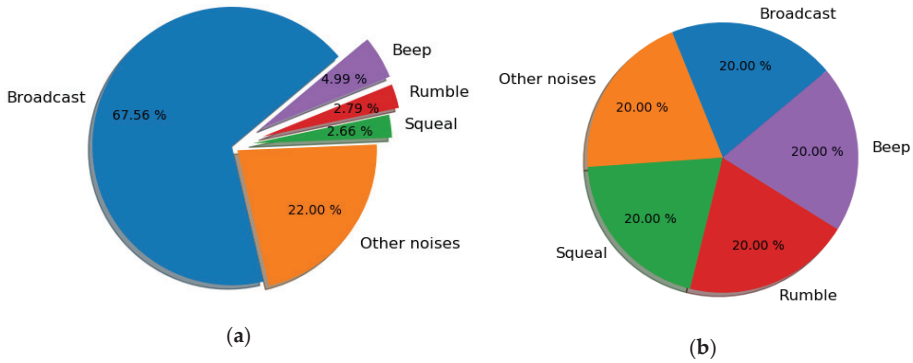


Figure 5. Data (a) before and (b) after synthetic minority oversampling technique (SMOTE) balance.

In this study, we adopted the synthetic minority oversampling technique (SMOTE) to overcome data imbalance. Generally, the class imbalance can be addressed by: (1) synthesizing new minority class instances; (2) oversampling minority class; (3) under-sampling majority class; and (4) tweaking the cost function to enhance the importance of misclassification of minority instances. The SMOTE used in this study utilizes the first solution because increasing the number of minority classes is better than merely duplicating minority classes, which has stronger robustness and generalization ability. This technique returns the original samples and an additional number of synthetic minority class samples. The SMOTE takes samples from the feature space of each minority class and its k nearest neighbors and generates new instances that combine the features of the target classes with the features of their k neighbors. Therefore, it increases the features available for each category and makes the samples more general. In this study, we increased the percentage of ‘Other noises’, ‘Squeal’, ‘Rumble’, and ‘Beep’ to be the same as ‘Broadcast’ via SMOTE when training the multi-classification model, as shown in Figure 5b.

4.3. Features

In ML, features are individual measurable properties of an observed phenomenon [42]. Selecting informative, independent, and discriminating features is a crucial process in classification or regression. The 45 features implied in this study are shown in Table 1. The feature sets include low-level signal properties (f1–f9) and Mel-frequency spectral coefficients (MFCCs) (f10–f45) [27].

Table 1 defines the features of low-level signal properties (f1–f9). N is the sample number of one segment; k refers to the k th sample point; x is the time-series signal; and X denotes the spectrum of Fourier transform (FT); $sign(\cdot)$ is the sign function; TH is the threshold, which takes the value of 0.85 in the definition of f6; $P(k)$, which is shown in the definition of f8, is the probability distribution of the power spectrum $S(k) = |X(k)|^2$. Moreover, MFCCs are features commonly used in speech and speaker recognition [38]. In this study, the first 12 MFCCs coefficients (f10–f21) were used to obtain more information from the audio segments. Because the audio signals vary intermittently, it is necessary to add features related to the change of cepstral characteristics over time [43]. Therefore, the first- and second-order derivatives of the first 12 MFCCs (f22–f33 and f34–f45) were also calculated.

Table 1. Features used in this study.

| Category | Feature | Definition |
|------------------|-------------------------------------|---|
| Time-domain | f1 | Segment energy $f1 = \sum_{k=0}^{N-1} x(k) ^2$ |
| | f2 | Root mean square (RMS) of the segment $f2 = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} x(k)^2}$ |
| | f3 | Zero cross rate $f3 = \frac{1}{2} \sum_{k=0}^{N-1} sign(x(k)) - sign(x(k-1)) $ |
| Frequency-domain | f4 | Spectral centroid $f4 = \sum_{k=0}^{N-1} X(k) \cdot k / \sum_{k=0}^{N-1} X(k) $ |
| | f5 | Spectral bandwidth $f5 = \sqrt{\sum_{k=0}^{N-1} (k - f4)^2}$ [29] |
| | f6 | Spectral roll-off $f6 = \max \left\{ \sum_{k=0}^m X(k) \leq TH \cdot \sum_{k=0}^{N-1} X(k) \right\}$ |
| | f7 | Spectral bandwidth to energy ratio $f7 = f5 / f1$ |
| | f8 | Spectral entropy $f8 = - \sum_{n=1}^N P(k) \log_2 P(k)$ |
| | f9 | Energy to spectral entropy ratio $f9 = f10 / f8$ |
| | f10–f21 | First 12 MFCCs |
| f22–f33 | First-order derivatives of f10–f21 | |
| f34–f45 | Second-order derivatives of f10–f21 | |

4.4. Feature Selection Based on IG

During data analysis, hundreds of features may be generated, many of which are redundant and not relevant to the data mining task. Removing these irrelevant features may waste vast amounts of computation time and influence the prediction results. Although experts in relevant files can select the useful features, this is a challenging and time-consuming task, especially when the characteristics of the dataset are not well known. The goal of feature selection is to find a minimum set of features so that the prediction results are as close as possible to (or better than) the original feature set.

In this study, we employed the IG as an index for feature selection. IG is a feature evaluation method based on entropy and is widely employed in the field of ML [44]. In feature selection, IG is defined as the complete information provided by the features for the classification task. IG measures the importance of features as:

$$IG(S, a) = E(S) - E(S|a), \quad (3)$$

where $IG(S, a)$ is the IG of the original feature set S for feature a ; $E(S)$ is the entropy for the feature set without any change; and $E(S|a)$ is the conditional entropy for the feature set, given feature a . The conditional entropy $E(S|a)$ can be written as:

$$E(S|a) = \sum_{v \in a} \frac{S_a(v)}{S} * E(Sa(v)), \quad (4)$$

where $\frac{S_a(v)}{S}$ is the categorical probability distribution of feature a at $v \in a$, and $E(Sa(v))$ is the entropy of a sample group where a has the value v . The greater the value of $IG(S, a)$, the more critical is a for the classification model.

4.5. Multi-Classification Model for Vehicle Interior Noise Based on XGBoost

XGBoost was designed based on gradient boosted decision trees [45]. We chose XGBoost due to its computation speed and model performance, which have been verified by a previous study [22]. As an ensemble model of decision trees, the definition of the XGBoost model can be written as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad (5)$$

where K is the total number of decision trees, f_k is the k th decision tree, and \hat{y}_i is the prediction result of sample x_i . The cost function with a regularization term is given by [45]:

$$L(f) = \sum_{i=1}^n l(\hat{y}_i, y_i) + \sum_{k=1}^K \Omega(f_k), \quad (6)$$

with:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2, \quad (7)$$

where T is the number of leaves of the classification tree f , and w is the score of each leaf. The Lasso regulation of coefficient γ and ridge regularization of coefficient λ can work together to control the complexity of the model. By expressing the objective function as a second-order Taylor expansion, the objective function at step t can be written as [46]:

$$L(f) \approx \sum_{i=1}^n \left[l(\hat{y}_i, y_i) + g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i) \right] + \Omega(f_t), \quad (8)$$

where $g_i = \partial_y l(\hat{y}_i, y_i)$, and $g_i = \partial_y^2 l(\hat{y}_i, y_i)$. By removing the constant term, the approximation of the objective at step t is available:

$$\hat{L}(f) = \sum_{i=1}^n \left[g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i) \right] + \Omega(f_t). \quad (9)$$

By expanding the regularization term Ω and defining I_j as the instance set at leaf j , Equation (9) can be rewritten as [47]:

$$\hat{L}(f) = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T. \quad (10)$$

By rewriting the objective function as a unary quadratic function of leaf score w , the optimal w and the value of the objective function are easily obtained. In XGBoost, the gain is used for splitting decision trees:

$$G_j = \sum_{i \in I_j} g_i, \quad (11)$$

$$H_j = \sum_{i \in I_j} h_i, \quad (12)$$

$$\text{gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma, \quad (13)$$

where the first and second terms are the score of the left and right child tree, respectively; the third term is the score if there is no splitting; and γ is the complexity cost when a new split is added. Despite

the serial relationship between the adjacent trees, the node in a certain level can be parallel during the splitting, which enables XGBoost to have a faster train speed.

5. Results and Discussions

In general, the parameters of an ML model can significantly impact its performance, and XGBoost is no exception. Through extensive testing and observation, we set the critical parameters of this model as follows: maximum depth of the tree (`max_depth`) = 6; learning rate (`eta`) = 0.01; minimum sum of instance weight needed in a child (`min_child_weight`) = 1; subsample ratio of the training instance (`subsample`) = 1; fraction of features (columns) to use (`colsample_bytree`) = 1. The ratio between the training dataset and the test dataset was set to 0.8/0.2 in this study.

5.1. Optimal Time Window Size and Data Balance

We divided the audio signals collected from the test line into segment sequences with different time windows. Figure 6 presents the calculated Shannon entropies under different time window sizes. The Shannon entropy maintains a relatively stable state when the time window size increases from 0.1 (10^{-1}) to 1.58 ($10^{0.2}$) s, after which it decreases dramatically. When the time window size is 1.58 s, the Shannon entropy reached its maximum value. According to the maximum Shannon entropy hypothesis, the optimal time window size is 1.58 s. However, we maintained a relatively small window in our study to avoid a situation where one window contains different vehicle interior noise events. Therefore, we set the time window size to 1 s.

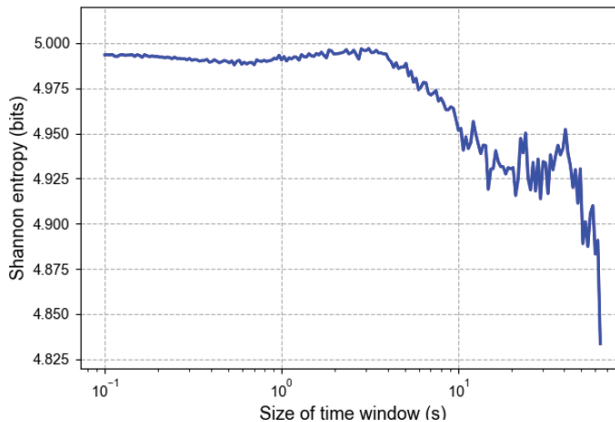


Figure 6. Entropy at different time window sizes.

We increased the proportion of four minority classes to the same as ‘Broadcast’ with SMOTE. The performance of the multi-classification model using balanced or unbalanced training data was compared. Table 2 reports the comparison results from the perspective of precision, recall, and F1 score. ‘Support’ in this table means the total number of occurrences in each category. Data balance increased the precision of ‘Broadcast’ and decreased its recall. In contrast, it decreased the precision and increased the recall of minority classes, namely ‘Beep’, ‘Rumble’, ‘Squeal’, and ‘Other noises’. Meanwhile, F1 scores presented a slight drop after the data balance, except for the classes of ‘Beep’ and ‘Squeal’.

Table 2. Classification reports of test results.

| Classes | The Model Trained with Unbalanced Data | | | The Model Trained with Balanced Training Data | | | Support |
|--------------|--|--------|----------|---|--------|----------|---------|
| | Precision | Recall | F1 score | Precision | Recall | F1 score | |
| Other noises | 0.94 | 0.94 | 0.94 | 0.87 | 0.95 | 0.91 | 3671 |
| Broadcast | 0.96 | 0.98 | 0.97 | 0.98 | 0.92 | 0.95 | 11,274 |
| Squeal | 0.95 | 0.97 | 0.96 | 0.86 | 1.00 | 0.92 | 444 |
| Rumble | 0.95 | 0.89 | 0.92 | 0.82 | 0.97 | 0.89 | 466 |
| Beep | 0.95 | 0.73 | 0.83 | 0.70 | 0.87 | 0.78 | 834 |

We also employed confusion matrices to describe the performance before and after the training data were balanced, as shown in Figure 7. These matrices provide insights into the errors by the classification model and distinguish the types of errors. For instance, the matrices imply that ‘Squeal’ is commonly mislabeled as ‘Broadcast’, and ‘Rumble’ is mislabeled as ‘Other noises’. One can also notice that the data balance improves the identification of the performance of minority classes such as ‘Beep’, ‘Rumble’, and ‘Squeal’. ‘Squeal’ and ‘Rumble’ have a strong relationship with vehicle-track conditions, which is a major concern in our research. It is therefore desirable to detect all ‘Squeal’ and ‘Rumble’ events. Therefore, we balanced the training dataset via SMOTE to improve the recall of ‘Squeal’ and ‘Rumble’, despite the slight decrease in precision.

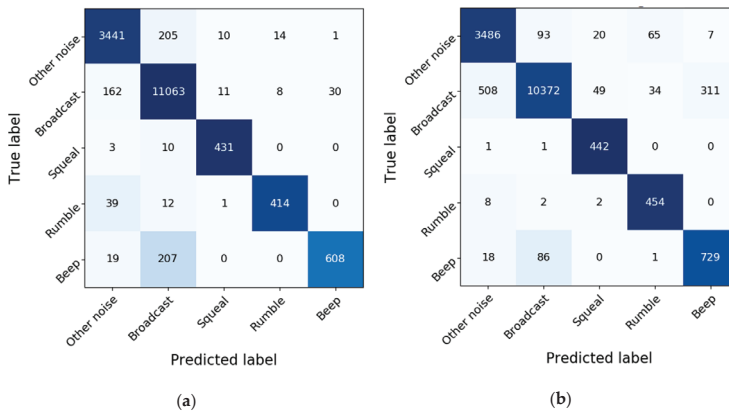


Figure 7. Confusion matrices of test results: (a) Model trained with unbalanced data; (b) Model trained with balanced data.

5.2. Feature Selection Based on the Importance Score

The importance was calculated explicitly for each feature by using the inbuilt feature importance property of XGBoost algorithm. The scores for features indicate how useful they were in the construction of the model and allows features to be ranked and compared with each other. Besides, a mutual information-based feature selection method is also used to verify the results of the importance-based method. In contrast to the importance score, the calculation of mutual information does not depend on the classifiers, but only considers the statistical characteristics of the input features and target variables.

In our classification model, 45 initial features were considered. Figure 8a shows the feature importance scores calculated by gain [45]. The importance scores of different features vary greatly, ranging from 0 to 378. The spectral centroid, denoted as f4, ranks first. In contrast, the importance score of f2, root mean square (RMS) of segments, equals zero, which means that it was not used during the training process. Figure 8a also shows that the low-order features and first 12 MFCCs are essential in the classification task. The results of the feature importance analysis indicate that the

contribution of different features to the model varies greatly. Thus, feature selection is necessary to improve the performance of the model and speed of calculations. Figure 8c shows the results for 45 features calculated by the mutual information-based method. The mutual information of these features has a similar trend with that of importance score. However, the importance scores of some features are very different from their mutual information value. For example, the importance score of feature f2 is 0, but its mutual information ranks fifth among all of the 45 features. The reason is that the mutual information only considering the features and target variables cannot reflect whether the features were engaged in the establishment of the classification model.

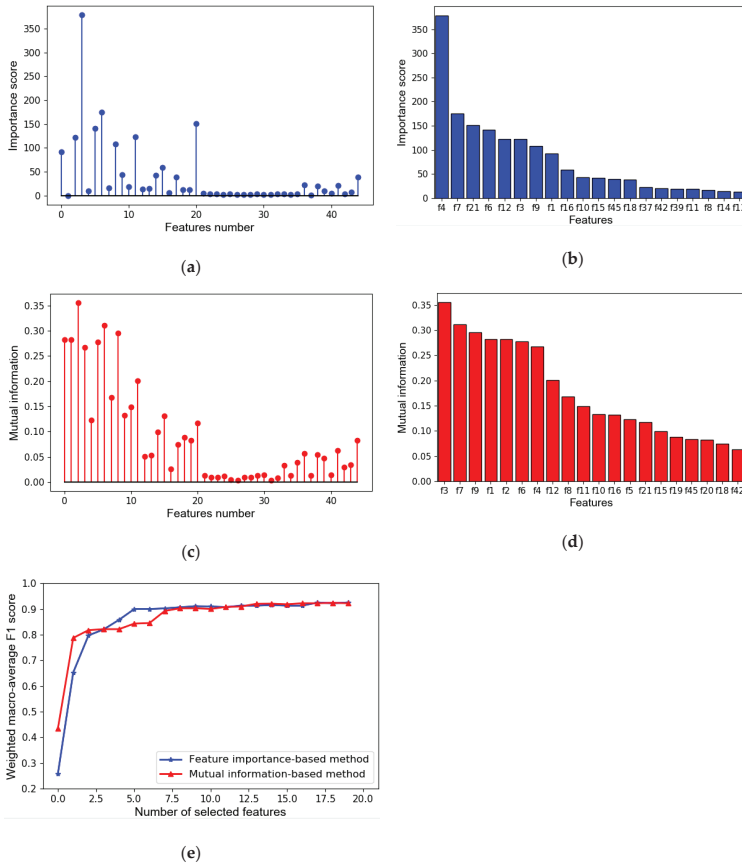


Figure 8. Illustration of feature selection based on different methods: (a) importance score of all the features; (b) importance score of the top 20 features; (c) mutual information of all the features; (d) mutual information of the top 20 features; (e) comparison of results of the two feature selection methods.

First, all 45 features were sorted in descending order of importance and mutual information, respectively. Figure 8b,d show the histograms of the top 20 features in descending order of the importance score and mutual information independently. We then constructed 20 feature sets incrementally with top 1, top 2, . . . , and top 20 features. Furthermore, the classification results with different features sets were compared, as shown in Figure 8e. There, the weighted macro average F1

score, $F1_{wm}$, was used to evaluate the performance of the multi-classification model, and it can be defined as follow:

$$F1_{wm} = \frac{\sum_{i=1}^N F1_i \times w_i}{N}, \quad (14)$$

where N is the total number of classes, in this study $N = 5$; $F1_i$ is the F1 score of the i th class; and w_i is the weight of the i th class and there is $\sum_{i=1}^N w_i = N$. Because this study mainly focuses on ‘Squeal’ and ‘Rumble’ we set both their weights to 1.3, and the weights of ‘Other noises’, ‘Beep’, and ‘Broadcast’, to 0.8. The value of $F1_{wm}$ varies from 0 to 1. The closer the weight is to 1, the better the model performs. The red line in Figure 8e corresponds to the classification results of 20 feature sets constructed by the mutual information-based feature selection method, and the blue line corresponds to that by the feature importance-based method. The results in Figure 8e show that $F1_{wm}$ by both feature selection methods increased rapidly when the feature set expanded from the top 1 to the top 8 features. Afterward, $F1_{wm}$ remained stable. The comparison of the results of the two methods indicates that the mutual information-based method performed better than the importance-based one when the number of selected features was less than 4. However, when the feature set expanded from the top 4 to the top 11, the importance-based method performed better. Then, the continuous increase in the number of the features selected causes no obvious difference between the performances of the two methods. According to the analysis, the set with the top 10 features selected by the importance-based method was employed in this study, the $F1_{wm}$ of which reached 0.91.

5.3. Comparisons with Other Methods

To validate the performance and execution speed of the XGBoost-based classifier used in our study, we conducted a comparison with other commonly used classifiers, including the K-nearest neighbors, decision trees, random forest, gradient boost, extra trees, AdaBoost, and artificial neural network (ANN) classifiers. This study ran all classifiers on the same computer and with the same training and testing data set. Table 3 shows the comparison results of $F1_{wm}$ and running time. The $F1_{wm}$ value of the gradient boost ranked first at 0.925. However, training and testing the gradient boost classifier also consumed the longest running time, 340.31 s, which was approximately 22 times longer than the time needed by the XGBoost classifier. In contrast, the K-nearest Neighbors presented the fastest computing speed and one of the lowest $F1_{wm}$. Besides, the accuracy and precision of different models are provided in Table 3. The accuracy and precision share a similar trend with $F1_{wm}$. The comparison with other classifiers depicts that the XGBoost model shows a good performance in accuracy and execution speed.

Table 3. Comparisons between XGBoost and other classifiers.

| Classifier | $F1_{wm}$ | Accuracy | Precision | Running Time (s) |
|----------------------|--------------|----------|-----------|------------------|
| XGBoost | 0.923 | 0.96 | 0.95 | 15.06 |
| K-nearest Neighbours | 0.704 | 0.84 | 0.72 | 2.51 |
| Decision Trees | 0.851 | 0.91 | 0.92 | 3.12 |
| Random Forest | 0.923 | 0.96 | 0.94 | 77.88 |
| Gradient Boost | 0.925 | 0.96 | 0.94 | 340.31 |
| AdaBoost | 0.651 | 0.77 | 0.64 | 67.70 |
| ANN | 0.880 | 0.93 | 0.94 | 173.22 |

5.4. Case Studies to Extend the Model Application Scenarios

In this paper, we provided two case studies to extend the application scenarios. First, we conducted a statistical analysis to investigate the relationship between the vehicle interior noises and the dynamic responses of the car body with multi-source data collected by smartphones. After that, we used the proposed multi-classification model to detect abnormal interior noise events and evaluate the effect of

rail grinding for guiding the implementation of maintenance work. Figure 9 illustrates the schematics of both case studies in this work.

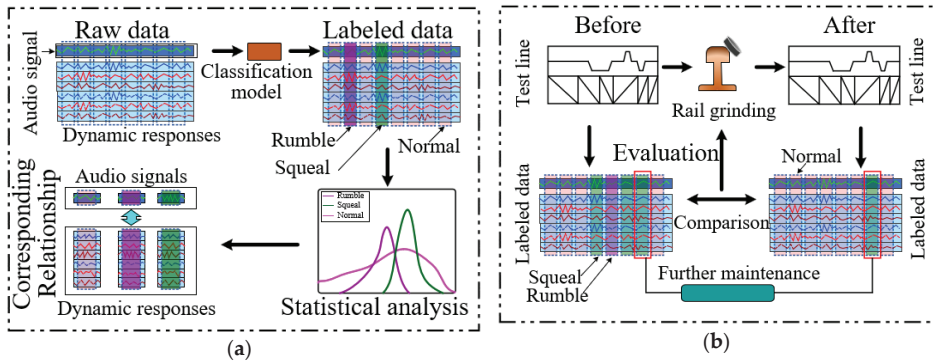


Figure 9. Schematics for case studies: (a) statistical analysis of vehicle interior noise and dynamic responses; (b) abnormal events detection and rail grinding effect evaluation using the XGBoost multi-classification model.

In the first case study, about 10 h of onboard monitoring data collected by smartphones were used. As shown in Figure 9a, the audio signals of the vehicle interior noise were fed into the multi-classification model established in this work. According to the classification results, the raw data were labeled into three categorizations: ‘Squeal’, ‘Rumble’, and ‘Normal’. ‘Normal’ contained all other events except for ‘Squeal’ and ‘Rumble’ events. Then, statistical analyses for the dynamic responses corresponding to different vehicle interior noise were performed. This case study aimed to investigate the causes of the abnormal noise events and find out the solutions through the statistical analysis results.

For ‘Squeal’, ‘Rumble’, and ‘Normal’, the probability distribution curves of running speed (v) and vertical acceleration (a_v) of the car body are presented in Figure 9a,b, respectively. The vehicle speed v used here was not measured directly but obtained by the first-order integration of the longitudinal acceleration a_l [47], which can be written as follows:

$$v = \int_0^t a_l dt + v_0, \quad (15)$$

where t denotes the time; v_0 is the initial velocity. Since the integration begins when the subway train starts, v_0 equals to 0. The probability distribution curves in Figure 10a shows that ‘Squeal’ usually occurs at higher running speed compared with ‘Normal’ and ‘Rumble’. This also suggests that we can reduce the occurrence of ‘Squeal’ by adjusting the operating speed of the train. In contrast, ‘Rumble’ occurs at a slower speed and higher vertical vibration level compared to ‘Squeal’, as shown in Figure 10b. This phenomenon implies that the occurrence of ‘Rumble’ is related to the resonance of the car body, which may be avoided by optimizing the structure of the car body.

The schematic of the second case study is presented in Figure 9b. The test interval selected in this study was between two adjacent stations with a length of 1631 m. The track alignment of the test interval is presented in the upper plot of Figure 11a. There are three curves in the test interval, the radii of which are 1200 m, 800 m, and 800 m. This case study aimed to test the capacity of this model for identifying abnormal noise events, evaluating the effect of rail grinding, and providing information relevant to designing a future maintenance plan.

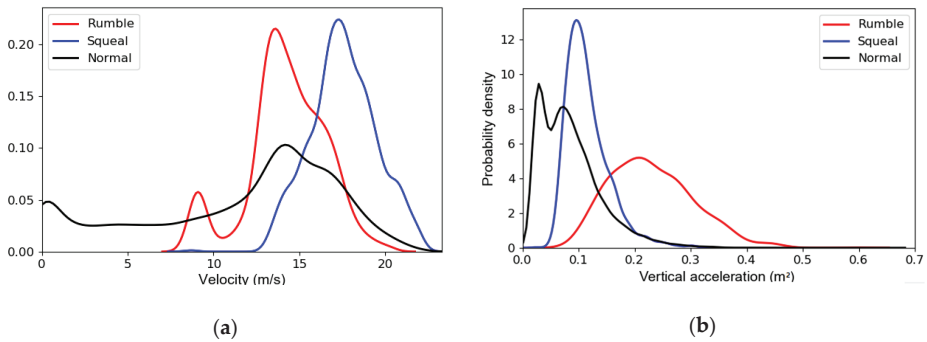


Figure 10. Statistical analysis of vehicle interior noise and dynamic responses: (a) The probability distribution curves of running speed (v); (b) The probability distribution curves of vertical acceleration (a_v).

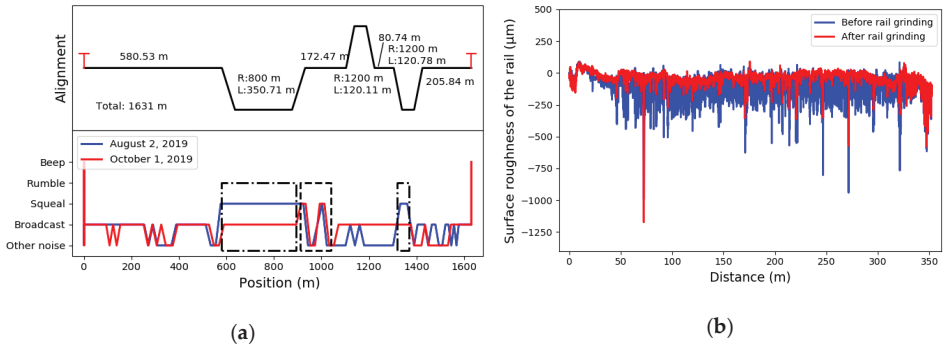


Figure 11. Abnormal events detection and rail grinding effect evaluation using the XGBoost multi-classification model: (a) track alignments of the test section and the identification results before and after rail grinding; (b) the surface roughness of the rail before and after rail grinding.

The authors first collected multi-source data with the onboard smartphone on 2 August 2019. The results of the multi-classification model are depicted in the lower plot of Figure 11a with a blue line. The results indicate that ‘Squeal’ occurred in the positions from 580 to 890 m, 910 to 1040 m, and 1320 to 1370 m. It can be seen that the figure the sections where ‘Squeal’ occurs have a high overlap ratio with the curve sections, especially the curve section with a radius of 800 m. According to the classification results and design information, we can make a preliminary conclusion that the sharp curves are the main causes of ‘Squeal’. The results also indicate the need for rail grinding or other corresponding maintenance measures.

Then, a scheduled rail grinding of the test interval was done on 21 August 2019. The surface roughness of the rail before and after rail grinding presented in Figure 11b indicates that rail grinding reduced the roughness of the rail surface effectively. Since reducing the rail roughness, that is, the unevenness on the tread of the rail benefits improving the rail-wheel contact relationship, rail grinding is a common measure for eliminating the abnormal noise and vibration of subway trains.

Another onboard test was conducted on 1 October 2019, to verify the effects of the maintenance work. The corresponding classification results after the rail grinding are displayed in red in the lower plot of Figure 11a. It can be seen that after rail grinding, the ‘Squeal’ was eliminated at 580–890 m and 1320–1370 m. However, the ‘Squeal’ at 910–1040 m remained. The results illustrate that rail grinding eliminated ‘Squeal’ at circular curves effectively. Nevertheless, it showed no apparent effect on the occurrences at transition curves and straight-line sections, which shows that there exist some other

factors that lead to ‘Squeal’ in these sections. Thus, future maintenance work should focus on the section from 910 to 1040 m. This case study demonstrates the potential of applying the proposed multi-classification model in evaluating the effect of rail grinding and providing more information about the track conditions to making a further rail maintenance plan.

6. Conclusions

This study proposed a vehicle interior noise multi-classification model based on the XGBoost method and onboard smartphone data. By considering the Shannon entropy, a 1-second time window was selected to perform the data segmentation task. The comparison between the performances before and after the training data was balanced demonstrated that data balancing can promote the recall of minority classes but decrease the precision of their results. Feature importance analysis results show that features calculated from the spectrum of the Fourier transform and the first 12 MFCCs are the most essential among all features. By comparing and analyzing the results of importance-based and mutual information-based methods, this study selected the top 10 features in importance score to form the features set, whose $F1_{wm}$ reached 0.91. Then, the comparison between the XGBoost and other commonly used classifiers showed that the proposed XGBoost-based classification model presents a faster computing speed while maintaining a good performance. The case studies verified that the proposed multi-classification model has the potential to investigate the correlation between abnormal vehicle interior noise and dynamic responses of the train. Moreover, the capacity of the model to monitor abnormal noise events and evaluate the effect of rail grinding was also proved.

There are a few directions for future research. A more detailed classification of vehicle interior noise could be developed based on specific track-vehicle conditions so that this model would be suitable for general cases. Furthermore, more experiments are needed to explain the performance among different vehicles and track slabs. Another interesting option is to investigate the relationship between abnormal noise and wheel-rail contact conditions. Furthermore, the authors intend to set up a data collection system with high-quality sensors for more accurate and reliable data.

Author Contributions: Conceptualization, P.W. and Q.H.; data curation, Y.W. and Q.H.; formal analysis, Y.W.; methodology, Y.W. and Q.W.; validation, Y.W. and Z.C.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W. and Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 51878576 and U1934214, and China Scholarship Council, file No. 201907000077.

Acknowledgments: The authors would like to thank Huajiang Ouyang, from the University of Liverpool, for his support when this study was being finished.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. China Urban Rail Transit Association. *Urban Rail Transit 2018 Annual Statistical Report*; China Urban Rail Transit Association: Beijing, China, 2019.
2. Atmaja, B.; Puabdillah, M.; Farid, M.; Asmoro, W. Prediction and simulation of internal train noise resulted by different speed and air conditioning unit. *J. Phys. Conf. Ser.* **2018**, *1075*, 012038. [[CrossRef](#)]
3. Zhang, J.; Xiao, X.; Sheng, X.; Li, Z.; Jin, X. A Systematic Approach to Identify Sources of Abnormal Interior Noise for a High-Speed Train. *Shock Vib.* **2018**, *2018*. [[CrossRef](#)]
4. Talotte, C. Aerodynamic noise: A critical survey. *J. Sound Vib.* **2000**, *231*, 549–562. [[CrossRef](#)]
5. Han, J.; Xiao, X.; Wu, Y.; Wen, Z.; Zhao, G. Effect of rail corrugation on metro interior noise and its control. *Appl. Acoust.* **2018**, *130*, 63–70. [[CrossRef](#)]
6. Wu, B.; Chen, G.; Lv, J.; Zhu, Q.; Kang, X. Generation mechanism and remedy method of rail corrugation at a sharp curved metro track with Vanguard fasteners. *J. Low Freq. Noise Vib. Act. Control.* **2019**. [[CrossRef](#)]
7. Li, L.; Thompson, D.; Xie, Y.; Zhu, Q.; Luo, Y.; Lei, Z. Influence of rail fastener stiffness on railway vehicle interior noise. *Appl. Acoust.* **2019**, *145*, 69–81. [[CrossRef](#)]

8. Meehan, P.A.; Liu, X. Modelling and mitigation of wheel squeal noise amplitude. *J. Sound Vib.* **2018**, *413*, 144–158. [[CrossRef](#)]
9. Zhang, J.; Han, G.; Xiao, X.; Wang, R.; Zhao, Y.; Jin, X. Influence of Wheel Polygonal Wear on Interior Noise of High-Speed Trains. In *China's High-Speed Rail Technology*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 373–401.
10. Fink, O.; Zio, E.; Weidmann, U. Predicting time series of railway speed restrictions with time-dependent machine learning techniques. *Expert Syst. Appl.* **2013**, *40*, 6033–6040. [[CrossRef](#)]
11. Sun, Y.; Zhao, Y. Characteristics of Interior Noise in MonoRail and Noise Control. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*; Institute of Noise Control Engineering: Chicago, IL, USA, 2018; Volume 258, pp. 1461–1467.
12. Hu, K.; Wang, Y.; Guo, H.; Chen, H. Sound quality evaluation and optimization for interior noise of rail vehicle. *Adv. Mech. Eng.* **2014**, *6*, 820875. [[CrossRef](#)]
13. Kurzweil, L.G. Prediction and control of noise from railway bridges and tracked transit elevated structures. *J. Sound Vib.* **1977**, *51*, 419–439. [[CrossRef](#)]
14. Zhang, J.; Xiao, X.; Sheng, X.; Li, Z. Sound Source Localisation for a High-Speed Train and Its Transfer Path to Interior Noise. *Chin. J. Mech. Eng.* **2019**, *32*, 59. [[CrossRef](#)]
15. Zhang, J.; Xiao, X.; Sheng, X.; Zhang, C.; Wang, R.; Jin, X. SEA and contribution analysis for interior noise of a high speed train. *Appl. Acoust.* **2016**, *112*, 158–170. [[CrossRef](#)]
16. Franzoni, L.; Rouse, J.; Duvall, T. A broadband energy-based boundary element method for predicting vehicle interior noise. *J. Acoust. Soc. Am.* **2004**, *115*, 2538. [[CrossRef](#)]
17. Wu, D.; Ge, J.M. Analysis of the Influence of Racks on High Speed Train Interior Noise Using Finite Element Method. *Appl. Mech. Mater.* **2014**, *675*, 257–260. [[CrossRef](#)]
18. Ghofrani, F.; He, Q.; Goverde, R.M.; Liu, X. Recent applications of big data analytics in railway transportation systems: A survey. *Transp. Res. Part. C Emerg. Technol.* **2018**, *90*, 226–246. [[CrossRef](#)]
19. Toque, F.; Come, E.; Oukhellou, L.; Trepanier, M. Short-Term Multi-Step Ahead Forecasting of Railway Passenger Flows During Special Events With Machine Learning Methods. In Proceedings of the CASPT 2018, Conference on Advanced Systems in Public Transport and TransitData 2018, Brisbane, Australia, 23–25 July 2018; p. 15.
20. Cui, Y.; Martin, U.; Zhao, W. Calibration of disturbance parameters in railway operational simulation based on reinforcement learning. *J. Rail Transp. Plan. Manag.* **2016**, *6*, 1–12. [[CrossRef](#)]
21. Ghofrani, F.; Pathak, A.; Mohammadi, R.; Aref, A.; He, Q. Predicting rail defect frequency: An integrated approach using fatigue modeling and data analytics. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 101–115. [[CrossRef](#)]
22. Mohammadi, R.; He, Q.; Ghofrani, F.; Pathak, A.; Aref, A. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 153–172. [[CrossRef](#)]
23. Ghofrani, F.; He, Q.; Mohammadi, R.; Pathak, A.; Aref, A. Bayesian Survival Approach to Analyzing the Risk of Recurrent Rail Defects. *Transp. Res. Rec.* **2019**, *2673*, 281–293. [[CrossRef](#)]
24. Li, H.; Parikh, D.; He, Q.; Qian, B.; Li, Z.; Fang, D.; Hampapur, A. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transp. Res. Part C Emerg. Technol.* **2014**, *45*, 17–26. [[CrossRef](#)]
25. He, Q.; Li, H.; Bhattacharjya, D.; Parikh, D.P.; Hampapur, A. Track geometry defect rectification based on track deterioration modelling and derailment risk assessment. *J. Oper. Res. Soc.* **2015**, *66*, 392–404. [[CrossRef](#)]
26. Li, Z.; He, Q. Prediction of railcar remaining useful life by multiple data source fusion. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2226–2235. [[CrossRef](#)]
27. Verhaegh, W.; Verhaegh, W.; Aarts, E.; Korst, J. *Algorithms in Ambient Intelligence*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2004; Volume 2.
28. Mato-Méndez, F.J.; Sobreira-Seoane, M.A. Blind separation to improve classification of traffic noise. *Appl. Acoust.* **2011**, *72*, 590–598. [[CrossRef](#)]
29. Sobreira-Seoane, M.A.; Rodriguez Molaes, A.; Alba Castro, J.L. Automatic classification of traffic noise. *J. Acoust. Soc. Am.* **2008**, *123*, 3823. [[CrossRef](#)]
30. Paulraj, P.; Melvin, A.A.; Sazali, Y. Car Cabin Interior Noise Classification Using Temporal Composite Features and Probabilistic Neural Network Model. *Appl. Mech. Mater.* **2014**, *471*, 64–68. [[CrossRef](#)]
31. Wang, Y.; Wang, P.; Wang, X.; Liu, X. Position synchronization for track geometry inspection data via big-data fusion and incremental learning. *Transp. Res. Part C Emerg. Technol.* **2018**, *93*, 544–565. [[CrossRef](#)]

32. Cho, C.J.; Park, Y.; Ku, B.; Ko, H. An implementation of environment recognition for enhancement of advanced video based railway inspection car detection modules. *Sci. Adv. Mater.* **2018**, *10*, 496–500. [[CrossRef](#)]
33. Yin, J.; Zhao, W. Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach. *Eng. Appl. Artif. Intell.* **2016**, *56*, 250–259. [[CrossRef](#)]
34. Li, C.; Luo, S.; Cole, C.; Spiriyagin, M. An overview: Modern techniques for railway vehicle on-board health monitoring systems. *Veh. Syst. Dyn.* **2017**, *55*, 1045–1070. [[CrossRef](#)]
35. Tsunashima, H.; Naganuma, Y.; Matsumoto, A.; Mizuma, T.; Mori, H. Condition monitoring of railway track using in-service vehicle. *Reliab. Saf. Railw.* **2012**, *12*, 334–356.
36. Wang, P.; Wang, Y.; Wang, L.; Chen, R.; Xiao, J. Measurement of Carbody Vibration in Urban Rail Transit Using Smartphones. In Proceedings of the Transportation Research Board 96th Annual Meeting, Washington, DC, USA, 8–12 January 2017.
37. Ghose, A.; Biswas, P.; Bhaumik, C.; Sharma, M.; Pal, A.; Jha, A. Road condition monitoring and alert application: Using in-vehicle Smartphone as Internet-connected sensor. In Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19–23 March 2012; pp. 489–491.
38. Han, W.; Chan, C.F.; Choy, C.S.; Pun, K.P. An efficient MFCC extraction method in speech recognition. In Proceedings of the 2006 IEEE International Symposium on Circuits and Systems, Island of Kos, Greece, 21–24 May 2006.
39. Banos, O.; Galvez, J.-M.; Damas, M.; Pomares, H.; Rojas, I. Window Size Impact in Human Activity Recognition. *Sensors* **2014**, *14*, 6474–6499. [[CrossRef](#)]
40. Zhang, X.; Feng, N.; Wang, Y.; Shen, Y. Acoustic emission detection of rail defect based on wavelet transform and Shannon entropy. *J. Sound Vib.* **2015**, *339*, 419–432. [[CrossRef](#)]
41. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [[CrossRef](#)]
42. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer Science + Business Media: Berlin/Heidelberg, Germany, 2006.
43. Martinez, J.; Perez, H.; Escamilla, E.; Suzuki, M.M. Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques. In Proceedings of the CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers, Puebla, Mexico, 27–29 February 2012; pp. 248–251.
44. Lei, S. A feature selection method based on information gain and genetic algorithm. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 2, pp. 355–358.
45. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
46. Omar, K. XGBoost and LGBM for Porto Seguro’s Kaggle Challenge: A Comparison. 2018. Available online: <https://pub.tik.ee.ethz.ch/students/2017-HS/SA-2017-98.pdf> (accessed on 7 July 2019).
47. Wang, Y.; Cong, J.; Tang, H.; Liu, X.; Gao, T.; Wang, P. A Data Fusion Approach for Speed Estimation and Location Calibration of a Metro Train. in Underground Environment Based on Low-Cost Sensors in Smartphones. In Proceedings of the Transportation Research Board 98th Annual Meeting, Washington, DC, USA, 13–17 January 2019.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Short Term Traffic State Prediction via Hyperparameter Optimization Based Classifiers

Muhammad Zahid ¹, Yangzhou Chen ^{2,*}, Arshad Jamal ³ and Muhammad Qasim Memon ⁴

¹ College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China; zahid@emails.bjut.edu.cn

² College of Artificial Intelligence and Automation, Beijing University of Technology, Beijing 100124, China

³ Department of Civil and Environmental Engineering, King Fahd University of Petroleum & Minerals, KFUPM Box 5055, Dhahran 31261, Saudi Arabia; arhad.jamal@kfupm.edu.sa

⁴ Advanced Innovation Center for Future education, Faculty of Education, Beijing Normal University (BNU), Beijing 100875, China; memon_kasim@bnu.edu.cn

* Correspondence: yzchen@bjut.edu.cn; Tel.: +86-10-6739-1632

Received: 9 January 2020; Accepted: 23 January 2020; Published: 27 January 2020

Abstract: Short-term traffic state prediction has become an integral component of an advanced traveler information system (ATIS) in intelligent transportation systems (ITS). Accurate modeling and short-term traffic prediction are quite challenging due to its intricate characteristics, stochastic, and dynamic traffic processes. Existing works in this area follow different modeling approaches that are focused to fit speed, density, or the volume data. However, the accuracy of such modeling approaches has been frequently questioned, thereby traffic state prediction over the short-term from such methods inflicts an overfitting issue. We address this issue to accurately model short-term future traffic state prediction using state-of-the-art models via hyperparameter optimization. To do so, we focused on different machine learning classifiers such as local deep support vector machine (LD-SVM), decision jungles, multi-layers perceptron (MLP), and CN2 rule induction. Moreover, traffic states are evaluated using traffic attributes such as level of service (LOS) horizons and simple if–then rules at different time intervals. Our findings show that hyperparameter optimization via random sweep yielded superior results. The overall prediction performances obtained an average improvement by over 95%, such that the decision jungle and LD-SVM achieved an accuracy of 0.982 and 0.975, respectively. The experimental results show the robustness and superior performances of decision jungles (DJ) over other methods.

Keywords: traffic state prediction; spatio-temporal traffic modeling; simulation; machine learning; hyper parameter optimization; ITS

1. Introduction

Smart cities have emerged at the heart of “next stage urbanization” as they are equipped with fully digital infrastructure and communication technologies to facilitate efficient urban mobility. The fundamental enabler of a smart city is dependent on connected devices, though the real concern is how the collected data are distributed city-wide through sensor technologies via the Internet of Things (IoT). Heterogeneous vehicular networks in a connected infrastructure network are able to sense, compute, and communicate information through various access technologies: Universal Mobile Telecommunications System (UTMS), Fourth Generation (4G), and Dedicated Short-Range Communications (DSRC) [1,2]. In vehicular sensor networks (VSN) and Internet of vehicles (IOV), each vehicle act as receivers, senders, and routers simultaneously to transmit data over the network or to a central transportation agency as an integral part of intelligent transportation systems (ITS) [3,4]. Furthermore, each and every network node in VSN is assumed to store, carry, and precisely transfer the

data with cooperative behavior. In recent years, following rapid diversification, navigation technologies and traffic information services enable a large amount of data to be collected from the different devices such as loop detectors, on-board equipment, speed sensors, remote microwave traffic sensors (RTMS), and road-side surveillance cameras etc., that have been proactively used for monitoring of traffic conditions in the ITS domain [5–9]. Sensor networks in the form of road side units (RSUs) offer numerous applications including broadcasting periodic informatory, warnings, and safety messages to road users. The data obtained from these different sources have provided myriad opportunities to estimate and predict travel time and future traffic states through a large number of data-driven computational and machine learning approaches. Accurate traffic state prediction (TSP) ensures efficient vehicle route planning, and pro-active real-time traffic management.

TSP is achieved in three distinct steps: (i) prediction of the desired traffic flow parameters (i.e., volume, speed, and occupancy); (ii) identification of traffic state; and (iii) realizing the traffic state output. TSP can be classified as either short-term prediction or long-term prediction. In the former prediction type, short-term changes in traffic status are predicted (e.g., during a 5, 10, 15, or 30 min prediction horizon), and long-term prediction is usually estimated in days and months [10]. Short-term predictions can either be used directly by traffic professionals to take appropriate actions or can be added as inputs for proactive solutions in congestion management. Short-term prediction reduces common problems such as traffic congestion, road accidents, and air pollution; meanwhile, it also offers road users and traffic management agencies with important information to assist in better decision-making [11]. Three factors affect the quality of prediction in real-time traffic information. These factors include: (i) variation in data collected from various sources like sensors and other sources; (ii) dynamic nature of traffic conditions; and (iii) randomness and stochastic nature of traffic appearing in the supply and demand. However, addressing these factors remained challenging and significant in the realm of quality prediction for real-time traffic information [12].

In TSP, prediction methodologies are broadly studied into two main categories: parametric and non-parametric techniques [8]. Parametric methods include auto aggressive integrated moving average method ARIMA [13], exponential smoothing (ES) [14], and seasonal auto aggressive integrated moving average method (SARIMA) [15,16]. In their study, Li et al. suggested that a multi-view learning approach estimates the missing values in traffic-related time series data [17]. Parametric methods focus on pre-determining the structure of the model based on theoretical or physical assumptions, later tuning a set of parameters that represent the traffic conditions (i.e., a trend in the actual world) [10,11]. These practices develop a mathematical function between historical and predicted states, for instance, model-based time series such as ARIMA, which is commonly used for traffic predictions in all parametric methods [18]. However, autoregressive models provide better accuracy for TSP models, while considering the traffic information about upstream and downstream locations is accounted for on freeways [9]. Parametric methods have good accuracy and high computational efficiency and are highly suited for linear or stationary time-series [19]. On the other hand, non-parametric approaches provide several advantages such as the ability to avoid model's strong assumptions and learn from the implicit dynamic traffic characteristics through archived traffic data. These models have the benefit of being able to manage non-linear, dynamic tasks, and can also utilize spatial-temporal relationships, whereas non-parametric methods require a large amount of historical data and training processes. Non-parametric techniques include artificial neural network (ANN) [20–22]; support vector regression (SVR) [23,24]; K-nearest neighbor (KNN) [25–29]; and Bayesian models [30,31]. Since non-parametric techniques yield better prediction accuracy compared to ordinary parametric techniques like time series as they require significantly high computational effort. Their prediction accuracy is largely dependent on the quantity and quality of training data [32]. The above-mentioned methods have been successfully deployed in various transport related applications where predictions are required for excessive passenger flow at a metro station or in a crowd gathered for a special event [33,34].

A critical review of literature for TSP indicates that time series and conventional ANN models have been widely employed for short term TSP. Although these models were aimed to fit the speed,

density, or volume data as they usually inherit an overfitting issue. Thereby, the ability of models that capture generalized trends for traffic prediction is compromised. Macroscopic traffic parameters such as traffic flow, traffic speed, and density are the state variables of interest used in TSP, and are subsequently evaluated using level of service (LOS). However, training and testing the accuracy for the majority of such modeling approaches is frequently questioned. To overcome this issue, we incorporated recent AI and machine learning state-of-art-approaches such as decision jungles and LD-SVM (via hyperparameter optimization) as these methods have been rarely been explored in the existing works. Data utilized in current study was extracted from traffic simulator 'VISSIM', which realistically simulates complex vehicle interaction in transportation systems. Furthermore, this study has major contributions in terms of spatiotemporal analysis of different LOS classes (i.e., A-F) under different data-collection time-intervals. In general, we emphasized short-term prediction, which is considered useful for improving the productivity of transportation systems, and also beneficial in reducing both the direct and indirect costs. Moreover, this study reviewed the different techniques and approaches that have been used for short-term TSP. A comprehensive comparative analysis was also conducted to evaluate the ability and efficiency of proposed methods in terms of prediction accuracy. The specific main contributions of this paper are:

- We extend the exploration of decision jungles and locally deep SVM (LD-SVM) for short term traffic state prediction using hyperparameter optimization (via random sweep).
- A comprehensive comparison was implemented to demonstrate the ability and the effectiveness of each machine learning model for TSP accuracy.
- Prediction performances were evaluated under different forecasting time-intervals at distinct time scales.
- Short-term traffic state was taken as a function of level of service (LOS) along a basic freeway segment. Study results demonstrated that decision jungles were more efficient and stable at different predicted horizons (time-intervals) than the LD-SVM, MLP, and CN2 rule induction.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of the methods and techniques for TSP in the existing literature. Section 3 describes the preliminaries for different machine learning models used in this study. Section 4 presents study area, data description, and key parameter settings. Section 5 highlights results and discussion. Section 6 includes the comparison of different models. Finally, Section 7 summarizes the conclusions, presents key study limitations, and outlook for future studies.

2. Related Work

Since early 1980, non-linear traffic flow prediction has been the focus of several research studies as it is regarded as extremely useful for real-time proactive traffic control measures [15,16]. From its inception in the 1980s, artificial neural networks (ANNs) have been widely used for the analysis and prediction of time series data. They have the ability to perceive the non-linear connection between features of input and output variables that in turn can produce effective TSP solutions. For example, Zheng et al. combined Bayesian inference and neural networks to forecast future traffic flow [35]. Ziang and Adeli proposed a time-delay via recurrent wavelet neural network, where the periodicity demonstrated the significance of traffic flow forecasting [36]. Parametric methods can obtain better prediction outcomes when the data flow of the traffic varies temporally. These methods assume a variety of difficult conditions such as residual normalization and predefined system structure and rarely converged due to the stochastic or non-linear traffic flow characteristics.

To address the limitations of parametric models, different approaches including linear kernel, polynomial kernel, Gaussian kernel, and optimized multi kernel SVM (MK-SVM) have been proposed by recent research studies for traffic flow prediction [37–40]. MK-SVM predicted the results by mapping the linear parts of historical traffic flow data using the linear kernel, while map residual was performed using the non-linear kernel. Alternatively, generating if-then rules, also known as

rule induction techniques that search the training data for proposition rules, can also be used. which CN2 is best-known example of this approach, that have been successfully utilized by previous for flow prediction [41,42]. Hashemi et al. developed different models for classification based on if-then rules in the short-term traffic state prediction for a highway segment [43]. In contrast, ANNs' popular network structure is multi-layer perceptron (MLP), which has been widely used in many transport applications due to its simplicity and capacity to conduct non-linear pattern classification and function approximation. The MLP model generally works well in the capture of complex and non-linear relations, but it usually requires a large volume of data and complex training. Many researchers, therefore, consider it as the most commonly implemented network topology [44–46]. Recently, in the study by Chen et al., they adapted a novel approach using dynamic graph hybrid automata for the modeling and estimation of density on an urban freeway in the city of Beijing, China [47]. The authors validated the feasibility of their modeling approach on Beijing's Third Ring Road. A recent study conducted by Zahid et al., proposed a new ensemble-based Fast forest quantile regression (FFQR) method to forecast short-term travel speed prediction [48]. It was concluded that proposed approach yielded robust speed prediction results, particularly at larger time-horizons.

Aside from the above-mentioned models, decision trees and forests have a rich history in machine learning and have shown significant progress in TSP, as reported in some of the recent literature [49,50]. Various studies have been conducted to address the shortcomings of traditional decision trees, for example, their sub-optimal efficiency and lack of robustness [51,52]. Similarly, in another research study, the researchers investigated the efficacy of the ensemble decision trees for the TSP [50]. It was concluded that trees generate efficient predictions traditionally. At the same time, researchers have concluded that learning with ideal decision trees could be problematic due to overfitting [53]. Henceforth, this approach has some limitations, such that the amount of data to be provided as the number of nodes in decision trees would increase exponentially with depth, affecting the accuracy [54]. Recently, a study proposed a novel online seasonal adjustment factors coupled with adaptive Kalman filter (OSAF-AKF) model for estimating the real-time seasonal heteroscedasticity in traffic flow series [55].

In contrast, machine learning techniques and their performances for classifying different problems have been encouraging such as decision jungles and LD-SVM, which are heavily dependent on a set of hyperparameters that, in turn, efficiently describes different aspects of algorithm behavior [54,56,57]. It is important to note that no suitable default configuration exists for all problem domains. Optimizing the hyperparameter for different models is important in achieving good performance in the realm of TSP [56]. There are two types of hyperparameter optimization: manual and automatic. Manual is time-consuming and depends on expert inputs, while an automatic approach removes expert input. Automatic approaches include the most common practice methods such as grid search and random search [58]. Several libraries have recently been introduced to optimize hyperparameters. Hyperopt Library is one of the libraries offering different hyper-optimization algorithms for machine learning algorithms [59]. Existing techniques for optimizing EC-based hyperparameters [60,61] such as differential evolution (DE) and particle swarm optimization (PSO) are useful since they are conceptually easy and can achieve highly competitive output in various fields [62–65]. However, these methods have a great deal of calculation and a low convergence rate in the iterative process. In contrast, hyperparameter optimization methods such as random grid, entire grid, and random sweep have achieved a great deal of attention in hyperparameter optimization. In a random grid, the matrix is computed for all combinations, and the values are extracted from the matrix by the number of defined iterations in relation to the entire grid incurred for all possible combinations. The difference between the random grid and the random sweep is that the latter technique selects random parameter values within the set, while the former only employs the exact values defined in the algorithm module. With this understanding, random sweep was chosen for the models conducted in this study for hyperparameter optimization with the intention of improving the accuracy of short-term TSP.

3. Preliminaries

Machine learning provides a number of supervised learning techniques for classification and prediction. The objective of a classification problem is to learn a model, which can predict the value of the target variable (class label) based on multiple input variables (predictors, attributes). This model is a function, which maps an input attribute vector X to the output class label (i.e., $Y \in \{C1, C2, C3, \dots, Cn\}$). The label training set is represented as follows:

$$(X, Y) = \{(x_0, x_1, x_2, x_3, \dots, x_n), Y\} \tag{1}$$

where Y is the target label class (dependent variable) and vector X is composed of $x_0, x_1, x_2, x_3, \dots, x_n$. The macroscopic flow, density, and speed obtained from traffic simulation are referred to as input parameters when fed/imported to machine learning models for short term traffic prediction. The model learns from these input variables for different time intervals (i.e., 5, 10, and 15 min). Either the next time interval level of service (LOS) is considered as a class label or target variable. The predicted label class for time (Time duration = 1), is given in the following form:

$$(Density_1, Speed_1, Flow_1, Time\ Duration_1, LOS_2) \tag{2}$$

The current study utilized four different machine learning methods for short term TSP. These methods included LD-SVM, decision jungles, CN2 rule induction, and MLP. The detailed methodology for each technique is presented below.

3.1. Local Deep Support Vector Machine (LD-SVM)

SVM is based on statistical learning theory as suggested by Vapnik in 1995 for classification and regression [66]. Local deep kernel learning SVM (LD-SVM) is a scheme for effective non-linear SVM prediction while preserving classification precision above an acceptable limit. Using a local kernel function allows the model to learn arbitrary local embedding features including sparse, high-dimensional, and computationally deep features that bring non-linearity into the model. The model employs routines that are effective and primarily infused to optimize the space of local tree-structured embedding features in more than half a million training points for big training sets. LD-SVM model training is exponentially quicker than traditional SVM models training [57]. LD-SVM can be used for both linear and non-linear classification tasks. It is considered as a special type of linear classifier (e.g., logistic regression LG), however, LG is unable to perform sufficiently in complicated and linear tasks. In addition, LD-SVM model learning is significantly faster and computationally more efficient than traditional SVM model training. The formulation of a local deep kernel learns a non-linear kernel $K(x_i, x_j) = K_L(x_i, x_j) K_G(x_i, x_j)$, where K_L and K_G are the local and global kernel. The product of local kernel $K_L = \phi_L^t \phi_L$ and global kernel $K_G = \phi_G^t \phi_G$ leads to the prediction function.

$$y(x) = \text{sign}(\phi_L^t(x) W^t \phi_G(x)) \tag{3}$$

$$y(x) = \text{sign} \left[\sum_{ijk} \alpha_i y_i \phi_{G_j}(x_i) \phi_{G_j} \phi_{L_k}(x_i) \phi_{L_k}(x) \right] \tag{4}$$

$$y(x) = \text{sign}(W^t(\phi_G(x) \otimes \phi_L(x))) \tag{5}$$

$$y(x) = \text{sign}(\phi_L^t(x) W^t \phi_G(x)) \tag{6}$$

$$y(x) = \text{sign}(W^t \phi_G(x)) \tag{7}$$

where $W_k = \sum_i \alpha_i y_i \phi_{L_k}(x_i) \phi_G(x_i)$, ϕ_{L_k} denote dimension k of $\phi_L \in R^M$, $W = [w_1 \dots w_M]$, $W(x) = W_{\phi_L}(x)$, and \otimes is the Kronecker product. ϕ_L is the local feature space and ϕ_G is the global features space.

$$\phi_{L_k}(x) = \tanh(\sigma \theta_k^t) I_k(x) \tag{8}$$

while training the LD-SVM and smoothing the tree are shown in Figure 1, Equation (1) can further written as below:

$$y(x) = \text{sign}[\tanh(\sigma v_1^t x) w_1^t x + \text{anh}(\sigma v_2^t x) w_2^t x + \text{anh}(\sigma v_4^t x) w_4^t x] \tag{9}$$

where $I_k(x)$ is the indicator function for each node k in the tree; θ is to go left or right; v stack with non-linearity; σ is sigmoid sharpness for the parameter scaling and could be set by validation. Higher values imply that the ‘tanh’ is saturated in the local kernel, while a lower value means a more linear range of operation for θ . The full optimization formula is given in Equation (10). The local deep kernel learning (LDKL) primal for jointly learning θ and W from the training data, where $\{(x_i, y_i)_{i=1}^N\}$ can be described as:

$$\min_{W, \theta, \theta'} P(W, \theta, \theta') = \frac{\lambda_w}{2} T_r(W^t W) + \frac{\lambda_\theta}{2} T_r(\theta^t \theta) + \frac{\lambda_{\theta'}}{2} T_r(\theta'^t \theta') + \sum_{i=1}^N L(y_i, \phi_L^t(x_i) W^t x_i) \tag{10}$$

where $L = \max(0, 1 - y_i \phi_L^t(x_i) W^t x_i)$; λ_w is the weight of the regularization term; and λ_θ specifies the amount of space between the region boundary and the nearest data point to be left. $\lambda_{\theta'}$ controls the curvature amount allowed in the model’s decision boundaries.

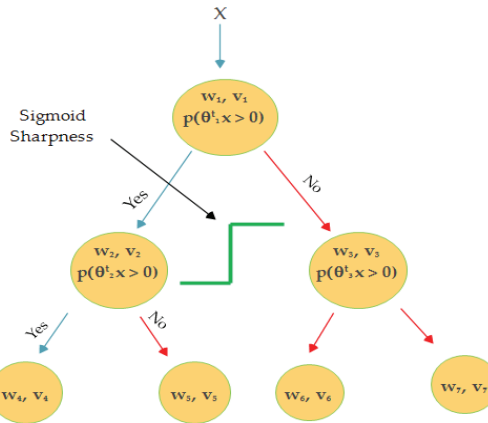


Figure 1. Schematic diagram of the as local deep support vector machine (LD-SVM).

3.2. Decision Jungles

Decision jungles are the latest addition to decision forests. They are comprised of a set of decision-making acyclic graphs (DAGs). Unlike standard decision trees, the DAG in the decision jungle enables different paths from root to leaf. A DAG decision has a reduced memory footprint and provides superior efficiency than a decision tree. Decision jungles are deemed as non-parametric models that provide integrated feature selection, classification, and are robust in the presence of noisy features. DAGs have the same structure as decision trees, except that the nodes have multiple parents. DAGs can limit the memory consumption by specifying a width at each layer in the DAG and potentially help to reduce overfitting [54]. Considering the nodes set at two consecutive levels of DAGs, Figure 2 shows that the nodes set consists of child nodes N_c and parent nodes N_p . Let θ_i denote the parameters of

the split function f for parent node $i \in N_p$. S_i denotes the categorized training samples (x, y) , where it reaches node i , and set of samples can be calculated from node i , which travels through its left or right branches. Given θ_i and S_i , the left and right are computed by $S_i^L(\theta_i) = \{(x, y) \in S_i | f(\theta_i, x) \leq 0\}$ and $S_i^R(\theta_i) = S_i / S_i^L(\theta_i)$, respectively. $l_i \in N_c$ indicates the left outward edge from parental node $i \in N_p$ to a child node, and $r_i \in N_c$ denotes the right outward edge. Henceforth, the number of samples reaching any child node $j \in N$ is given as:

$$S_j(\{\theta_i\}, \{l_i\}, \{r_i\}) = \left[\bigcup_{i \in N_p^S : l_i=j} S_i^L(\theta_i) \right] \cup \left[\bigcup_{i \in N_p^S : r_i=j} S_i^R(\theta_i) \right] \tag{11}$$

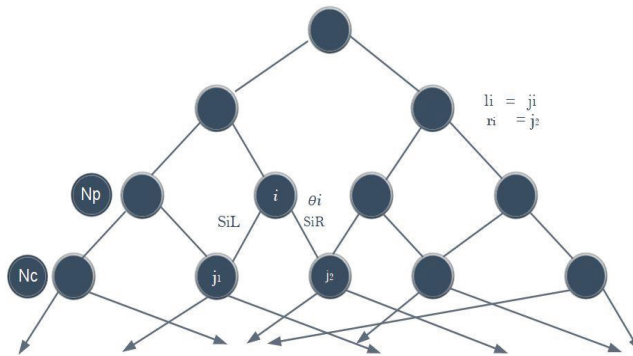


Figure 2. Decision jungles (DAGs).

3.3. CN2 Rule Induction

In this study, rule learning models were also explored for TSP. These models are usually used for classification and prediction solutions. The CN2 algorithm is a method of classification designed to induce simple efficiency; “if condition then predicts class,” even in areas where noise may occur. Inspired by Iterative Dichotomiser 3 (ID3), the original CN2 uses entropy as the function for rule evaluation; Laplace estimation may be defined as an alternative measure of the rule quality to fix unpleasant entropy (downward bias), and it is described as follows [67]:

$$Laplace\ Estimation\ (R) = \frac{p + 1}{P + n + k} \tag{12}$$

where ‘ p ’ represents the number of positive examples in the training set covered by Rule ‘ R ’; n represents the number of negative instances covered by R ; and ‘ k ’ is the number of the training classes available in the training set.

3.4. Multi-Layer Perceptron

The most common ANN model is the multi-layer perceptron (MLP). In MLP, input values are transformed by activation function f , giving the value as an output from the neuron. The MLP is made up of various layers including one input layer, one or more hidden layers, and one output layer. In MLP, parameters such as the number of input variables, number of hidden layers, activation function, and learning rate play an important role in the design of neural network architecture. The multi-layer perceptron (MLP) is shown in Figure 3. Neurons have activation functions for both the

hidden layer and the output layer; neurons receive only the input dataset and have no activation functions on the input layer. Weights are multiplied with inputs, and are summarized accordingly as;

$$f(x_i) = \sum_{i=1}^n (w_i x_i) + bias \quad (13)$$

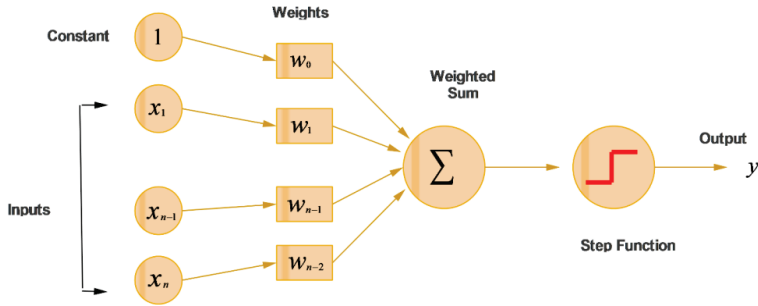


Figure 3. Schematic algorithm of multi-layer perceptron (MLP).

Whilst the most commonly applied activation function is logistic function (sigmoid function), given in following equation:

$$f(x_i) = \frac{1}{1 + e^{-x}} \quad (14)$$

4. Study Area

This study was conducted in the city of Beijing, China, which covers an area of 16,410 km², and hosts 21.7 million people. Road transportation is an integral part of the city's routine businesses, linking most households to workplaces or schools. There are 21,885 km of paved public road in Beijing (as of June 2016), 982 km of which are classified as highways [68]. According to the Beijing census, the number of private cars was close to 5.4 million, in addition to 5.3 million other vehicles in different categories including 330,100 trucks. The Second-Ring Road consists of six percent of the urban space of Beijing, with clusters of major companies, businesses, and administrative institutions, but generate 30% of the traffic volume per day [68,69]. Within this perspective, integrated urban planning is becoming difficult, so much so that 60% of the historical site of the city is lying on the Second Ring Road. Since the traffic hotspots are concentrated mainly in the center of Beijing, we have chosen an area as the study area at this location [68,70]. The Second Ring Road is approximately 33 km long including 37 on-ramps and 53 off-ramps. Figure 4 shows the study area on the Second Ring Road along with other different ring roads. In this study, a basic freeway segment of the Second Ring (L = 478.5 m) was selected.

Data Collection and Parameters Setting

The first step in preparing the experiment was to develop a microscopic model using VISSIM (Micro Traffic Simulation Software) to capture all the essential data for the Second Ring Road. When simulating the field conditions, it is essential to calibrate the driving behavior parameters for the traffic simulator, and this was accomplished by standard procedures, as reported in the existing work [71]. In doing so, several simulation iterations were performed, incurring a different random seed to ensure that the model works under the real-time scenario. The proposed methodology for the present study is presented in Figure 5.

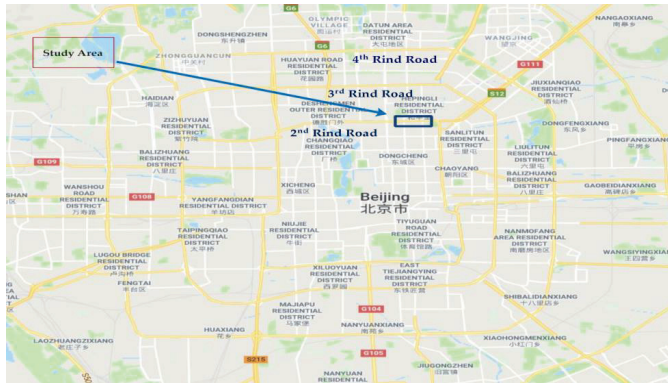


Figure 4. Second ring road (from Google Maps). Note: The Chinese words on map indicate names of surrounding infrastructure.

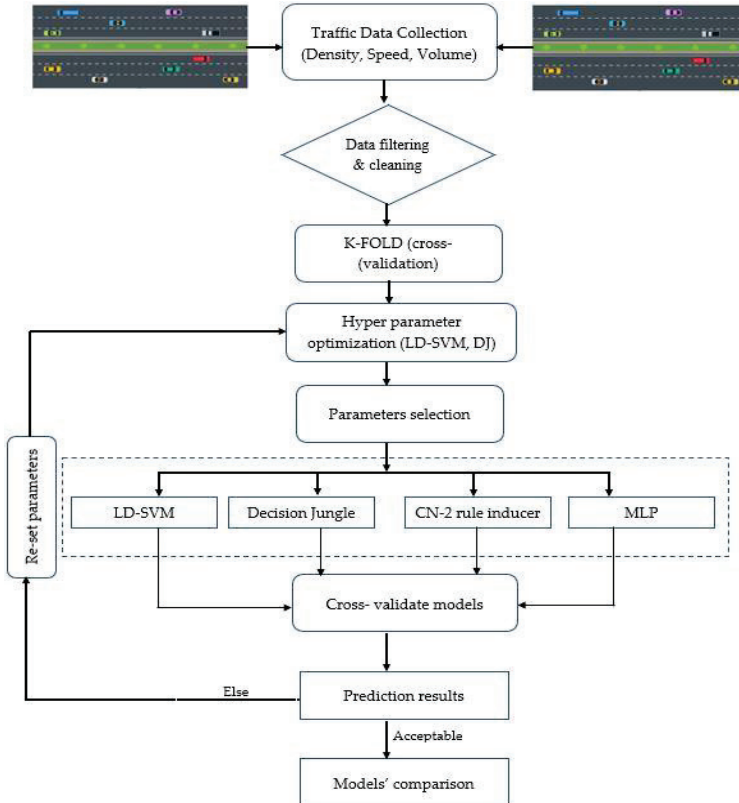


Figure 5. Proposed methodology for the study.

In this study, macroscopic traffic parameters (volume, speed, density) were obtained from the VISSIM simulation analysis. Traffic volume or flow rate can be defined as the number of vehicles that pass through a point on a highway or lane at a specific time, and is usually expressed in units of

vehicles per hour per lane (v/h/l), while density is referred to the number of vehicles occupying a unit length of roadway, and is denoted by vehicles per km/mile per lane (v/m/ln). Occupancy is sometimes synonymously used with density; however, it should be noted that it shows the percentage of time that a road segment is occupied by vehicles. Traffic speed is another important state parameter, and can be found by the distance traversed per unit of time, and is typically expressed in km/h. or miles/h. These parameters are further calculated by using the link evaluation in VISSIM. Once the factual freeway architecture is achieved, the key macroscopic characteristics are identified in order to adjust the entire microscopic simulator (e.g., demand flow and split ratio). Demand flow is defined as the traffic volume as it utilizes the facility, while split ratio is the directional hourly volume (DHV) in the peak direction, which varies with respect to time, that is, the peak time and off-peak time. Additionally, the real traffic state of the Second Ring Road in this study was obtained from the Beijing Collaborative Innovation Center for Metropolitan Transportation. Thereby, the model of the road network deemed for the Second Ring Road was constructed by VISSIM. It has three lanes, where each lane is designated with an average width of 3.75 m, as shown in Figure 6. Simulations in the VISSIM were carried for 6 h, during the period 6:00 am to 12:00 pm, and a congested regime prevailed from 1.5 to 2 h (i.e., between 7:30 am to 9:30 am), leveraging the almost free flow for the remaining hours. Therefore, the transition state from D to F encountered few labels. Meanwhile, data were collected using different prediction horizons such as 5, 10, and 15 min.

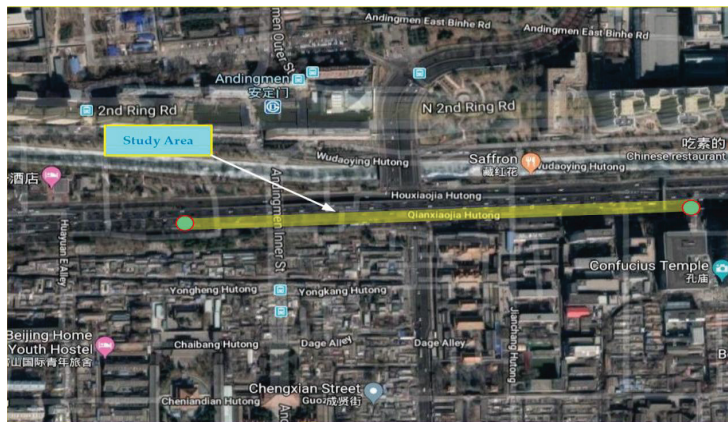


Figure 6. Basic freeway segment of the Second Ring Road (from Google Earth). Note: The Chinese words on map indicate names of surrounding infrastructure.

To assess the freeway operations, level-of-service (LOS), a commonly used performance indicator, was used for qualitative evaluation purposes. The data collected from the VISSIM simulation was further divided into six levels [72], wherein the LOS defines the traffic state of each level. Traffic state is usually characterized by traffic-density on a given link, and is directly related with the number of vehicles occupying the roadway segment. It also represents the transient boundary conditions between two LOS levels. Moreover, to test the efficacy, classification models were built in python scripting orange software and azure machine learning to write the required procedures for extracting the traffic parameters, and level-of-service corresponded to highway capacity manual (HCM) [43,73]. The data points (in Figure 7) represent different points in time distributed spatially, which together define the LOS at the road segment. In the mentioned figure, different colors showed the states for 15 min, which is actually the LOS divided into six sub-levels based on density along the highway segment. We termed these levels as different states (from A to F) and further evaluated them for 5, 10, and 15 min intervals. Since stratified K-fold cross validation was opted to address the issue of imbalance data, the method

aimed to choose the proportionate frequencies for each LOS class. Thus, it is likely that label D or any other label will be associated with true representative class. The actual density–flow captured on a segment of the Second Ring was simulated in VISSIM for a prediction horizon of 15 min and is shown in Figure 7.

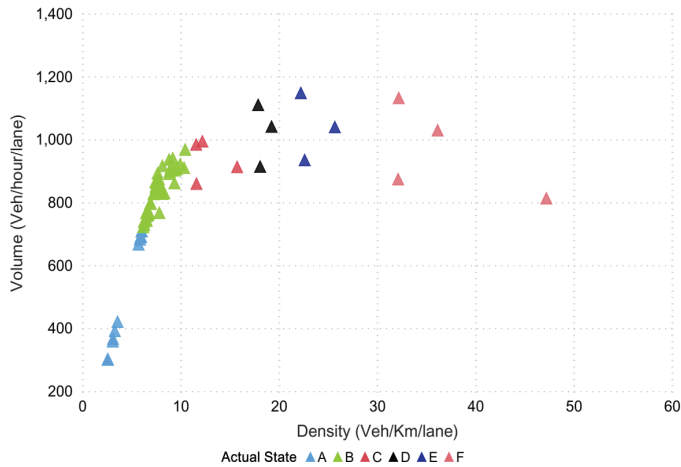


Figure 7. The actual density–flow via VISSIM.

5. Results and Discussion

5.1. K-Fold Cross-Validation

We selected the K-Fold cross-validation method (using $k = 10$), which is used for a better f -model, and it provides the appropriate settings for parameters. The original instances were randomly split into k equal parts. A single part was used for validation from the k split, and the remaining k minus one ($k - 1$) parts were used for the training set in order to develop the model. To do so, we revised the same technique k times. Each time a distinct validation dataset was selected, until the model's final accuracy was equal to the average accuracy, that in turn, was achieved in each iteration. This technique has the advantage over repeated random sub-sampling as all the samples are used for training as well as in the validation, where each sample is used once for the validation. To avoid the problems of data imbalance and enhance the prediction accuracy of the proposed methods, several strategies have been suggested by previous studies. In this study, K-fold cross validation was used to overcome the issues and bias associated with imbalance and small datasets as the K-fold validation method is more efficient and robust compared to other conventional techniques, since it preserves the percentage of samples for each group or class. We tuned the parameters to obtain the best results with accuracy and they were selected using hyperparameter tuning.

5.2. Model Evaluation

In this study, we selected the most common evaluation metrics in order to assess the performances of the models known as F score and Accuracy. The F score is a measure of the accuracy of a test, also known as the F-1 score or F measure. The F-1 score is defined as the weighted average of recall and precision. To measure the overall performances of the model, the F-1 score was derived as follows:

$$F - 1 = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \quad (15)$$

Accuracy is one of the classifications' performance measures, which is defined as the ratio of the correct sample to the total number of samples as follows [74],

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

where P and N denote the number of positive and negative samples, respectively. TP and TN indicate the true positive and true negative. FP and FN indicate the false positive and false negative, respectively.

5.3. Local Deep Kernel Learning SVM (LD-SVM)

The tuning parameters include LD-SVM tree depth, Lambda W, Lambda theta, Lambda theta prime, number of iterations, and sigmoid sharpness or sigma. Figure 8a shows the LD-SVM tree depth impact on accuracy and 92.00% accuracy was achieved when the tree depth was 3. The impact of the other parameters, Lambda W, Lambda theta, Lambda theta prime, number of iterations, and sigmoid sharpness or sigma, can be seen in Figure 8c. The best hyperparameter tuned values for these parameters were 0.00052, 0.34587, 0.1025, 49,247, and 0.0068, which were encircled and obtained using 10-fold cross-validation. Figure 8b shows the predicted state for the next 15 min

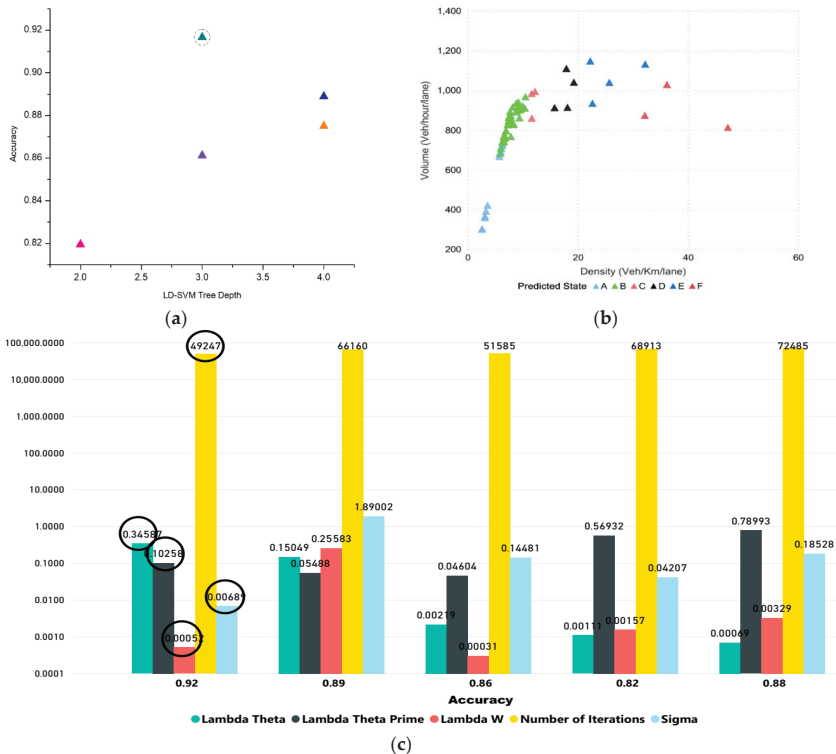


Figure 8. The LD-SVM model. (a) The impact of tree depth on accuracy. (b) Predicted state for next 15 min (c) Impact of Lambda theta, Lambda theta prime, Lambda W, and sigmoid function on accuracy.

5.4. Decision Jungle

The tuning parameters in the decision jungle model were described by the maximum depth of the decision (DAGs), number of decision DAGs, number of optimization steps per decision, DAGs layer, and maximum width of the decision DAGs'. Figure 9b shows the impact of the maximum

depth of decision DAGs on the accuracy of the model. The accuracy was 92% and was achieved when the maximum depth of the decision (DAGs) was 77. The best-tuned values for the other parameters are depicted in Figure 9c (such as the number of decisions DAGs, number of optimization steps per decision, and maximum width of decision DAGs' were 22, 5786, and 19, respectively), and were obtained using 10-fold cross-validation. Since, our study considered 15 min prediction horizons as the structure of DAG is illustrated in Figure 9d, which shows the number of DAGs is 22 with a maximum depth of levels 77. The predicted state for 15 min horizons can be seen in Figure 9a.

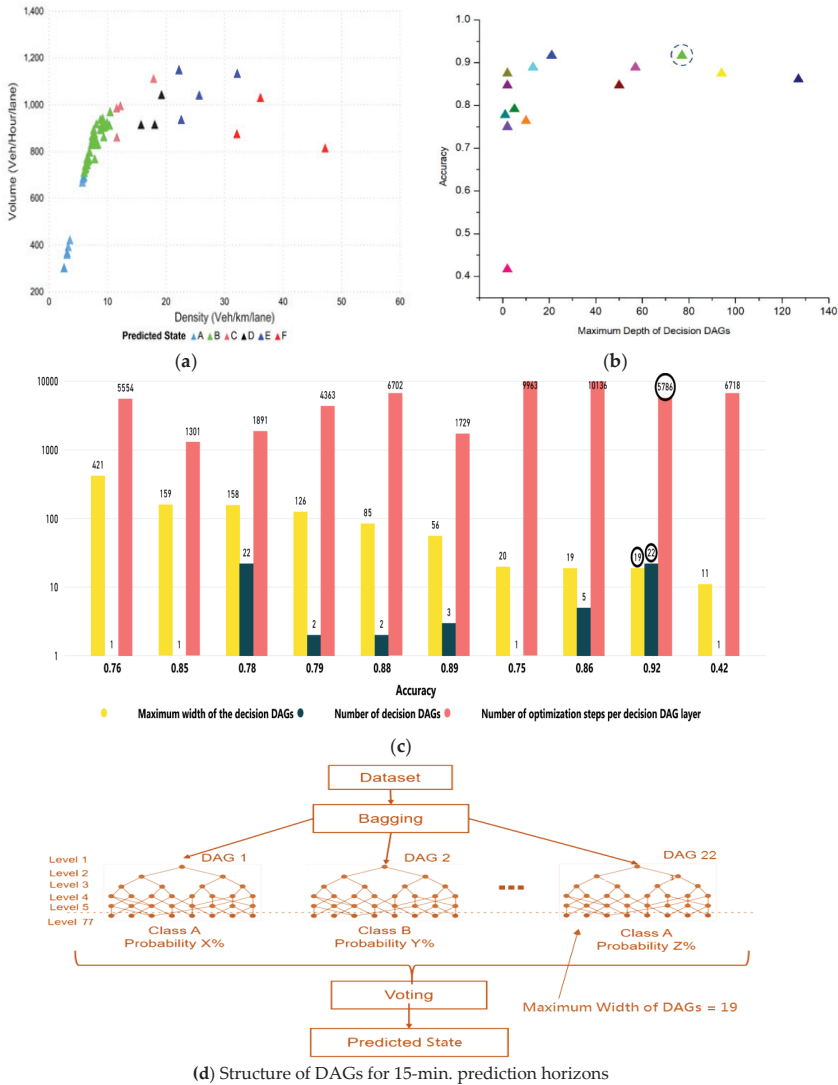


Figure 9. Decision jungle model. (a) The impact of maximum depth on accuracy. (b) Predicted state for the next 15 min (c) Impact of maximum width of the decision DAGs and number of decision DAGs on accuracy. (d) Number of DAGs, width, and depth of DAGs are shown for 15-min prediction horizons.

5.5. CN2 Rule Induction

CN2 utilizes a statistical significance test in order to ensure that the fresh rule represents a real correlation between features and classes. In fact, it is a pre-pruning technique that prevents particular rules after their implementation. Moreover, it performs a sequential covering approach at the upper stage (also defined as split-and-conquer or cover-and-remove), once used by the algorithm quasi-optimal (AQ) algorithm. The CN2 rule returns a class distribution in terms of the number of examples covered and distributed over classes. The distribution in Table 1 and the tables in the Appendix show that each number corresponded to the number of example(s) that belonged to class $LOS = i$, where $i = \{A, B, C, D, E, F\}$ and “i” is the observed frequency distribution of examples between different classes. In another words, it represents number of the relevant class membership. The derived probabilities shown in Table 1 can be used to check the accuracy and efficiency of that particular rule. We adopted exclusive coverage in our implementation at the upper level such as unordered CN2 [62], whereas Laplace estimation was used for function evaluation at the lower level. Pre-pruning of rules was performed using two methods: (i) likelihood ratio statistic (LRS) tests, and (ii) minimum threshold for coverage of rules. The LRS test indicates two tests: first, a rule’s minimum level of significance α_1 , and the second LRS test is likened to its parent rule, as it checks whether the last rule specialization has a sufficient level of significance α_2 . The values for the LRS tests and rules for the different prediction horizons were obtained using 10-fold cross-validation. Figure 10 shows the predicted state for 15 min intervals. The values of α_1 and α_2 are listed in Table 2. The rule for the next 5 min and 10 min horizons is given in Appendix A, whilst the rule for the next 15 min horizons is given in Table 1.

Table 1. Selected rules (for 15-min prediction horizon) with rule quality.

| IF Condition | Then (Next State) | Distribution | Probabilities [%] | Rule Quality | Rule Length |
|---|-------------------|---------------------|------------------------|--------------|-------------|
| Time (Seconds) ≤ 13500.0 AND Speed (Km/h) ≥ 117.83 | A | [8, 0, 0, 0, 0, 0] | 6: 4: 7: 7: 7: 7 | 0.903 | 2 |
| Speed (Km/h) ≥ 88.43 AND Volume (Veh./h/lane) ≥ 723.35 | B | [0, 45, 0, 0, 0, 0] | 2: 90: 2: 2: 2: 2 | 0.98 | 2 |
| Time (Seconds) ≤ 9000.0 AND Density (Veh/Km/lane) ≥ 11.54 | C | [0, 0, 3, 0, 0, 0] | 11: 11: 44: 11: 11: 11 | 0.805 | 2 |
| Density (Veh/Km/lane) ≤ 22.19 AND Density (Veh/Km/lane) ≥ 17.85 | D | [0, 0, 0, 4, 1, 0] | 9: 9: 9: 45: 18: 9 | 0.715 | 2 |
| Speed (Km/h) ≥ 36.91 AND Density (Veh/Km/lane) ≥ 22.19 | E | [0, 0, 0, 0, 3, 0] | 11: 11: 11: 11: 44: 11 | 0.805 | 2 |
| Density (Veh/Km/lane) ≥ 32.09 | F | [0, 0, 0, 0, 0, 4] | 10: 10: 10: 10: 10: 50 | 0.855 | 1 |

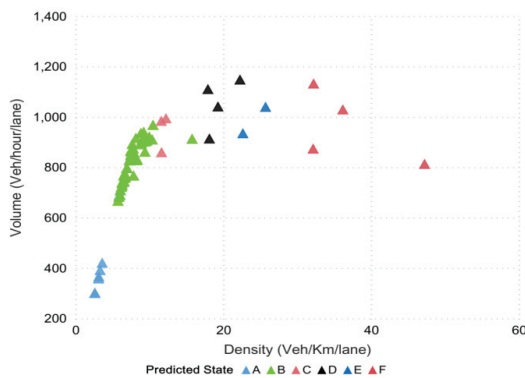


Figure 10. Predicted state for the next 15 min horizon.

Table 2. CN2 rule setting parameter values.

| Time Intervals (min) | α_1 | α_2 |
|----------------------|------------|------------|
| 5 | 0.05 | 0.03 |
| 10 | 0.05 | 0.02 |
| 15 | 0.05 | 0.03 |

5.6. Multi-Layer Perceptron (MLP)

In neural networks, learning includes adjusting the connection weights between neurons and each functional neuron’s threshold. We considered one input layer and one hidden layer with 35 neurons. The input layer had four nodes: speed, density, flow, and time duration (interval). The accuracy achieved using 10-fold cross-validation for different prediction horizons was compared (shown in Table 3) against the learning rate, momentum, activation function, and epochs. Figure 11a shows the predicted state for the next 15 min horizons. The input layer, hidden layers with neurons, and output layers for the MLP network are depicted in Figure 11b.

Table 3. Configuration of the parameters for the multi-layer perceptron (MLP).

| Prediction Horizons | Algorithm | Hidden Layers | Hidden Neurons | Activation Function | Epochs | Learning Rate | Momentum | Accuracy |
|---------------------|-----------|---------------|----------------|---------------------|--------|---------------|----------|----------|
| 5 min | MLP | 01 | 35 | Sigmoid | 500 | 0.2 | 0.2 | 0.949 |
| 10 min | MLP | 01 | 35 | Sigmoid | 500 | 0.2 | 0.2 | 0.924 |
| 15 min | MLP | 01 | 35 | Sigmoid | 500 | 0.3 | 0.2 | 0.875 |

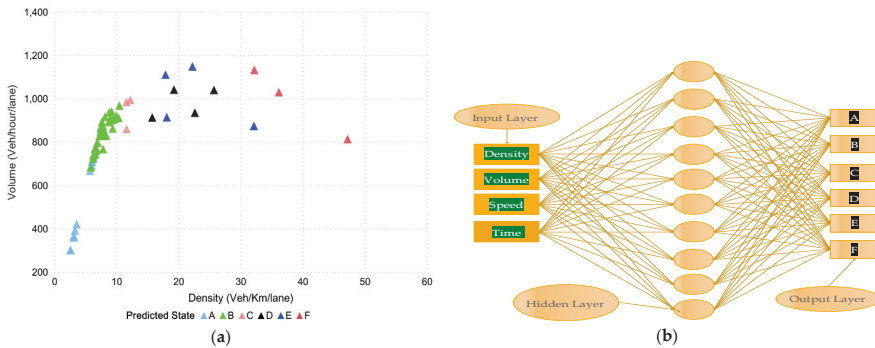


Figure 11. MLP model. (a) Predicted state for next 15 min horizon. (b) MLP network with 01 hidden layer.

6. Model Comparison

The weighted average F-1 score and accuracy were evaluated in order to assess the performances of different models. The results suggest that decision jungles outperformed the LD-SVM, CN2, and MLP, as shown in Figure 12. Additionally, the decision jungles and LD-SVM achieved a higher weighted average F-1 score. In particular, the decision jungle was found to have improved results over the LD-SVM, CN2, and MLP, and obtained high F-1 scores of 0.9777, 0.952, and 0.915 were predicated for time horizons of 15, 10, and 5 min, respectively. Similarly, the LD-SVM was slightly better than the MLP and CN2 as the F1-score was higher (0.904, 0.926, 0.946) for the 15, 10, and 5 min prediction horizons. However, the CN2 rule induction performed better, except for decision jungles, while the other models failed to achieve a higher F–1 score for the same prediction horizon. On the other hand, Figure 13a,b shows that decision jungles and LD-SVM also achieved higher accuracy when

compared to the remaining models such as CN2 rule induction and MLP. It can be noted that as the prediction horizons increases, the F-1 score and accuracy decreases. This indicates that decision jungles were stable when compared to the results in accordance with time horizons of 15, 10, and 5. Unlike the LD-SVM, MLP and CN2 were found to be less effective at maintaining the stability of accuracy in different time horizons. However, the CN2 rule induction in Figure 13c,d) performed well and provided stable results only for the 10, and 15 min prediction horizons.

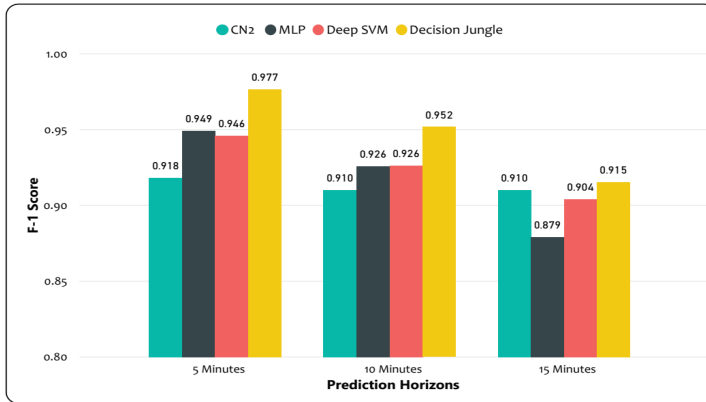


Figure 12. Model comparison. Weighted average F-1 score for decision jungles; weighted average F-1 score for LD-SVM; weighted average F-1 score for MLP; weighted average F-1 score for CN2 rule induction.

The experimental results are summarized in Tables 4 and 5, where the models’ performances were computed using F-1 score and the average accuracy for different prediction horizons, respectively. It can be clearly seen that decision jungles achieved a higher F-1 score and gained a higher accuracy when compared to the other models for different prediction horizons. This shows that decision jungles achieved an average improvement of 95% and outperformed the remaining models. However, the LD-SVM performed better than the MLP and CN2 rule induction.

Table 4. F-1 score for the different model comparisons.

| F-1 Score | Prediction Horizons (min) | | |
|--------------------|---------------------------|-------------|-------------|
| | 5 | 10 | 15 |
| Decision Jungle | 0.976683061 | 0.951784174 | 0.915209941 |
| LD-SVM | 0.946083305 | 0.926083351 | 0.904265873 |
| MLP | 0.949 | 0.926 | 0.879 |
| CN2 Rule Induction | 0.92 | 0.91 | 0.910 |

Table 5. Accuracy for different models comparisons.

| Accuracy | Prediction Horizons (min) | | |
|--------------------|---------------------------|----------|----------|
| | 5 | 10 | 15 |
| Decision Jungle | 0.992212 | 0.984277 | 0.972222 |
| LD-SVM | 0.982866 | 0.974843 | 0.967593 |
| MLP | 0.983262872 | 0.973 | 0.9577 |
| CN2 Rule Induction | 0.975 | 0.97183 | 0.9581 |

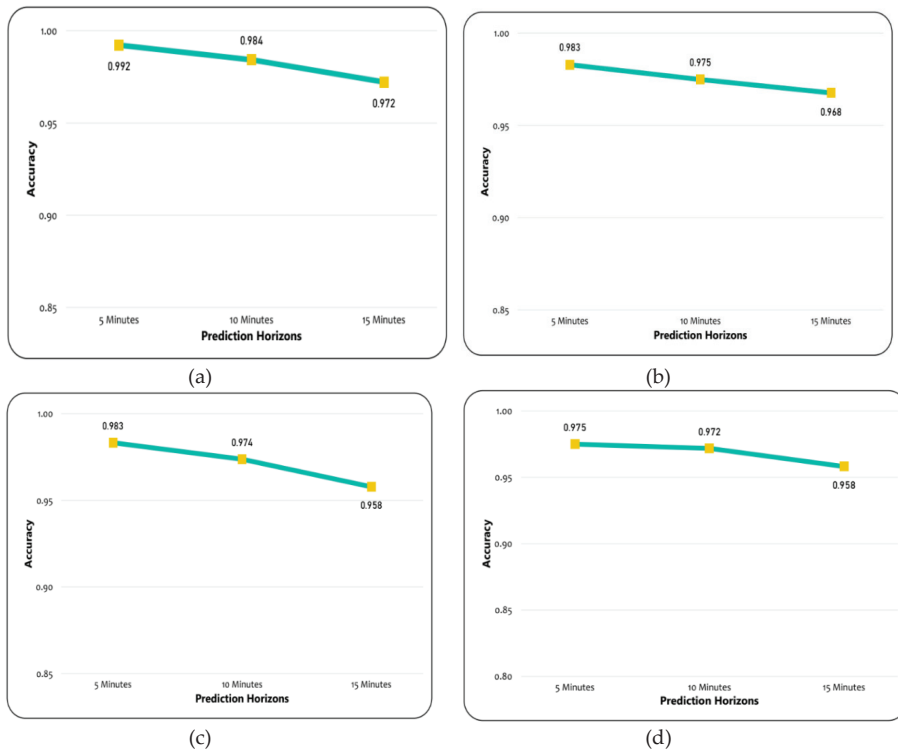


Figure 13. Model Comparison. (a) Accuracy for the decision jungles. (b) Accuracy for the LD-SVM. (c) Accuracy for the MLP. (d) Accuracy for the CN2 rule induction.

7. Conclusions

In this study, we improvised machine learning models with hyperparameter tuning optimization for short term TSP. Different schemes offered in parameter tuning were examined by performing the number of simulation iterations incurring different random seeds to ensure that the model worked efficiently under a real-time scenario. To do so, a comprehensive demonstration and the ability of different machine learning models were evaluated using different forecasting time-intervals at distinct time scales. The short-term traffic state was taken as a function of level-of-service (LOS) on a basic freeway segment along Second Ring Road in Beijing, China. Simulation of a transportation road demonstrated that decision jungles were more efficient and stable at different predicted horizons (time intervals) than the LD-SVM, MLP, and CN2 rule induction. Data utilized in this study was collected from traffic simulator VISSIM. Actual density–flow was captured on freeway segment via different prediction horizons of 15, 10, and 5 min. The experimental results showed and demonstrated the superior and robust performance of decision jungles compared to the LD-SVM, CN2 rule induction, and MLP. The overall performance of prediction results were improved by over 95 percent on average, which led to an accuracy of 0.982 and 0.975 for the decision jungle and LD-SVM. Moreover, the prediction performance for CN2 rule induction were also observed to be improved based on if–then rules in terms of the traffic patterns for different prediction horizons.

This study has some limitations that must be acknowledged. First, the proposed study was deployed in a developed urban freeway network model, so the simulated data need to be enhanced in future studies. Second, instead of justifying the efficacy of the suggested techniques using microscopic

simulation platform via VISSIM, forthcoming studies may focus on investigating and verifying the performance of proposed methods with an improved model on real traffic data.

In the future, studies may focus on long-term traffic state prediction (hours, days, weeks), which could also be divided into different LOS groups. The study area can be extended from the basic freeway segment to weaving, merging, and diverging segments that cover the entire network range of the Second Ring Road. Studies could incorporate temperature, air quality, weather, and other external factors that are likely to affect travel demand, thus, enhance prediction accuracy. In addition, it could rely on considering larger and various types of traffic datasets to analyze various combinations of flow, occupancy, speed, and other characteristics of road traffic to improve the predictive accuracy by using improved machine learning methods for prediction and analytics.

Author Contributions: Conceptualization, M.Z. and Y.C.; Methodology, M.Z. and Y.C.; Software, M.Z.; Validation, M.Z. and Y.C.; Formal analysis, M.Z. and Y.C.; Investigation, M.Z. and A.J.; Resources, M.Z. and Y.C.; Writing—original draft preparation, M.Z., Y.C., and M.Q.M.; Writing—review and editing, M.Z., A.J., and M.Q.M.; Visualization, M.Z. and A.J. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Natural Science Foundation of China (Grant No. 61573030).

Acknowledgments: The authors acknowledge the support of the Beijing University of Technology in providing the essential resources for conducting this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. If–then rules for the next 5-min horizon.

| IF Conditions | THEN (Next State) | Distribution | Probabilities [%] | Rule Quality | Rule Length |
|---|-------------------|---------------------|------------------------|--------------|-------------|
| Density (Veh/Km/lane) ≤ 7.03 | A | [95, 1, 0, 0, 0, 0] | 94: 2: 1: 1: 1: 1 | 0.98 | 1 |
| Speed (Km/hr) ≥ 105.65 AND Volume (veh/h/lane) ≥ 847.03 | B | [0, 38, 0, 0, 0, 0] | 2: 89: 2: 2: 2: 2 | 0.975 | 2 |
| Density (Veh/Km/lane) ≤ 7.03 AND Speed (Km/h) ≥ 85.03 | B | [0, 39, 1, 0, 0, 0] | 2: 87: 4: 2: 2: 2 | 0.952 | 2 |
| Density (Veh/Km/lane) ≥ 11.45 AND Speed (Km/h) ≥ 64.05 | C | [0, 0, 9, 0, 0, 0] | 7: 7: 67: 7: 7: 7 | 0.909 | 2 |
| Density (Veh/Km/lane) ≤ 22.26 AND Density (Veh/Km/lane) ≥ 16.84 | D | [0, 0, 0, 7, 1, 0] | 7: 7: 7: 57: 14: 7 | 0.8 | 2 |
| Speed (Km/h) ≥ 37.3 AND Density (Veh/Km/lane) ≥ 22.26 | E | [0, 0, 0, 0, 3, 0] | 11: 11: 11: 11: 44: 11 | 0.8 | 2 |
| Density (Veh/Km/lane) ≥ 29.64 | F | [0, 0, 0, 0, 0, 19] | 4: 4: 4: 4: 4: 80 | 0.952 | 1 |

Table A2. If–then rules for the next 10-min horizon.

| IF Conditions | THEN (Next State) | Distribution | Probabilities [%] | Rule Quality | Rule Length |
|---|-------------------|---------------------|------------------------|--------------|-------------|
| Time (Seconds) ≤ 8400.0 AND Speed (Km/h) ≥ 118.5 | A | [14, 0, 0, 0, 0, 0] | 75: 5: 5: 5: 5: 5 | 0.938 | 2 |
| Density (Veh/Km/lane) ≥ 6.04 | A | [7, 1, 0, 0, 0, 0] | 57: 14: 7: 7: 7: 7 | 0.8 | 1 |
| Speed (Km/h) ≥ 87.12 AND Density (Veh/Km/lane) ≥ 6.04 | B | [0, 56, 0, 0, 0, 0] | 2: 92: 2: 2: 2: 2 | 0.983 | 2 |
| Density (Veh/Km/lane) ≤ 16.59 AND Density (Veh/Km/lane) ≥ 11.01 | C | [0, 0, 12, 1, 0, 0] | 5: 5: 68: 11: 5: 5 | 0.867 | 2 |
| Density (Veh/Km/lane) ≤ 23.65 AND Density (Veh/Km/lane) ≥ 16.59 | D | [0, 0, 0, 6, 1, 0] | 8: 8: 8: 54: 15: 8 | 0.778 | 2 |
| Speed (Km/h) ≥ 43.9 AND Density (Veh/Km/lane) ≥ 23.65 | E | [0, 0, 0, 0, 2, 0] | 12: 12: 12: 12: 38: 12 | 0.75 | 2 |
| Density (Veh/Km/lane) ≥ 28.42 | F | [0, 0, 0, 0, 0, 7] | 8: 8: 8: 8: 8: 62 | 0.889 | 1 |

References

1. Atallah, R.F.; Khabbaz, M.J.; Assi, C.M. Vehicular networking: A survey on spectrum access technologies and persisting challenges. *Veh. Commun.* **2015**, *2*, 125–149. [[CrossRef](#)]
2. Lloret, J.; Canovas, A.; Catalá, A.; Garcia, M. Group-based protocol and mobility model for VANETs to offer internet access. *J. Netw. Comput. Appl.* **2013**, *36*, 1027–1038. [[CrossRef](#)]
3. Soleymani, S.A.; Abdullah, A.H.; Zareei, M.; Anisi, M.H.; Vargas-Rosales, C.; Khurram Khan, M.; Goudarzi, S. A secure trust model based on fuzzy logic in vehicular Ad Hoc networks with fog computing. *IEEE Access* **2017**, *5*, 15619–15629. [[CrossRef](#)]
4. Ji, B.; Hong, E.J. Deep-learning-based real-time road traffic prediction using long-term evolution access data. *Sensors* **2019**, *19*, 5327. [[CrossRef](#)]
5. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.V.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Image Process.* **2017**, *11*, 68–75. [[CrossRef](#)]
6. El-Sayed, H.; Sankar, S.; Daraghmi, Y.A.; Tiwari, P.; Rattagan, E.; Mohanty, M.; Puthal, D.; Prasad, M. Accurate traffic flow prediction in heterogeneous vehicular networks in an intelligent transport system using a supervised non-parametric classifier. *Sensors* **2018**, *18*, 1696. [[CrossRef](#)]
7. Wan, J.; Liu, J.; Shao, Z.; Vasilakos, A.V.; Imran, M.; Zhou, K. Mobile crowd sensing for traffic prediction in internet of vehicles. *Sensors* **2016**, *16*, 88. [[CrossRef](#)]
8. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
9. Jamal, A.; Subhan, F. Public perception of autonomous car: A case study for Pakistan. *Adv. Transp. Stud. Int. J. Int. J. Sect. A* **6** **2019**, *49*, 145–154.
10. Abdulhai, B.; Porwal, H.; Recker, W. Short-Term Traffic Flow Prediction Using Neuro-Genetic Algorithms. *J. Intell. Transp. Syst.* **2002**, *7*, 3–41. [[CrossRef](#)]
11. Van Lint, J.W.C.; Van Hinsbergen, C. Short-term traffic and travel time prediction models. *Artif. Intell. Appl. to Crit. Transp. Issues* **2012**, *22*, 22–41.
12. Du, L.; Peeta, S.; Kim, Y.H. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. *Transp. Res. Part B Methodol.* **2012**, *46*, 235–252. [[CrossRef](#)]
13. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we're going. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 3–19. [[CrossRef](#)]
14. Chan, K.Y.; Dillon, T.S.; Singh, J.; Chang, E. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 644–654. [[CrossRef](#)]
15. Williams, B.M. Flow Prediction Evaluation of ARIMAX Modeling. *Transp. Res. Rec.* **2001**, *1776*, 194–200. [[CrossRef](#)]
16. Williams, B.M.; Hoel, L.A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J. Transp. Eng.* **2003**, *129*, 664–672. [[CrossRef](#)]
17. Li, L.; Zhang, J.; Wang, Y.; Ran, B. Missing Value Imputation for Traffic-Related Time Series Data Based on a Multi-View Learning Method. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2933–2943. [[CrossRef](#)]
18. Van Der Voort, M.; Dougherty, M.; Watson, S. Combining kohonen maps with arima time series models to forecast traffic flow. *Transp. Res. Part C Emerg. Technol.* **1996**, *4*, 307–318. [[CrossRef](#)]
19. Meng, M.; Shao, C.F.; Wong, Y.D.; Wang, B.B.; Li, H.X. A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. *J. Cent. South Univ.* **2015**, *22*, 779–786. [[CrossRef](#)]
20. Ming, Z.; Satish, S.; Pawan, L. Short-Term Traffic Prediction on Different Types of Roads with Genetically Designed Regression and Time Delay Neural Network Models. *J. Comput. Civ. Eng.* **2005**, *19*, 94–103.
21. Dougherty, M.S.; Cobbett, M.R. Short-term inter-urban traffic forecasts using neural networks. *Int. J. Forecast.* **1997**, *13*, 21–31. [[CrossRef](#)]
22. Chen, D. Research on Traffic Flow Prediction in the Big Data Environment Based on the Improved RBF Neural Network. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2000–2008. [[CrossRef](#)]
23. Castro-Neto, M.; Jeong, Y.-S.; Jeong, M.-K.; Han, L.D. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [[CrossRef](#)]
24. Sun, Y.; Leng, B.; Guan, W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomputing* **2015**, *166*, 109–121. [[CrossRef](#)]

25. Bernaś, M.; Płaczek, B.; Porwik, P.; Pamuła, T. Segmentation of vehicle detector data for improved k-nearest neighbours-based traffic flow prediction. *IET Intell. Transp. Syst.* **2014**, *9*, 264–274. [[CrossRef](#)]
26. Seo, T.; Bayen, A.M.; Kusakabe, T.; Asakura, Y. Traffic state estimation on highway: A comprehensive survey. *Annu. Rev. Control* **2017**, *43*, 128–151. [[CrossRef](#)]
27. Wu, S.; Yang, Z.; Zhu, X.; Yu, B. Improved k-nn for short-term traffic forecasting using temporal and spatial information. *J. Transp. Eng.* **2014**, *140*, 1–9. [[CrossRef](#)]
28. Dell’acqua, P.; Bellotti, F.; Berta, R.; De Gloria, A. Time-Aware Multivariate Nearest Neighbor Regression Methods for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3393–3402. [[CrossRef](#)]
29. Sun, B.; Cheng, W.; Goswami, P.; Bai, G. Short-term traffic forecasting using self-adjusting k-nearest neighbours. *IET Intell. Transp. Syst.* **2018**, *12*, 41–48. [[CrossRef](#)]
30. Wang, J.; Deng, W.; Guo, Y. New Bayesian combination method for short-term traffic flow forecasting. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 79–94. [[CrossRef](#)]
31. Xu, Y.; Kong, Q.J.; Klette, R.; Liu, Y. Accurate and interpretable bayesian MARS for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2457–2469. [[CrossRef](#)]
32. Comert, G.; Bezuglov, A. An online change-point-based model for traffic parameter prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1360–1369. [[CrossRef](#)]
33. Liu, Y.; Liu, Z.; Jia, R. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part C Emerg. Technol.* **2019**, *101*, 18–34. [[CrossRef](#)]
34. Chen, E.; Ye, Z.; Wang, C.; Xu, M. Subway Passenger Flow Prediction for Special Events Using Smart Card Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, 1–12. [[CrossRef](#)]
35. Zheng, W.; Lee, D.H.; Shi, Q. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *J. Transp. Eng.* **2006**, *132*, 114–121. [[CrossRef](#)]
36. Jiang, X.; Adeli, H.; Asce, H.M. Dynamic Wavelet Neural Network Model for Traffic Flow Forecasting. *J. Transp. Eng. ASCE* **2005**, *131*, 771–779. [[CrossRef](#)]
37. Ouyang, J.; Lu, F.; Liu, X. Short-term urban traffic forecasting based on multi-kernel SVM model. *J. Image Graph.* **2010**, *15*, 1688–1695.
38. Kong, X.; Xu, Z.; Shen, G.; Wang, J.; Yang, Q.; Zhang, B. Urban traffic congestion estimation and prediction based on floating car trajectory data. *Futur. Gener. Comput. Syst.* **2016**, *61*, 97–107. [[CrossRef](#)]
39. Yang, Y.; Lu, H. Short-term traffic flow combined forecasting model based on SVM. In Proceedings of the 2010 International Conference on Computational and Information Sciences, Chengdu, China, 17–19 December 2010; pp. 262–265.
40. Ling, X.; Feng, X.; Chen, Z.; Xu, Y.; Haifeng, Z. Short-term traffic flow prediction with optimized Multi-kernel Support Vector Machine. In Proceedings of the Evolutionary Computation (CEC), San Sebastian, Spain, 5–8 June 2017; pp. 294–300.
41. Clark, P.; Niblett, T. The CN2 Induction Algorithm. *Mach. Learn.* **1989**, *3*, 261–283. [[CrossRef](#)]
42. Peterson, A.H.; Martinez, T.R. Reducing decision tree ensemble size using parallel decision dags. *Int. J. Artif. Intell. Tools* **2009**, *18*, 613–620. [[CrossRef](#)]
43. Hashemi, S.M.; Almasi, M.; Ebrazi, R.; Jahanshahi, M. Predicting the next state of traffic by data mining classification techniques. *Int. J. Smart Electr. Eng.* **2012**, *1*, 181–193.
44. Kumar, K.; Parida, M.; Katiyar, V.K. Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport* **2015**, *30*, 397–405. [[CrossRef](#)]
45. Sharma, B.; Kumar, S.; Tiwari, P.; Yadav, P.; Nezhurina, M.I. ANN based short-term traffic flow forecasting in undivided two lane highway. *J. Big Data* **2018**, *5*. [[CrossRef](#)]
46. Chhabra, A. Road Traffic Prediction Using KNN and Optimized Multilayer Perceptron. *Int. J. Appl. Eng. Res.* **2018**, *13*, 9843–9847.
47. Chen, Y.; Guo, Y.; Wang, Y. Modeling and density estimation of an urban freeway network based on dynamic graph hybrid automata. *Sensors* **2017**, *17*, 176. [[CrossRef](#)]
48. Zahid, M.; Chen, Y.; Jamal, A. Freeway Short-Term Travel Speed Prediction Based on Data Collection Time-Horizons: A Fast Forest Quantile Regression Approach. *Sustainability* **2020**, *12*, 646. [[CrossRef](#)]
49. Meinshausen, N. Quantile regression forests. *J. Mach. Learn. Res.* **2006**, *7*, 983–999.

50. Alajali, W.; Zhou, W.; Wen, S. Traffic flow prediction for road intersection safety. In Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; pp. 812–820.
51. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2012.
52. Larivière, B.; Van Den Poel, D. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Syst. Appl.* **2005**, *29*, 472–484. [[CrossRef](#)]
53. Murthy, K.V.S. *On Growing Better Decision Trees from Data*; The Johns Hopkins University: Baltimore, MD, USA, 1997.
54. Shotton, J.; Sharp, T.; Kohli, P.; Nowozin, S.; Winn, J.; Criminisi, A. Decision Jungles: Compact and Rich Models for Classification. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 234–242.
55. Huang, W.; Jia, W.; Guo, J.; Williams, B.M.; Shi, G.; Wei, Y.; Cao, J. Real-Time Prediction of Seasonal Heteroscedasticity in Vehicular Traffic Flow Series. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3170–3180. [[CrossRef](#)]
56. Bing, Q.; Qu, D.; Chen, X.; Pan, F.; Wei, J. Short-Term Traffic Flow Forecasting Method Based on LSSVM Model Optimized by GA-PSO Hybrid Algorithm. *Discret. Dyn. Nat. Soc.* **2018**, *2018*, 3093596. [[CrossRef](#)]
57. Jose, C.; Goyal, P.; Aggrwal, P.; Varma, M. Local deep kernel learning for efficient non-linear SVM prediction. *30th Int. Conf. Mach. Learn. ICML 2013* **2013**, *28*, 1523–1531.
58. Xianglou, L.I.U.; Dongxu, J.I.A.; Hui, L.I.; Ji-Yu, J. Research on Kernel parameter optimization of support vector machine in speaker recognition. *Sci. Technol. Eng.* **2010**, *10*, 1669–1673.
59. Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D.D. Hyperopt: A Python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **2015**, *8*, 014008. [[CrossRef](#)]
60. Takahashi, K. Remarks on SVM-based emotion recognition from multi-modal bio-potential signals. In Proceedings of the RO-MAN 2004. In Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759), Okayama, Japan, 20–22 September 2004; pp. 95–100.
61. Ghosh, A.; Danieli, M.; Riccardi, G. Annotation and prediction of stress and workload from physiological and inertial signals. *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS* **2015**, *2015-Novem*, 1621–1624.
62. Rastgoo, M.N.; Nakisa, B.; Nazri, M.Z.A. A hybrid of modified PSO and local search on a multi-robot search system. *Int. J. Adv. Robot. Syst.* **2015**, *12*. [[CrossRef](#)]
63. Nakisa, B.; Rastgoo, M.N.; Nasrudin, M.F.; Nazri, M.Z.A. A multi-swarm particle swarm optimization with local search on multi-robot search system. *J. Theor. Appl. Inf. Technol.* **2015**, *71*, 129–136.
64. Nakisa, B.; Rastgoo, M.N.; Norodin, M.J. Balancing exploration and exploitation in particle swarm optimization on search tasking. *Res. J. Appl. Sci. Eng. Technol.* **2014**, *8*, 1429–1434. [[CrossRef](#)]
65. Memon, M.Q.; He, J.; Yasir, M.A.; Memon, A. Improving efficiency of passive RFID tag anti-collision protocol using dynamic frame adjustment and optimal splitting. *Sensors* **2018**, *18*, 1185. [[CrossRef](#)]
66. Boser, E.; Vapnik, N.; Guyon, I.M.; Laboratories, T.B. Training Algorithm Margin for Optimal Classifiers. *Perception* **1992**, 144–152.
67. Clark, P.; Boswell, R. Rule induction with CN2: Some recent improvements. In *Proceedings of the Machine Learning—EWSL-91*; Kodratoff, Y., Ed.; Springer: Berlin/Heidelberg, Germany, 1991; pp. 151–163.
68. National Bureau of Statistics of China. *China Statistical Yearbook 2019*; 2019. Available online: <http://www.stats.gov.cn/english/> (accessed on 27 January 2020).
69. China's Major Cities Traffic Analysis Report. 2015. Available online: <https://gbtimes.com/china-reveals-its-top-10-most-traffic-congested-cities> (accessed on 29 November 2017).
70. Ni, X.Y.; Huang, H.; Du, W.P. Relevance analysis and short-term prediction of PM_{2.5} concentrations in Beijing based on multi-source data. *Atmos. Environ.* **2017**, *150*, 146–161. [[CrossRef](#)]
71. Al-Ahmadi, H.M.; Jamal, A.; Reza, I.; Assi, K.J.; Ahmed, S.A. Using Microscopic Simulation-Based Analysis to Model Driving Behavior: A Case Study of Khobar-Dammam in Saudi Arabia. *Sustainability* **2019**, *11*, 3018. [[CrossRef](#)]

72. Honghui, D.; Limin, J.; Xiaoliang, S.; Chenxi, L.; Yong, Q.; Min, G. Road traffic state prediction with a maximum entropy method. In Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, Seoul, Korea, 25–27 August 2009; pp. 628–630.
73. Manual, H.C. Highway Capacity Manual. Available online: <http://onlinepubs.trb.org/onlinepubs/trnews/rpo/rpo.trn129.pdf> (accessed on 18 January 2020).
74. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *Proceedings of the Advances in Artificial Intelligence*; Sattar, A., Kang, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1015–1021.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Abnormal Road Surface Recognition Based on Smartphone Acceleration Sensor

Ronghua Du ¹, Gang Qiu ¹, Kai Gao ^{1,2,*}, Lin Hu ¹ and Li Liu ¹

¹ College of Automotive and Mechanical Engineering, Changsha University of Science & Technology, Changsha 410114, China; csdrh@csust.edu.cn (R.D.); qiugang.px@stu.csust.edu.cn (G.Q.); hulin@csust.edu.cn (L.H.); lukeliuli@csust.edu.cn (L.L.)

² Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems, Changsha University of Science & Technology, Changsha 410114, China

* Correspondence: kai_g@csust.edu.cn

Received: 29 November 2019; Accepted: 9 January 2020; Published: 13 January 2020

Abstract: In order to identify the abnormal road surface condition efficiently and at low cost, a road surface condition recognition method is proposed based on the vibration acceleration generated by a smartphone when the vehicle passes through the abnormal road surface. The improved Gaussian background model is used to extract the features of the abnormal pavement, and the k-nearest neighbor (kNN) algorithm is used to distinguish the abnormal pavement types, including pothole and bump. Comparing with the existing works, the influence of vehicles with different suspension characteristics on the detection threshold is studied in this paper, and an adaptive adjustment mechanism based on vehicle speed is proposed. After comparing the field investigation results with the algorithm recognition results, the accuracy of the proposed algorithm is rigorously evaluated. The test results show that the vehicle vibration acceleration contains the road surface condition information, which can be used to identify the abnormal road conditions. The test result shows that the accuracy of the recognition of the road surface pothole is 96.03%, and the accuracy of the road surface bump is 94.12%. The proposed road surface recognition method can be utilized to replace the special patrol vehicle for timely and low-cost road maintenance.

Keywords: road surface recognition; Gaussian background model; abnormal road surface; acceleration sensor

1. Introduction

During the operation of the road, the road surface will inevitably suffer from some defects or damage due to the crushing, impact, and weather changes of the passing vehicles. These defects and damage are often referred to as abnormal road conditions [1]. Abnormal road conditions have a negative impact on vehicle speed, fuel consumption, mechanical wear, ride comfort, and even safety. The traditional abnormal road condition information collection mainly relies on the manual site survey and special patrol vehicle, which is inefficient and high in cost. According to the statistics of the Ministry of Transport of China, the maintenance expenditure of the national toll road in 2017 has reached 53.39 billion yuan.

In order to save money and time costs, many experts and scholars have studied abnormal road surface identification methods [2,3]. There are three main methods for identifying abnormal roads: Visual [4], three-dimensional reconstruction [5,6], and dynamic vehicle response. The method of road surface recognition based on visual or three-dimensional reconstruction has high-performance requirements and high cost, which is not conducive to comprehensive promotion and use. The test scope is also very limited. The method based on the smartphone acceleration sensor to identify the abnormal road surface has certain advantages [2].

In early research, accelerometers are used for pavement recognition. De Zoysa [7] proposed to deploy a small number of mobile sensors in the public transportation system to detect road conditions, but the system exhibited low detection efficiency and a false-positive rate. Chen [8] proposed a crowdsourcing based road surface monitoring system. By adding acceleration sensors and GPS modules to the vehicle to obtain the acceleration, velocity, and position information of the vehicle, the Gaussian background model is used to identify the abnormal road surface. Based on the research of Chen, Harikrishnan [9] improved the Gaussian background model and proposed an abnormal road surface recognition method that can adapt to different vehicle speeds. This method could classify the speed bump and road surface bumps according to the X-Z axis acceleration ratio.

In order to improve the accuracy of road surface recognition, Wang [10] proposed a method by fusing the feature data from an acceleration sensor and camera to identify the abnormal road type. Celaya [11] installed sensors at the front of the vehicle to obtain the vehicle vibration response when the vehicle passed the speed bump and used the multivariate genetic algorithm to detect the road surface anomaly. This method can realize the recognition of abnormal road surfaces with a low false alarm rate, but the calculation is complicated, and a large number of statistical features such as mean, variance, peak, and standard deviation are needed for machine learning. With the rapid development of mobile intelligent terminal technology, smartphones equipped with sensors such as accelerometers and global positioning navigation systems can be used to detect abnormal road surfaces [12]. Cong [13] used the probabilistic statistical method and wavelet analysis method to establish an identification model of abnormal data and uses median filtering and wavelet filtering to process the data so that the data detected by the smartphone can truly reflect the vibration of the vehicle. Yi [14] proposed a smartphone detection vehicle to monitor the road surface, using an anomaly indexing algorithm to detect the speed bump, the pit, and the manhole cover. However, in the subsequent experiments, it was found that the proposed algorithm can only identify the speed bump, and the identification of other abnormal road conditions is not favorable. Mukherjee [15] established a quarter-vehicle model and a half-vehicle model and studied the acceleration response of the vehicle when crossing the deceleration belt. They developed a mathematical statistics method to identify the deceleration belt. Zhao [16] proposed a feature extraction method combining time-domain parameter characteristics and wavelet packet energy characteristics based on vehicle suspension vibration response and used a probabilistic neural network to classify road surface.

In summary, the research on the road surface condition identification method has achieved certain results. Recent studies have proven that smartphone accelerometers can effectively capture vehicle vibrations caused by abnormal road surfaces. By analyzing the signals from these mobile sensors, we have the potential to identify road anomalies. In this study, a Gaussian background model is used to identify abnormal roads, and an adaptive adjustment mechanism based on vehicle speed is proposed to improve the recognition accuracy. The parameters of the Gaussian background model are optimized by using fuzzy logic inference machines, making the method suitable for different types of vehicles, and using the kNN algorithm to classify abnormal roads.

2. Road Information Sharing System

Obtaining abnormal road information in advance can effectively prevent traffic accidents, but due to road geometry, weather, lighting conditions, etc., the driver may not be able to notice the abnormal road ahead. The information-sharing technology is used to construct an abnormal road sharing system, especially encouraged by the development of the vehicle network and intelligent transportation system [17–21]. The road information sharing system is designed to promptly warn the driver when the driver approaches an abnormal road at a dangerous speed. In addition, the relevant information of the abnormal road surface is sent to the municipal road maintenance unit, so that the road maintenance personnel can timely get the road damage status and repair it to ensure the safety and comfort of travel [22–25].

In order to monitor the road surface conditions in realtime, it is necessary to find an efficient and reliable communication technology to transmit abnormal road information. This paper uses existing cellular network technology to collect abnormal road surface data. Figure 1 shows an architectural diagram of a road information sharing system. A smartphone with an acceleration sensor and a GPS module is fixed on the vehicle to obtain the latitude and longitude coordinates of the vehicle, the traveling speed, and the vibration acceleration data of the vehicle body. When the vehicle passes the abnormal road surface, the smartphone will upload the location and type of the abnormal road surface to the cloud. The abnormal road information is sent to the road maintenance personnel. When other vehicles approach the abnormal road surface, the cloud will issue an abnormal road surface reminder to ensure that the vehicle can pass the area safely and smoothly.

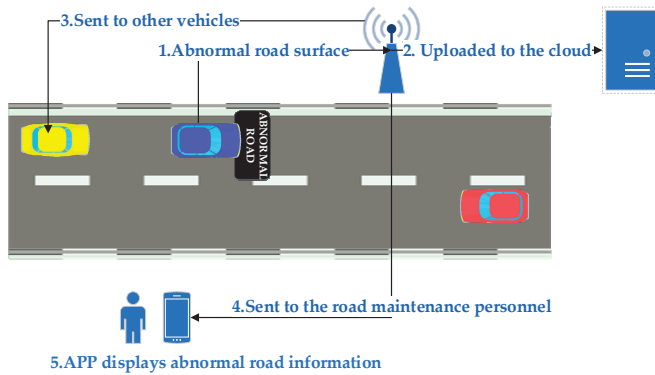


Figure 1. Road information sharing system architecture diagram.

3. Data Processing

There are many sources of vibration in a vehicle. Different vibration sources have certain differences in frequency domain characteristics. In this study, a quarter of the vehicle model and the abnormal road surface model are established to analyze the excitation effect of the abnormal road on the overall vehicle system. In order to obtain the vibration frequency range of the vehicle body, a frequency spectrum analysis of the vibration acceleration of the vehicle body is implemented, and a filter based on this analysis result is designed.

3.1. Vehicle Dynamics Analysis

In order to study the interaction between the vehicle and the road, a quarter-vehicle model is utilized to analyze the vibration of the vehicle in the vertical direction. The simplified vehicle dynamics model is shown in Figure 2.

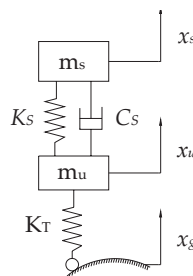


Figure 2. The quarter vehicle model.

m_u is the unsprung mass. m_s is the sprung mass. K_S is the stiffness of the spring. C_S is the damping coefficient of the shock absorber. K_T is the stiffness of the tire. x_s is the vertical displacement of the vehicle body. x_u is the vertical displacement of the wheel. x_g is the road surface excitation. The differential equation of motion of the vehicle is described as Equation (1).

$$\begin{cases} m_s \ddot{x}_s + C_S (\dot{x}_s - \dot{x}_u) + K_S (x_s - x_u) = 0 \\ m_u \ddot{x}_u - C_S (\dot{x}_s - \dot{x}_u) - K_S (x_s - x_u) + K_T (x_u - x_g) = 0 \end{cases} \quad (1)$$

This paper analyzes the dynamic response of the vehicle when the vehicle passes through the road surface at different speeds [26]. An abnormal pavement model is shown in Figure 3. The length of the abnormal road surface is L , and the height is h . v is the speed of the vehicle. t_1 and t_2 are the starting moment and the ending moment of the road surface excitation, respectively. The mathematical model is described as Equation (2) [27].

$$x_g(t) = \begin{cases} 0.5h \left(1 - \cos\left(\frac{2\pi v}{L}t\right)\right), t_1 \leq t \leq t_2 \\ 0, t_1 < t \text{ or } t > t_2 \end{cases} \quad (2)$$

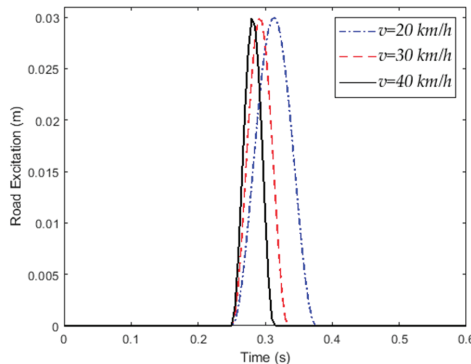


Figure 3. Abnormal road surface excitation at different speeds.

When the vehicle passes through an abnormal road at different speeds, the vertical acceleration response of the vehicle body is shown in Figure 4. The speed of the vehicle has a significant effect on the vertical acceleration of the vehicle. The frequency-domain analysis of the acceleration signal is performed to obtain a spectrogram of the vehicle vibration acceleration signal, as shown in Figure 5. As the vehicle speed increases, the frequency of body vibration increases, but its main component is still in the low-frequency range (30 Hz).

3.2. Butterworth Filter

When the data is collected, the position of the smartphone is not flat, and the slope of the test road surface may interfere with the recognition of the abnormal road surface. Therefore, the data needs to be filtered before using to recognize the abnormal road surface. According to the simulation data analysis results, the vibration caused by the abnormal road surface excitation is mainly distributed in the low-frequency range.

In this paper, the Butterworth filter is used to filter out the through component and the high-frequency noise with the frequency greater than 30 Hz. There are several reasons for choosing the Butterworth filter. Firstly, the Butterworth filter does not generate a ripple in the passband. Secondly, it has been successfully implemented in many commercial tools. Figure 6 shows the frequency response of the fifth-order Butterworth's numerical low-pass filters with a cut-off frequency of 30 Hz.

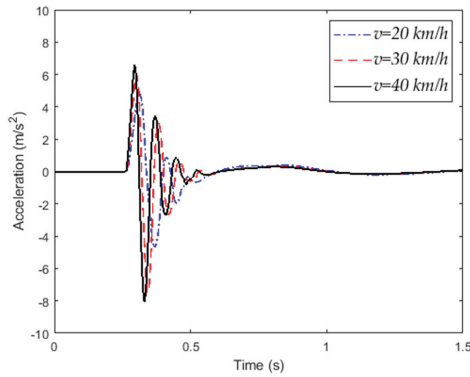


Figure 4. Vertical acceleration response of vehicle.

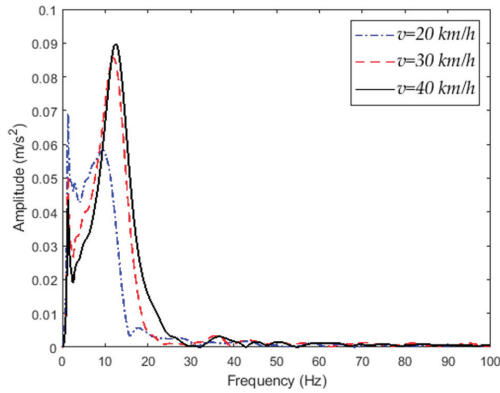


Figure 5. Vertical acceleration signal spectrum.

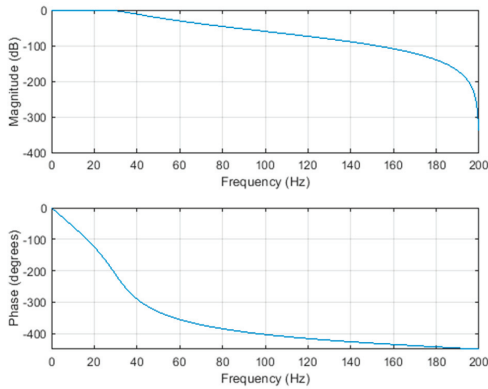


Figure 6. The frequency response of the Butterworth filter.

In addition, the filter delay has a negative impact on the positioning accuracy of abnormal roads surface. The filter delay will shift the positioning of the abnormal road surface towards the vehicle driving direction for a distance. However, because the filtering delay is very small (about 0.1 s), the positioning error caused by the filtering delay is not large, and the positioning accuracy requirements

for abnormal road recognition are not high. Therefore, the Butterworth filter can be applied to the recognition of abnormal roads.

4. Abnormal Road Surface Recognition

4.1. Overview of Abnormal Road Surface Recognition Algorithm

The framework of the abnormal road recognition algorithm is shown in Figure 7. In order to collect vehicle speed, acceleration, and position information, the smartphone's built-in accelerometer and global positioning navigation system are used. First, the raw data is preprocessed using a Butterworth filter. Secondly, the Gaussian background model is improved by using fuzzy logic control. The improved Gaussian model is combined with the acceleration threshold condition to extract the characteristic acceleration value caused by the abnormal pavement. Finally, the kNN algorithm is used to classify the abnormal road surface.

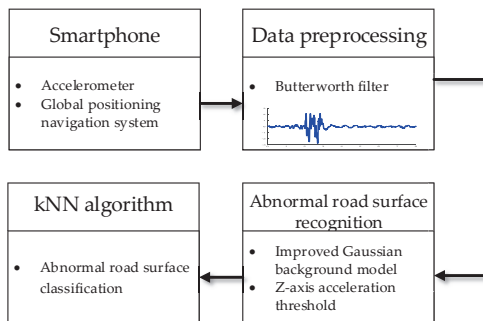


Figure 7. Road surface recognition algorithm framework.

4.2. Gaussian Background Model

When the vehicle is traveling on a flat road surface, the vibration acceleration in the vertical direction of the vehicle is Gaussian-distributed. In order to verify the conclusion, we used both Kolmogorov–Smirnov and Lilliefors test methods to test whether the acceleration generated by the vehicle on a flat road conforms to the Gaussian distribution. The results show that the assumption is valid. By measuring the vibration acceleration of the vehicle, the vehicle's vertical acceleration will be abrupt when the vehicle passes through an abnormal road surface compared to traveling on a flat road [28]. The Gaussian model is described as Equation (3).

$$\eta(z|\mu, \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} e^{\left(\frac{-(z-\mu)^2}{2\sigma^2}\right)} \quad (3)$$

μ is a mathematical expectation. σ is a standard deviation, and z is a body vibration acceleration value. If the vehicle passes over an abnormal road, the body acceleration at this time will no longer conform to the Gaussian distribution. Therefore, if z is the vertical acceleration of the vehicle caused by abnormal road excitation, the absolute value of the difference between z and μ is greater than the product of the threshold T_G and the standard deviation σ . The equation can be described as Equation (4).

$$|(z - \mu)| > T_G * \sigma \quad (4)$$

If Equation (4) is not satisfied, the vehicle is considered to be traveling on a flat road. At this time, the background of the Gaussian model will change, and the model parameters will be updated. The Gaussian model parameter update equation is described as Equation (5).

$$\begin{cases} \mu_{t+1} = (1 - \alpha)\mu_t + \alpha(z - \mu_t) \\ \sigma_{t+1}^2 = (1 - \alpha)\sigma_t^2 + \alpha(z - \mu_t)^2 \end{cases} \quad (5)$$

where α is the learning rate, and the size of the learning rate indicates the speed of the update. μ_{t+1} and σ_{t+1} are the updated mean and standard deviation.

4.3. Improved Gaussian Background Model

The vibration acceleration of the vehicle is affected by the traveling speed. When the vehicle passes over abnormal roads at a different speed, the amplitude of the vibration acceleration is different. In order to avoid false alarms at high speed and false negatives at low speed, the Gaussian model described as Equation (3) needs to be improved. After the improvement, if z is the vertical vibration acceleration caused by the abnormal road surface, it can be described as Equation (6).

$$|(z - \mu)| > \left(\frac{v}{T_V}\right) * T_G * \sigma \quad (6)$$

In addition, when the vehicle passes an abnormal road surface, the vertical acceleration of the vehicle is large. Therefore, if z is the acceleration caused by passing through the abnormal road, z will satisfy the following Equation (7).

$$z > \left(\frac{v}{T_V}\right) T_Z \quad (7)$$

Similar to the Gaussian background model, the improved Gaussian background model will also update the model parameters. When the acceleration z does not satisfy Equations (6) and (7), the data is considered as background data and will be used to update the background parameters. The parameter update formula is shown in Equation (5).

Where v is the current vehicle speed, and T_V is the speed threshold. Vehicle suspension parameters are different for different types of vehicles. The vibration response of the vehicles body will be significantly different when different types of vehicles pass through the same road surface. The original Gaussian background model has a fixed threshold T_Z , which is obviously not applicable to different types of vehicles. Therefore, this paper proposes designing a fuzzy logic controller to optimize the Gaussian background model.

Due to the complexity of the vehicle system, it is difficult to establish accurate mathematical models to describe the relationship between different vehicles and abnormal road surfaces. Fuzzy logic control is based on artificial experience and does not require an accurate mathematical model of the controlled object. Using fuzzy logic, the basic idea of optimizing the abnormal road surface recognition algorithm is: Firstly, find the fuzzy relationship between the road surface recognition algorithm parameter T_Z and the vehicle suspension parameters. Second, in the process of road surface recognition, according to the difference of K_S and C_S parameters of different vehicles, fuzzy logic is used to modify T_Z , so that the abnormal road surface recognition algorithm can adapt to different types of vehicles.

In this section, to make the abnormal road recognition algorithm applicable to different types of vehicles, a fuzzy logic inference machine is used to calculate an appropriate T_Z . The input quantities of the fuzzy logic inference machine are defined as the vehicle suspension stiffness K_S and damping C_S . The basic domain of K_S is [0,200], the basic domain of C_S is [0,4], and the basic domain of T_Z is [0.5,1].

As shown in Figure 8, the membership function is gaussmf type. The input 1 (K_S) is qualitative into five sets, denoted as NB, NS, ZO, PS, and PB. The input 2 (C_S) is qualitative into four sets, denoted as NB, NS, PS, and PB. The output (T_Z) is qualitative into three sets, denoted as NB, ZO, and PB, where NB is negative big, and PB is positive big.

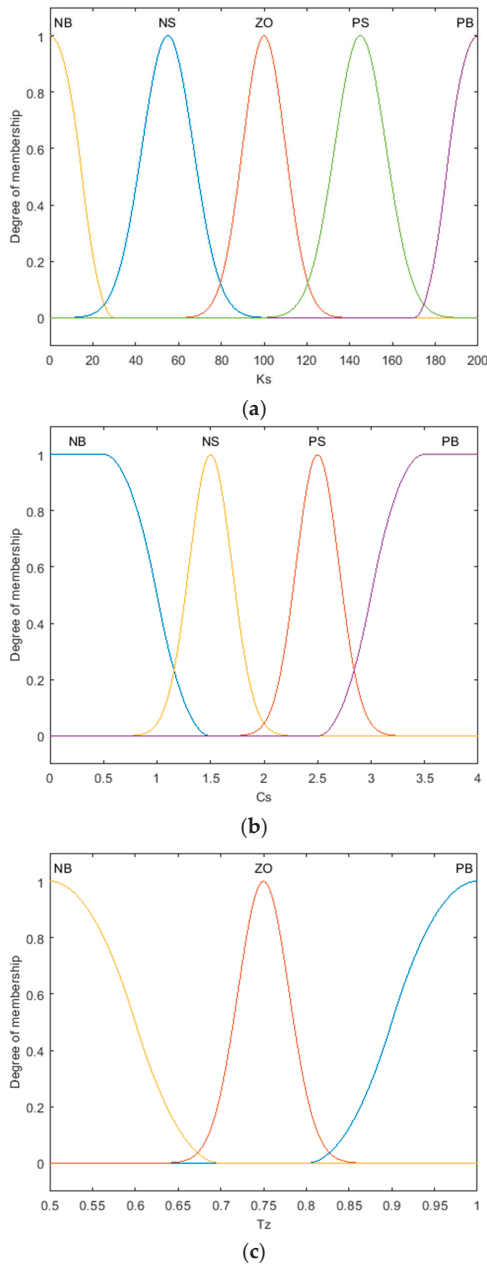


Figure 8. Membership function. (a) Membership function of input 1 (K_s). (b) Membership function of input 2 (C_s). (c) Membership function of the output (T_z).

The establishment of the rule bases of the fuzzy logic inference machine is based on the experiments and CarSim simulation. The rule bases of the fuzzy logic inference machine are shown in Table 1, and Figure 9 shows the interface of the output.

Table 1. Rule bases of T_Z .

| C_S \ K_S | NB | NS | ZO | PS | PB |
|---------------|----|----|----|----|----|
| NB | ZO | NB | NB | NB | NB |
| NS | NB | PB | NB | NB | NB |
| PS | ZO | NB | NB | ZO | NB |
| PB | ZO | NB | ZO | ZO | ZO |

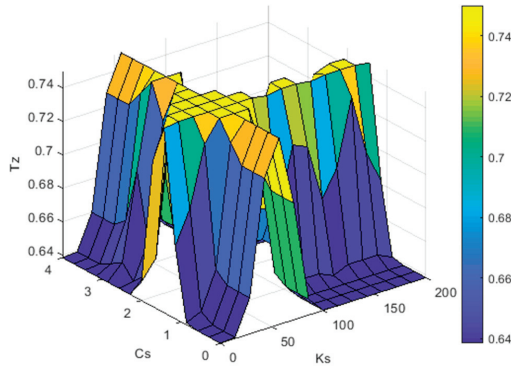


Figure 9. The surface of fuzzy logic controller’s output.

The improved Gaussian background model is described in Algorithm 1.

Algorithm 1: Abnormal road surface recognition method

Input: z , the Z-axis acceleration; v , the vehicle speed; K_S , the spring stiffness; C_S , the damping coefficient.

Output: $event_z$, Acceleration due to abnormal road surface.

1. **Algorithm begin:**
 2. $\mu = 0$ % μ is a mathematical expectation
 3. $\sigma = 0$ % σ is a standard deviation
 4. $T_G = 2$ % T_G is the Gaussian matching threshold
 5. $T_V = 20$ % T_V is the speed threshold
 6. if ($v > T_V$)
 7. $z_match \leftarrow \text{abs}(z - \mu)/\sigma$
 8. % calculating thresholds T_Z using fuzzy logic inference machines
 9. $T_Z \leftarrow \text{fuzzy_control}(K_S, C_S)$
 10. if ($z_match > T_G * v/T_V$) && ($\text{abs}(z) > (T_Z * v/T_V)$)
 11. $event_z \leftarrow z$
 12. else % update μ and σ , as described in Equation (5)
 13. $\mu \leftarrow (1 - \alpha) * \mu + \alpha * z$
 14. $\sigma \leftarrow \text{SQRT}((1 - \alpha) * \sigma^2 + \alpha * (z - \mu)^2)$ % SQRT means calculate square root
 15. end if
 16. end if
 17. return $event_z$
 18. **Algorithm end**
-

5. kNN Algorithm Abnormal Road Surface Classification

The kNN classifier is a sample-based machine learning algorithm. Due to the simple and effective characteristics of kNN, kNN has been widely used in engineering applications [29]. First, the abnormal road surface acceleration signal extracted by the abnormal road surface recognition algorithm is used to identify the k nearest neighbor values in the training data set. Then, the abnormal pavement labels

corresponding to these nearest neighbor values are counted, and the number of neighboring samples belonging to each possible type is calculated. The most common type of pavement that belongs to most of the k nearest neighbors is the type of pavement being measured.

5.1. Training and Testing Sample Data Sets

Both the training samples and the test samples are collected by the acceleration sensor of a smartphone. In this study, multiple sets of acceleration data caused by abnormal road surfaces are collected as training data sets. In addition, in order to maintain the independence between training data and test samples, the training samples and test samples are collected on two different roads. The number of training and test samples obtained in this paper is shown in Table 2.

Table 2. Number of training and testing sample.

| Road Surface | Training | Testing |
|--------------|----------|---------|
| Bump | 118 | 151 |
| Flat | 174 | 283 |
| Pothole | 103 | 68 |

From the perspective of the classification process, kNN most directly establishes a relationship between training samples and test samples, which can effectively avoid the negative impact caused by the improper selection of category features. Another widely used classification algorithm support vector machine (SVM) is utilized to compare with kNN on classifying the road surface. The same training and testing samples are used and the results are shown in Figure 10. The results show that kNN has an advantage over SVM in the classification of abnormal roads.

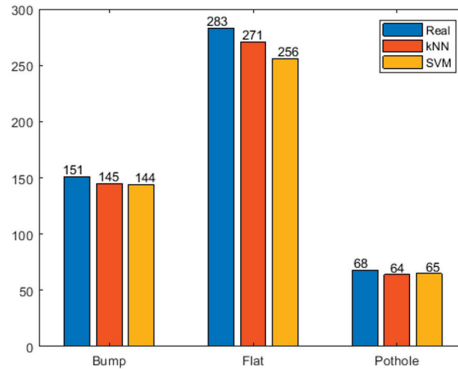


Figure 10. The performance of kNN and SVM.

5.2. Classification Algorithms and Tuning Parameters

The kNN classification algorithm classifies objects based on the attributes of k neighbors. The value of k is a key parameter of the kNN algorithm. This paper tests the classification effect when k takes different values in the range of 1–19 (odd numbers). Figure 11 shows the classification accuracy when k takes different values. It can be seen from the figure that as the value of k increases, the classification accuracy decreases, and the effect is best when $k = 3$. Therefore, k is set as 3 in this work.

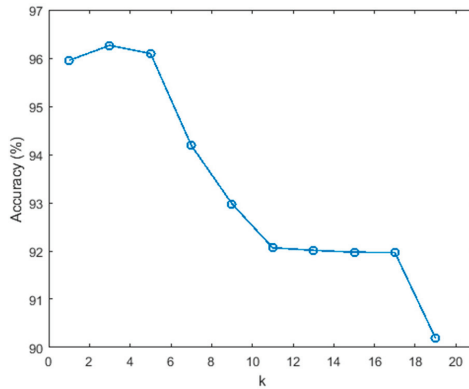


Figure 11. The relationship between classification accuracy (y-axis) and k value (x-axis).

6. Test and Analysis

6.1. Test Conditions

To validate the performance of the proposed algorithm, an A-class vehicle (Cavalier) and SUV (Qoros 5) are used to perform the test. An app working on a smartphone (Redmi Note 8 Pro) is developed to collect the acceleration, speed, and position. The sampling frequency is 400 Hz. According to the sampling theorem, if the frequency information of a signal is to be saved, the sampling frequency must be twice as much as the frequency of the measured object. If the amplitude information of a signal is to be saved, the sampling frequency is preferably ten times as much as the frequency of the measured object. The sampling frequency of vehicle speed and position is 1 Hz. This is because the GPS module data update frequency in the smartphone used in the experiment is 1 Hz. The smartphone is fixed on the handrail of the driver’s seat during the test, as shown in Figure 12.

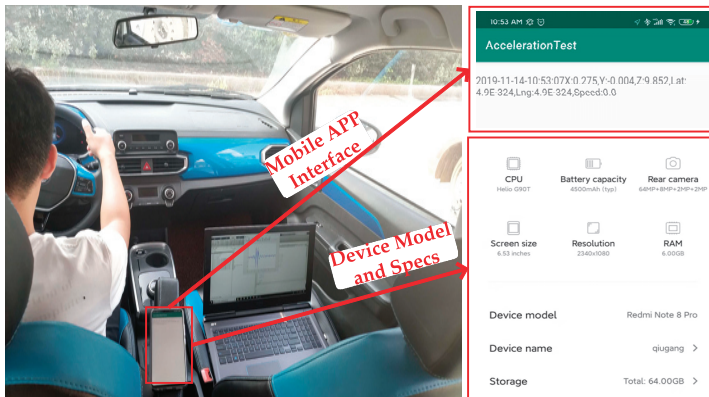


Figure 12. Smartphone installation location.

The vehicle travels at a different speed on an abnormal road and a flat road and performs multiple tests. There are many abnormal road surfaces on the test road, as shown in Figure 13. The actual measurement shows that the area of the pothole on the road is about 1 square meter and the maximum depression depth is 30 mm. The diameter of the raised manhole cover is 700 mm, and the height is 50 mm. The width of the speed bump is 350 mm, and the height is about 30 mm.

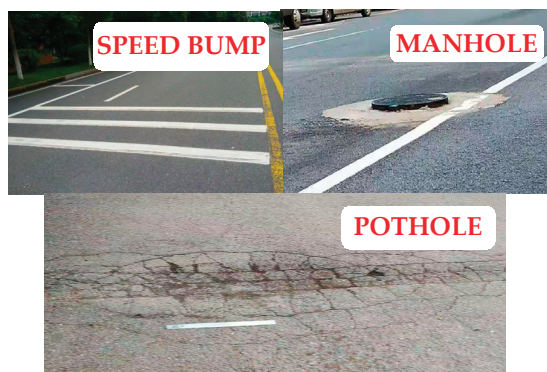


Figure 13. Abnormal road surfaces on the test road.

6.2. Vehicle Dynamic Response under Different Road Excitations

When the vehicle is travelling on a flat road surface, its vibration acceleration is relatively stable. When the vehicle passes through an abnormal road surface such as a road pit or a road surface bump, the acceleration of the vehicle in the vertical direction changes significantly. Figure 14 shows the vibration acceleration of the vehicle when the vehicle passes through different roads at a speed of 30 km per hour. Figure 14a shows the vehicle traveling on a flat road surface. In Figure 14b, the vehicle is excited by the pothole road surface. In Figure 14c, the vehicle is excited by the bump road surface. Under the influence of gravity acceleration, the Z-axis acceleration fluctuates around 9.8.

After the original acceleration data is processed by the Butterworth filter, the filtered Z-axis vibration acceleration shown in Figure 15 is obtained. It can be clearly observed that the valid data collected by the smartphone is retained, and the through component and the high-frequency noise are eliminated.

6.3. Test Results and Analysis

Figure 16 depicts the vertical acceleration values of the vehicle body when the experimental vehicle passes through the same abnormal road surface at different vehicle speeds. The blue triangle in Figure 16 represents the experimental data of the A-class vehicle, and the red rectangle represents the experimental data of the SUV.

It can be observed from Figure 16 that as the vehicle speed increases, the vertical vibration acceleration of the vehicle body also increases. In addition, when the different types of vehicles pass the same abnormal road surface, the vertical vibration acceleration of the vehicle body also has a significant difference.

Comparing the field measurement results with the algorithm identification results, as shown in Table 3, the results show that the proposed method can effectively identify the road surface potholes and bumps. In the 68 sets of road surface pothole data, 64 groups are successfully identified and classified, and the accuracy rate is 94.12%. Among the 151 sets of road surface bumps data, 145 groups are successfully identified and classified, and the accuracy rate is 96.03%. Through onsite investigation of the reported road surface, it is found that the abnormal road surface size of the false alarm is small, or the multiple abnormal road surfaces are close to each other, which leads to erroneous recognition results.

Table 3. Abnormal road surface identification result.

| Road Surface | Field Measurement | Algorithm Identification | Accuracy Rate |
|--------------|-------------------|--------------------------|---------------|
| Pothole | 151 | 145 | 96.03% |
| Bump | 68 | 64 | 94.12% |

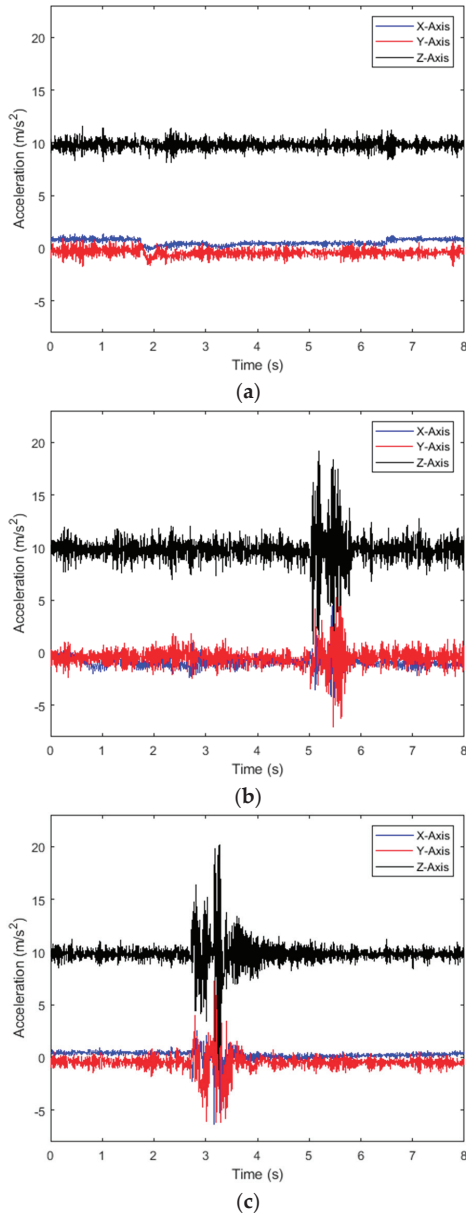
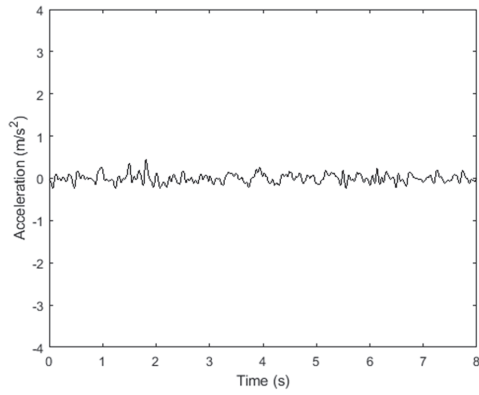
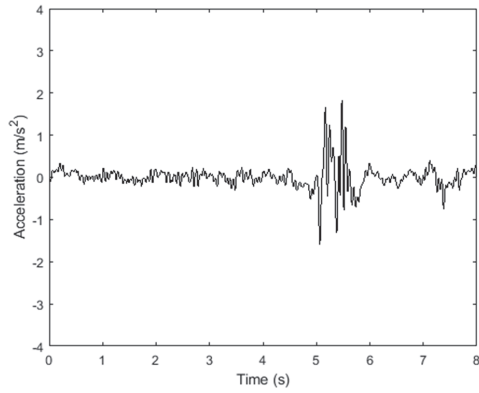


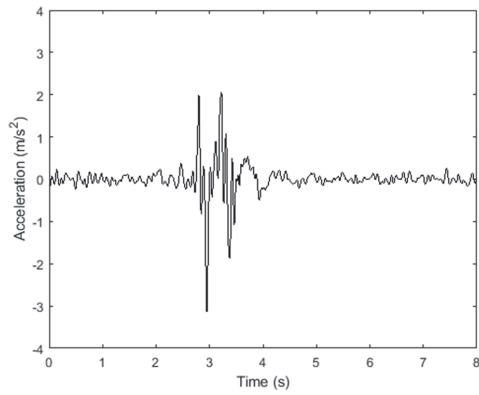
Figure 14. Vehicle vibration acceleration signal. (a) Flat road surface; (b) pothole road surface; (c) bump road surface.



(a)



(b)



(c)

Figure 15. Z-axis vibration acceleration after filtering. (a) Flat road surface; (b) pothole road surface; (c) bump road surface.

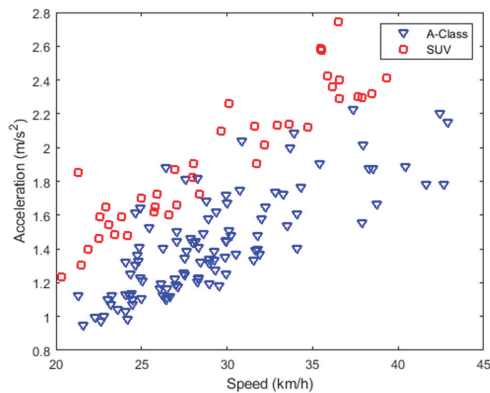


Figure 16. Vehicle speed and body vibration acceleration.

7. Conclusions

In this paper, a method for abnormal road surface recognition using a smartphone acceleration sensor is proposed. The Gaussian background model is optimized by a fuzzy logic inference machine so that the road surface recognition algorithm can be applied to different types of vehicles. Vehicle acceleration, speed, and position data are collected by the built-in acceleration sensor and global positioning navigation system of the smartphone. The vibration acceleration caused by the abnormal road surface is extracted using the improved Gaussian background model and the Z-axis acceleration threshold condition. An adaptive adjustment mechanism based on vehicle speed is proposed to improve the recognition accuracy. The classification of the abnormal road surface is realized by utilizing the kNN classification algorithm. Multiple sets of samples are used to test the abnormal road surface identification method. Comparing the algorithm identification results with the artificial site survey results, it is found that the proposed method can effectively identify and classify abnormal road surfaces such as potholes and bumps.

It is worth noting that with the increase of the total mileage of the road, the intelligent transportation system will be more and more widely used in the transportation industry. In this paper, only the two main types of the abnormal road surface are identified. The next step would be studying the evaluation and identification methods of the degree and size of abnormal road surface damage.

Author Contributions: R.D. and G.Q. conceived the idea; K.G. and G.Q. designed the experiments and analyzed the data; G.Q. implemented the experiments and wrote the manuscript; L.H. and L.L. helped with the simulation. R.D. and K.G. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61973047, 51678077, 51875049, and also by the Science Fund for Distinguished Young Scholars of the Hunan Province (2019JJ20017). The project was supported by the Open Fund of Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems (kf190701) (Changsha University of Science & Technology).

Acknowledgments: The authors would like to thank Changsha Intelligent Driving Research Institute for their help in the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sha, A.M.; Tong, Z.; Gao, J. Recognition and Measurement of Road Surface Disasters Based on Convolutional Neural Networks. *Chin. J. Highway Transp.* **2018**, *31*, 1–10.
2. Mohan, P.; Padmanabhan, V.N.; Ramjee, R. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 5–7 November 2008; pp. 323–336.

3. Wahlström, J.; Skog, I.; Händel, P. Smartphone-based vehicle telematics: A ten-year anniversary. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2802–2825. [[CrossRef](#)]
4. Quintana, M.; Torres, J.; Menendez, J.M. A Simplified Computer Vision System for Road Surface Inspection and Maintenance. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 608–619. [[CrossRef](#)]
5. Li, W.; Burrow, M.; Li, Z. Automatic Road Condition Assessment by Using Point Laser Sensor. In Proceedings of the 2018 IEEE SENSORS, Montreal, QC, Canada, 28–31 October 2018; pp. 1–4.
6. Aki, M.; Rojanaarapa, T.; Nalano, K. Road Surface Recognition Using Laser Radar for Automatic Platooning. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2800–2810. [[CrossRef](#)]
7. De, Z.K.; Keppitiyagama, C.; Seneviratne, G.P. A Public Transport System Based Sensor Network for Road Surface Condition Monitoring. In Proceedings of the 2007 Workshop on Networked Systems for Developing Regions, Kyoto, Japan, 2 August 2007; p. 9.
8. Chen, K.; Lu, M.; Tan, G. CRSM: Crowdsourcing Based Road Surface Monitoring. In Proceedings of the 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, Zhangjiajie, China, 13–15 November 2013; pp. 2151–2158.
9. Harikrishnan, P.M.; Gopi, V.P. Vehicle Vibration Signal Processing for Road Surface Monitoring. *IEEE Sens. J.* **2017**, *17*, 5192–5197. [[CrossRef](#)]
10. Wang, S.F.; Du, K.Y.; Meng, Y. Machine Learning-Based Road Terrain Recognition for Land Vehicles. *Acta Armamentarii* **2017**, *38*, 1642–1648.
11. Celaya, P.J.; Galvan, T.C.; Lopez, M.F. Speed Bump Detection Using Accelerometric Features: A Genetic Algorithm Approach. *Sensors* **2018**, *18*, 443. [[CrossRef](#)]
12. Li, Y.H.; Zhang, Y.T. Research of Intelligent Patrol of Road Surface Abnormality under the Background of Intelligent Transportation—From Perspective of Crowd-sourcing and DTW Technology. *Sci. Technol. Prog. Policy* **2018**, *35*, 93–97.
13. Cong, J.L.; Wang, Y.; Yang, C.P. Data Preprocessing Method of Vehicle Vibration Acceleration by Smartphone. *J. Data Acquis. Process.* **2019**, *34*, 349–357.
14. Yi, C.W.; Chuang, Y.T.; Nian, C.S. Toward Crowdsourcing-Based Road Pavement Monitoring by Mobile Sensing Technologies. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1905–1917. [[CrossRef](#)]
15. Mukherjee, A.; Majhi, S. Characterisation of Road Bumps Using Smartphones. *Eur. Transp. Res. Rev.* **2016**, *8*, 13. [[CrossRef](#)]
16. Zhao, K.; Dong, M.M.; Zhao, F. Study on Terrain Classification Based on Vehicle Suspension Vibration. *Trans. Beijing Inst. Technol.* **2018**, *38*, 155–159.
17. Gao, K.; Han, F.; Dong, P.; Xiong, N.; Du, R. Connected Vehicle as a Mobile Sensor for Real Time Queue Length at Signalized Intersections. *Sensors* **2019**, *19*, 2059. [[CrossRef](#)] [[PubMed](#)]
18. Li, W.; Xu, H.; Li, H.; Yang, Y.; Sharma, P.; Wang, J.; Singh, S. Complexity and Algorithms for Superposed Data Uploading Problem in Networks with Smart Devices. *IEEE Internet Things J.* **2019**. [[CrossRef](#)]
19. Hu, L.; Zhong, Y.; Hao, W. Optimal Route Algorithm Considering Traffic Light and Energy Consumption. *IEEE Access* **2018**, *6*, 59695–59704. [[CrossRef](#)]
20. Hao, W.; Lin, Y.; Cheng, Y. Signal Progression Model for Long Arterial: Intersection Grouping and Coordination. *IEEE Access* **2018**, *6*, 30128–30136. [[CrossRef](#)]
21. Wu, W.; Huang, L.; Du, R. Simultaneous Optimization of Vehicle Arrival Time and Signal Timings within a Connected Vehicle Environment. *Sensors* **2020**, *20*, 191. [[CrossRef](#)]
22. Zeng, Q.; Gu, W.; Zhang, X. Analyzing freeway crash severity using a Bayesian spatial generalized ordered logit model with conditional autoregressive priors. *Accid. Anal. Prev.* **2019**, *127*, 87–95. [[CrossRef](#)]
23. Hao, W.; Daniel, J. Motor vehicle driver injury severity study under various traffic control at highway-rail grade crossings in the United States. *J. Saf. Res.* **2014**, *51*, 41–48. [[CrossRef](#)]
24. Hao, W.; Kamga, C.; Yang, X. Driver injury severity study for truck involved accidents at highway-rail grade crossings in the United States. *Transp. Res. Part F Traffic Psychol. Behav.* **2016**, *43*, 379–386. [[CrossRef](#)]
25. Salau, I.B.; Onumanyi, A.J.; Onwuka, E.N. A Survey of Accelerometer Based Techniques for Road Anomalies Detection and Characterization. *Int. J. Eng. Sci. Appl.* **2019**, *3*, 8–20.
26. Hu, L.; Hu, X.; Wan, J. The injury epidemiology of adult riders in vehicle-two-wheeler crashes in China, Ningbo, 2011–2015. *J. Saf. Res.* **2019**. [[CrossRef](#)]

27. Tan, R.H.; Chen, Y.; Lu, Y.X. The Mathematical Models in Time Domain for the Road Disturbances and the Simulation. *China J. High Way Transp.* **1998**, *11*, 98–104.
28. Rishiwal, V.; Khan, H. Automatic Pothole and Speed Breaker Detection Using Android System. In Proceedings of the 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 30 May–3 June 2016; pp. 1270–1273.
29. Thanh, P.N.; Kappas, M. Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors* **2018**, *18*, 18. [[CrossRef](#)] [[PubMed](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Design of a Human Evaluator Model for the Ride Comfort of Vehicle on a Speed Bump Using a Neural Artistic Style Extraction

Donggyun Kim ¹, MyeonGyu Jeong ², ByungGuk Bae ² and Changsun Ahn ^{1,*}

¹ School of Mechanical Engineering, Pusan National University, Busan 46241, Korea; donggyunkim@pusan.ac.kr

² Research & Development Division, Hyundai Motor Company, Hwaseong-si, Gyeonggi 18280, Korea; creator@hyundai.com (M.J.); bkbbae@hyundai.com (B.B.)

* Correspondence: sunahn@pusan.ac.kr; Tel.: +82-51-510-2979

Received: 10 November 2019; Accepted: 5 December 2019; Published: 8 December 2019

Abstract: The subjective evaluation of vehicle ride comfort is costly and time-consuming but is crucial for vehicle development. To reduce the cost and time, the objectification of subjective evaluation has been widely studied, and most of the approaches use a regression model between objective metrics and subjective ratings. However, the accuracy of these approaches is highly dependent on the selection of the objective metrics. In most of the methods, it is not clear that the selected metrics are sufficiently significant or whether all significant metrics are included in the selection. This paper presents a method to build a correlation model between measurements and subjective evaluations without using predefined features or objective metrics. A numerical representation of ride comfort was extracted from raw signals based on the idea of the artistic style transfer method. The correlation model was designed based on the extracted numerical representation and subjective ratings. The model has a much better accuracy than any other correlation models in the literature. This better accuracy is contributed to not only by using a neural network, but also by the extraction of the numerical representation of ride comfort using a pre-trained neural network.

Keywords: deep neural network; neural artistic extraction; objectification; ride comfort; subjective evaluation

1. Introduction

The evaluation of the quality of goods is based on human feelings in many areas, such as works of art [1,2], consumer electronics [3,4], cars [5,6], and other products [7,8]. Subjective evaluations are as important as objective metrics for engineering products because consumers evaluate the value of products based on their feelings. However, subjective evaluations are easily affected by the experience level of the evaluator, geographical locations, and the time of evaluation. Furthermore, subjective evaluations generally take a longer time than an objective evaluation. Therefore, subjective evaluation has a lower consistency and lower repeatability than measurement-based objective evaluations. Subjective evaluations also require a finished product, so feedback from a subjective evaluation cannot be considered in the beginning phase of the product design process.

Much effort has been made to take into account subjective evaluations in earlier design phases. A mockup can be used if the subjective evaluations are based on visual and tactile aspects, such as the hands-on feel of handheld devices [9]. If the evaluations are based on a comprehensive feeling, a designer can use a correlation model between the objective and subjective evaluations with a mathematical product model that generates virtual measurements for an objective evaluation, such as vehicle ride comfort [10,11], vehicle steering feel [12,13], and art education [14].

In the automotive industry, objective evaluations of ride comfort and steering feel are inevitable and crucial during the development process. Furthermore, they cannot be substituted with any other objective evaluation metrics because customers judge cars through their feel. Subjective evaluations of a car can only be performed in the very last phase of the development process, so engineers have very little chance to modify the design parameters when the subjective evaluations become available.

Professional evaluators or special test drivers are required for subjective evaluations, which can be performed only several times per day because modification of the testing car or the testing environment takes time. Due to these constraints, several approaches have been suggested to predict the results of a subjective evaluation using values measured from a real car or a simulation model. Some of examples are linear regression. Data et al. [15] investigated a method to find the best correlation between objective metrics and subjective ratings. Rothhämel et al. [16] proposed a method to find correlations using a driving simulator and a vehicle model. Nybacka and coworkers [17,18] identified the links between objective metrics using a steering robot and the subjective evaluation of expert drivers. Gil Gómez and coworkers [19,20] found correlations between objective metrics. The other methods that have been used are nonlinear regression including a fuzzy model [21] and an artificial neural network [17–20,22]. In References [17–20], a simple neural network with two hidden layers is used. Liu et al. [22] investigated a method to find correlations between signals measured by an electromyogram and subjective evaluations. However, the methods have shown a low correlation and are effective for only limited cases because a designer selects the correlation variables between the objective measurements and the subjective evaluation, so the sufficiency and optimality of the selection are not clear. Furthermore, the generality of such correlation models is not guaranteed due to the small number of evaluation data.

Deep neural network (DNN) techniques are actively applied to the design of correlation models between objective measurements and subjective evaluations. For subjective video quality evaluations, a subjective video quality prediction model was introduced based on a DNN. Varga [23] evaluated video quality through a DNN architecture consisting of a pre-trained network, transfer learning, temporal pooling, and regression layers. In the medical field, DNN techniques are also widely used. Mahendran et al. predicted major depressive disorder using a weighted average ensemble machine learning model [24]. Weber et al. [25] calculated muscle fat infiltration using a previously developed convolution neural network. In the material field, Yao et al. used the neural network model to predict subjective tactile properties from objective test results of porous polymeric materials [26].

In the automotive field, an interesting result has been reported on regarding modeling the correlation between objective and subjective evaluations of vehicle dynamic performance using a DNN technique [27]. A method was presented to identify the relationship between the objective metrics and subjective assessments. A quite meaningful correlation model was generated and can foresee the subjective characteristics of a new vehicle based on a simulation and measurements. The correlation model was trained using 22 test drivers with 51 vehicles. The number of training sets did not seem to be sufficient compared to general DNN cases. However, the number was quite large when considering the number of test drivers and cars used for a general vehicle evaluation. For typical subjective evaluations, a few test drivers drive a few cars (a test car and few reference cars), and the process takes several days. A lack of sufficient datasets is a major difficulty in applying DNN techniques for the objectification of the subjective evaluation of cars. Even though the results were quite impressive and successful, the robustness and generality of the model are not clear due to the small training datasets.

Another weakness of this approach is that the inputs to the model are predefined objective metrics, such as the yaw gain, torque dead band, and phase time lag. The use of predefined metrics raises questions about the appropriate definitions of the metrics, the selection of the best metrics, and whether the selected metrics represent human perceptions well.

Not all artificial intelligent techniques require a large datasets. Mordvintsev et al. [28] modified an input image such that the output from a given pre-trained neural network would be as close to the expected output as possible. This method is called Deep Dream and has been used to synthesize

two images to create a new image. An image synthesis method called artistic style transfer was also developed [29], which synthesizes two images by transforming the style of one image to the other image. This technique needs only two images: one for the style and the other for the content.

In this method, the network parameters are not optimized, and transforming the style of the input image requires a recursive computational or training process. The network structure and parameters use those of a pre-trained network, such as VGG-19 [30]. VGG-19 is a pre-trained convolution neural network that was developed by the Visual Geometry Group of the University of Oxford for classification tasks. This method numerically extracts the style of an image that is recognized by human senses from a raw image without using predefined metrics, such as lines or edges.

This paper presents a method to build a correlation model between measurements and subjective evaluations without using predefined features or objective metrics, as shown in Figure 1. The key idea is that a numerical representation of ride comfort is extracted from raw signals that were measured in a test vehicle without preprocessing to define and calculate objective metrics. This method is based on the ideas of the artistic style transfer method. The proposed method was applied to the evaluation of ride comfort when a vehicle passes over a speed bump. A comparative model is proposed for the ride comfort of two vehicles to minimize the effect of using a small dataset. The input of the model is the measurements from the two vehicles, and the output is the differences in their subjective ratings of ride comfort.

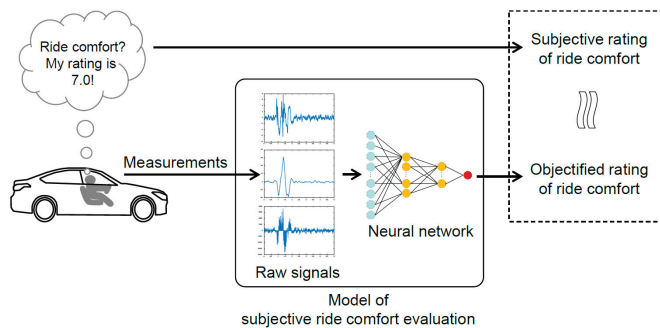


Figure 1. Objectification of the subjective evaluation of ride comfort.

The rest of the paper is organized as follows. Section 2 presents the method for the objectification of subjective evaluation, and Section 3 presents the results of the model training. Section 4 suggests possible applications of the model, and Section 5 concludes the paper.

2. Method

2.1. Data Collection

The data used in this study were collected from ride comfort evaluations by an automotive manufacturer. In the test, a vehicle was driven at 30 km/h on a road with a speed bump, as shown in Figure 2. The test vehicle was a small sedan with several sensors. Accelerometers were installed at several positions of the vehicle to measure three-dimensional accelerations at those positions. Gyroscopes were also installed at several positions to measure three-dimensional angular rates. The front and rear dampers had three adjustable settings: hard, medium, and soft. A professional test driver evaluated the vehicle with nine combinations of dampers.

The subjective rating was reported in the form of absolute ratings defined in SAE J1441 [31], which are presented in Table 1. The ratings given by the test driver are shown in Table 2. The test driver evaluated the ride comfort in two categories: primary ride and impact comfort. Fractional expressions

were used to obtain a finer resolution, which were transformed to numbers; for example, “6+” was changed to 6.33, “6+ to 7” was changed to 6.5, and “6 to 6+” was changed to 6.16.

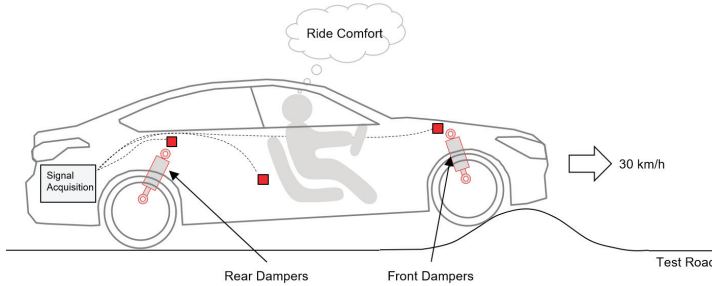


Figure 2. Ride comfort evaluation on a road with a speed bump.

Table 1. Subjective rating scale.

| Rating Scale | Event Type | |
|--------------|----------------|-----------|
| | Disturbance | Control |
| 10 | Imperceptible | Excellent |
| 9 | Trace | |
| 8 | A Little | Good |
| 7 | Some | |
| 6 | Moderate | Fair |
| 5 | Borderline | |
| 4 | Annoying | Poor |
| 3 | Strong | |
| 2 | Severe | Very Poor |
| 1 | Not Acceptable | |

Most ride issues were disturbance events, while most handling issues were control events.

Table 2. Subjective evaluation results of ride comfort.

| Damper Settings (Front/Rear) | Primary Ride | Impact (Secondary Ride) |
|------------------------------|--------------|-------------------------|
| H/H | 6+ | 6 |
| H/M | 6+ to 7- | 6+ |
| H/S | 6 | 6 to 6+ |
| M/H | 7- | 6+ |
| M/M | 6+ | 6+ |
| M/S | 6 | 6+ |
| S/H | 6 | 6 to 6+ |
| S/M | 6 to 6+ | 6+ |
| S/S | 6+ | 6+ to 7- |

H, M, and S stand for hard, medium, and soft, respectively.

The objective data were collected using several sensors and included the velocities, accelerations, and forces at several positions on the test vehicles. A total of 120 types of sensor signals were collected. The driver kept the speed at 30 km/h as much as possible to avoid disturbances from speed differences. The test driver drove the vehicle multiple times for each damper setting, and four sets of objective data were collected for each setting. Therefore, the total number of datasets was 36.

2.2. Ride Comfort Evaluation Model

The basic concept of the model is as follows. First, an image was created by combining the spectrograms of measured signals, and then gram matrices were extracted from the image as numerical representations of ride comfort. An artificial neural network was then trained to find a relationship between the numerical representations and the subjective ratings. Based on this concept, we designed a model structure to predict the difference of the subjective ratings between two vehicles rather than predicting the absolute rating of ride comfort, as shown in Figure 3. A comparative model was designed to increase the number of training datasets.

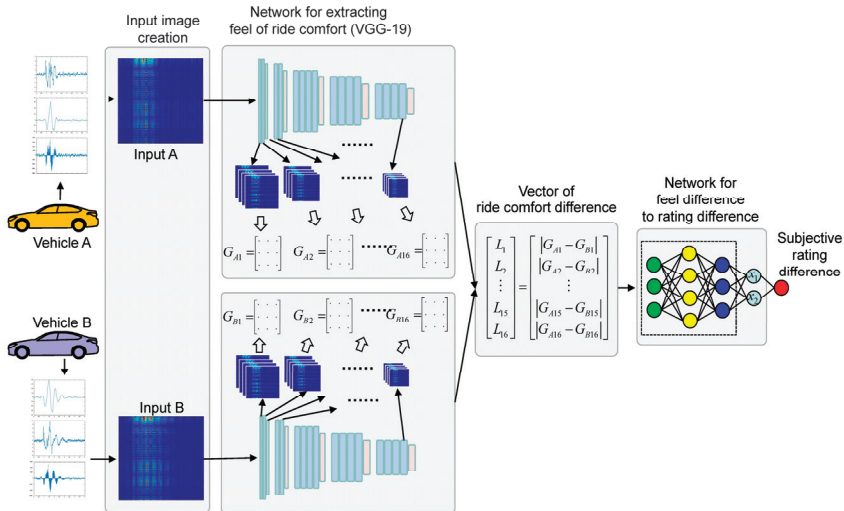


Figure 3. Structure of ride comfort evaluation model. VGG: Visual Geometry Group.

Two artificial neural networks were used in the model: one for extracting a numerical representation of the ride comfort and another for the correlation model between the extracted numerical representation and the subjective ratings. Extracting ride comfort did not require any datasets for training because the extraction was performed by a pre-trained convolutional neural network (CNN). However, building the correlation model required datasets for training. The 36 datasets collected were not sufficient. We designed a comparative model to increase the number of training sets. The input was the difference between the numerical representations of ride comfort that were extracted from the measurements of two vehicles, and the output was the difference between the vehicles' subjective ratings.

2.2.1. Extraction of Ride Comfort from Measurements

Instead, the style was implicitly defined in the pre-trained network. The numerical representation of the style of an image is a set of gram matrices, $[G_1, G_2, \dots, G_{16}]$, of the filter responses in the layers of VGG-19, as shown in Figure 4. Each layer produces an abstract concept of the image. In the CNN, each layer further abstracts the pixel representations of the image.

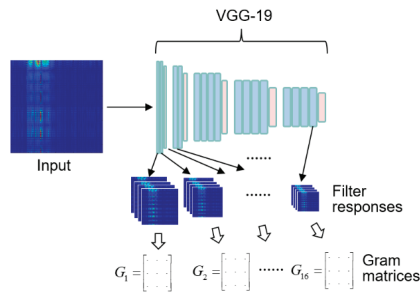


Figure 4. Extracting a numerical representation of ride comfort based on the ideas of the artistic style transfer algorithm. The network input is an image transformed from the measured temporal data.

The algorithm uses the filter responses at all layers as a representation of the contents of the original image. The gram matrices of the filter responses of all layers are used as a representation of the artistic style of the original image. The content and style are separate, and the data dimensions of the content and style are larger than the dimensions of the original image because they are expressed with several levels of abstraction. Using the same method, the gram matrices of the filter responses of all layers were extracted as a numerical representation of the style or feel of ride comfort. Extraction of the ride comfort did not require any modification of the networks. The only difference between the extraction of ride comfort and the extraction of the artistic style of an image was the nature of the input data. The input to the former was the measured temporal data, and the input to the latter was spatial data or an image. The temporal data were transformed into an image to use the pre-trained CNN without re-training the network.

2.2.2. Preprocessing Input Data

A spectrogram of the temporal data was used to transform it into an image. Sometimes called a color map, a spectrogram is a visual representation of the spectrum of frequencies of a temporal signal as it varies with time. It effectively shows signal patterns in both the frequency domain and the time domain simultaneously. Most objective metrics for ride comfort are defined using characteristics in the frequency domain or in the time domain, which makes a spectrogram a good visual representation of the possible metrics of ride comfort.

Another issue in the transformation of the temporal data is that multiple spectrograms are generated from the temporal data because the data are a set of signals. A single image was input to the VGG-19 network, and thus multiple spectrogram images must be combined to feed them to the network. As shown in Figure 5, the spectrogram images were stacked line by line to strengthen the spectral and temporal correlations of the input data. Before the transformation, all signals were normalized so that the ranges of the values were between -1 and 1 . Furthermore, the spectrograms were clipped along the frequency axis to remove the signal noise and bias.

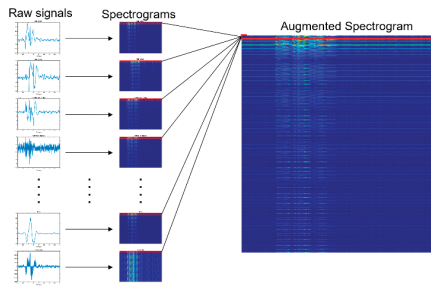


Figure 5. Preprocessing input data. Raw measured signals were transformed into an image for input.

2.2.3. Vector of Ride Comfort Difference

Once the numerical representation of ride comfort was extracted, a neural network could be built to map from the numerical representation, $[G_1, G_2, \dots, G_{16}]$, to the subjective rating. The output of the comparative model was the difference between the subjective ratings of two vehicles, and the input was the difference between the two numerical representations of the two vehicles. The set of norms of the gram matrix difference was defined as a vector of the ride comfort difference, as shown in Figure 6. This approach expanded the number of training sets from 36 to ${}_{36}C_2 = 630$, as shown in Figure 7.

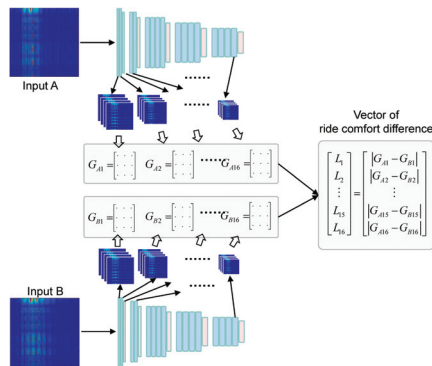


Figure 6. Comparative model of ride comfort. The output was a vector of the ride comfort difference. If the vector was a zero vector, the two vehicles had an identical ride comfort.

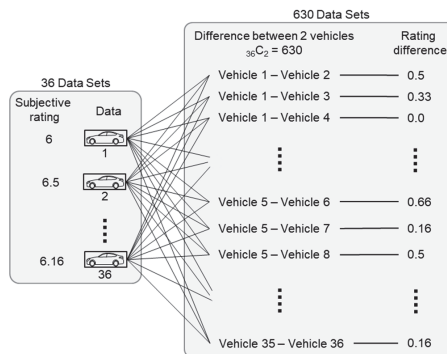


Figure 7. Expansion of datasets for model training.

2.2.4. Comparative Model of Ride Comfort

We designed a neural network for the correlation between the vector of ride comfort difference and the subjective rating difference. The network had eight fully connected hidden layers, as shown in Figure 8. The last hidden layer was designed for the visualization of the results with two nodes, x_1 and x_2 .

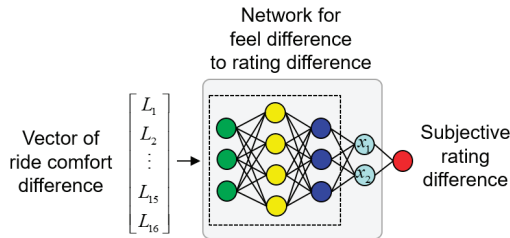


Figure 8. Correlation networks from the vector of ride comfort difference to the subjective rating difference. The neural network was composed of eight fully connected layers.

3. Results and Discussion

For both the primary ride comfort and impact comfort, 500 out of 630 datasets were used for training, and 130 sets were used for testing. Seven measured signals were used as the input data. The signals were the front-wheel damping force, rear-wheel damping force, vertical acceleration of the center of gravity, vertical acceleration of the left seat rail, vertical acceleration of the right seat rail, pitch, and pitch rate.

For the primary ride comfort, the root mean square error (RMSE) for training was 0.0049, and that of the test was 0.0465. For impact comfort, the RMSE for training was 0.0040, and that of the test was 0.071. The RMSEs of previously reported correlation models for the objectification of subjective evaluation were in the range of 0.1 to 0.7 [13,17,20,27]. One possible reason for the higher accuracy of the proposed method was the use of a CNN that enabled rich feature extraction without the predefinition of features. Another reason was the use of a neural network for the correlation between the vector of ride comfort difference and the subjective rating difference.

In Figure 9, the trained models for primary ride comfort and impact comfort were plotted as functions of x_1 and x_2 , which were the node values of the last hidden layer. x_1 predominantly affected the primary ride comfort, whereas x_2 predominantly affected the ride comfort during impact. These models predicted how much the ride comfort differed between two given vehicles but did not give any information about which vehicle had the better ride comfort. One possible way to overcome this limitation was by comparing the ride comfort of a vehicle to that of another vehicle that had the lowest subjective rating.

In our data, the vehicle with the lowest rating was the one with the H/S damper setting. Figure 10a shows the result of the comparison to this vehicle. A result farther from the origin indicated a better ride comfort. A similar model could also be extracted using the vehicle with the highest subjective rating, as shown in Figure 10b. The vehicle with the highest ride comfort was the one with the M/H damper setting. In this case, a closer result to the origin meant a better ride comfort.

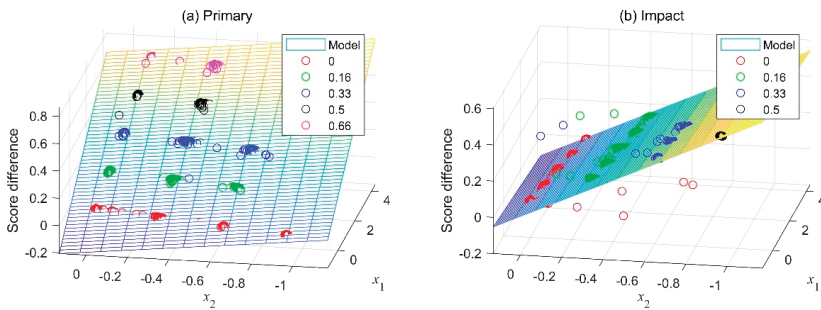


Figure 9. Three-dimensional view of the comparative models. The data points on the model were from test dataset. (a) Primary ride comfort (b) Impact ride comfort.

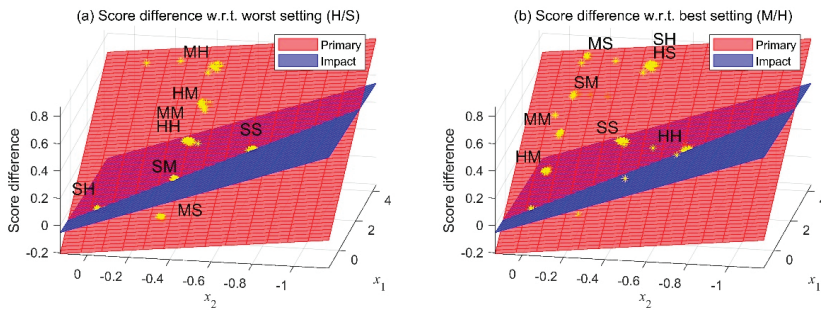


Figure 10. Comparative model (a) with respect to a vehicle with the lowest rating of ride comfort and (b) with respect to a vehicle with the highest rating of ride comfort.

4. Case Study for Use of the Correlation Model

Sensitivity Evaluation of Signal Changes to Ride Comfort

One useful application for this method is evaluating the sensitivity of the ride comfort rating to changes in the measured signals. The proposed model represented mapping functions from measured signals or vehicle dynamic states to subjective ratings. This model could tell what kinds of dynamic responses are beneficial for good ride comfort.

As examples of sensitivity analyses, we synthesized signals of the pitch angle and vertical acceleration, which were the most representative signals for primary ride comfort. Pitch angle variations that remained for a long time after passing over a speed bump negatively affected ride comfort. We synthesized the pitch angle signal such that it would decay faster after a speed bump. This signal change seemed to help improve primary ride comfort, as shown in Figure 11. The comparative model based on the worst-rated vehicle predicted a low correlation between this change and the ride comfort, but the comparative model based on the best-rated vehicle predicted a positive relationship.

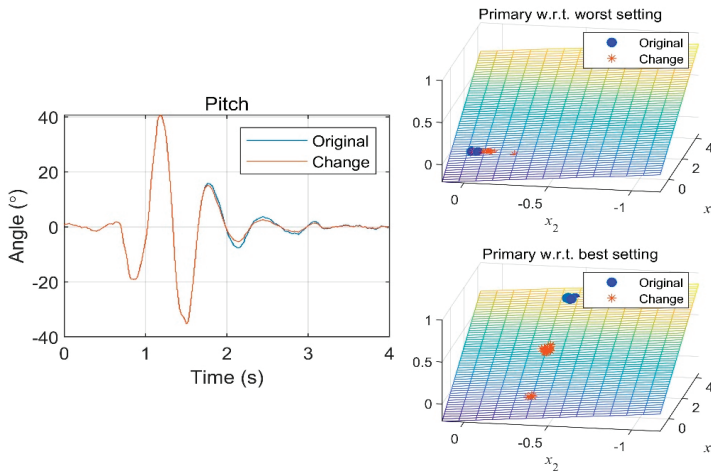


Figure 11. Ride comfort rating with respect to pitch decay rate change.

We performed a similar analysis with the vertical acceleration at the seat rail position. The vertical acceleration was synthesized to have a small magnitude reduction after passing over the bump, as shown in Figure 12. Both comparative models predicted no significant change in the primary ride comfort.

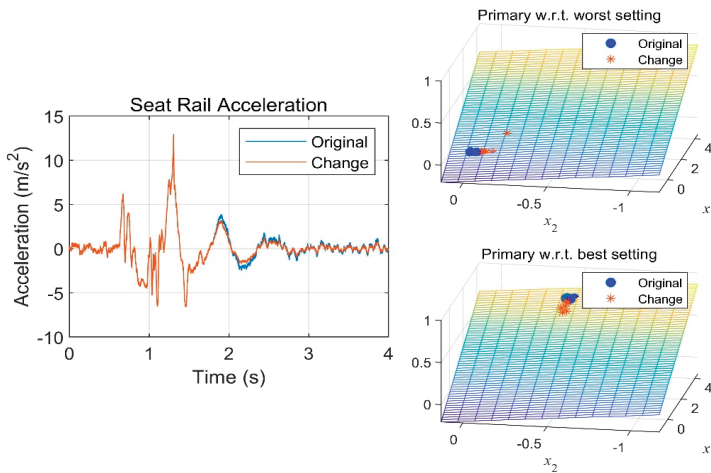


Figure 12. Ride comfort rating with respect to vertical acceleration change.

In the last case study, an analysis was performed on the reduction of the phase difference between the pitch rate and vertical acceleration at the center of gravity. When the phase difference was reduced, both comparative models predicted a significant improvement in the ride comfort, as shown in Figure 13. Other examples are shown in Table 3, which demonstrate that the proposed model could be used to identify the designed signal patterns without doing additional expensive experiments. Once the desired signal patterns were identified, tuning could be performed to improve the ride comfort in the early design stages if a simulation of a vehicle model with a high fidelity was available, as shown in Figure 14.

Table 3. Subjective evaluation of the ride comfort in a speed bump test.

| Cases | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------------------------------------|---|---|---|---|---|---|---|
| Pitch rate reduction | O | X | X | O | O | X | O |
| Acceleration reduction | X | O | X | O | X | O | O |
| Phase lag reduction | X | X | O | X | O | O | O |
| Improvement of primary ride comfort | Y | N | Y | Y | Y | Y | Y |

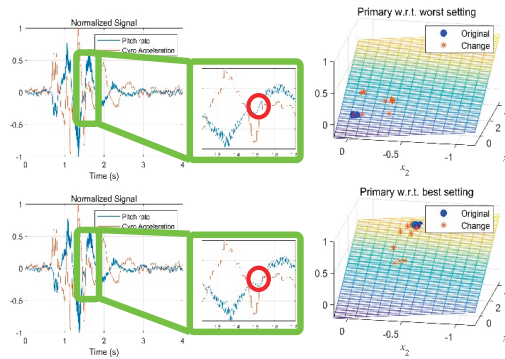


Figure 13. Ride comfort rating with respect to the change of phase difference between the pitch rate and vertical acceleration.

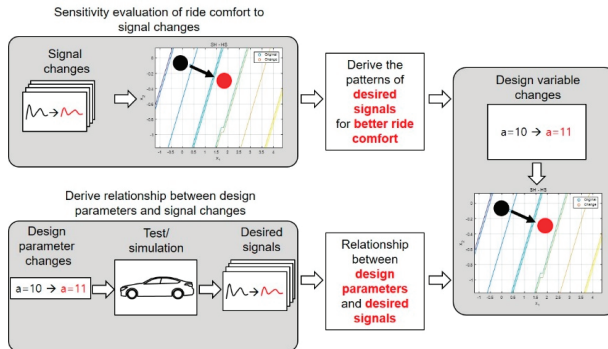


Figure 14. Possible use of evaluation model in vehicle design.

5. Conclusions

We have proposed a methodology to objectify subjective assessments using a pre-trained DNN. The proposed method does not require any feature definition or objective metric definition to extract ride comfort from measured signals. A DNN technique called artistic style transfer was used to extract a numerical form of the driving comfort without any predefined features, and then a comparative model was designed. The model showed a higher accuracy than any other correlation models in the literature. This was because of use of CNN in extracting performance metrics and use of a comparative model. The limitations of this research include a small number of test drivers, few test surfaces, and the limited number of datasets. These limitations can be a huge barrier to the use of a DNN, and thus, a general evaluator model could not be achieved with the given dataset. The limitations are typical and unavoidable for the subjective evaluation of vehicle ride comfort due to the expensive and long evaluation process. Therefore, the designed evaluator model does not work for all kinds of general conditions. Rather, the designed model can be used as a numerical evaluator model for

the given road surface with the given vehicle speed. During the research, the authors concluded that designing a general evaluator model for vehicle subjective evaluation working for any conditions is practically impossible and designing several case-dependent models for each different test condition would be practically viable. The authors showed that the proposed method was effective for such cases. Even though the proposed method itself can be applied to both general and case-dependent models, the strength of the proposed method lies on the design of a subjective evaluation model when there is a small number of data points.

Author Contributions: Writing—original draft preparation, D.K.; data collection and formal analysis, M.J.; funding acquisition and project administration, B.B.; supervision, writing—review and editing, C.A.

Funding: This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2019R1A2C1003103).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Li, C.; Chen, T. Aesthetic visual quality assessment of paintings. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 236–252. [[CrossRef](#)]
- Hayn-Leichsenring, G.U.; Lehmann, T.; Redies, C. Subjective ratings of beauty and aesthetics: Correlations with statistical image properties in western oil paintings. *1-Perception* **2017**, *8*. [[CrossRef](#)] [[PubMed](#)]
- Jeon, J.Y.; You, J.; Chang, H.Y. Sound radiation and sound quality characteristics of refrigerator noise in real living environments. *Appl. Acoust.* **2007**, *68*, 1118–1134. [[CrossRef](#)]
- Lee, C.; Cho, Y.; Baek, B.; Lee, S.; Hwang, D.; Jo, K. Analyses of refrigerator noises. In Proceedings of the IEEE International Symposium on Industrial Electronics, Dubrovnik, Croatia, 20–23 June 2005; pp. 1179–1184.
- Xin, Z.; Zhang, X.; Shi, G.; Lin, Y. Steering feel study on the performance of EPS. In Proceedings of the IEEE Vehicle Power and Propulsion Conference, Harbin, China, 3–5 September 2008; pp. 1–5.
- Huang, F. G1 model of expressway ride comfort assessment. In Proceedings of the International Conference on Measuring Technology and Mechatronics Automation, Changsha, China, 13–14 March 2010; pp. 341–342.
- Zheng, Y.; Meng, F.; Zhang, Y. A new subjective assessment system for video quality. In Proceedings of the International Congress on Image and Signal Processing, Dalian, China, 14–16 October 2014; pp. 586–590.
- Park, H.-J.; Har, D.-H. Subjective image quality assessment based on objective image quality measurement factors. *IEEE Trans. Consum. Electron.* **2011**, *57*, 1176–1184. [[CrossRef](#)]
- Wang, E.M.-Y.; Shih, S.S.-Y. A study on thumb and index finger operated interface for personal mobile devices: Mobile phone keypad and joystick. In Proceedings of the International Conference on Industrial Engineering and Engineering Management, Beijing, China, 21–23 October 2009; pp. 766–770.
- Arima, M.; Tamura, Y.; Yoshihira, M. Evaluation of ride comfort of passenger craft. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, 8–11 October 2006; pp. 802–807.
- Sawabe, T.; Okajima, T.; Kanbara, M.; Hagita, N. Evaluating passenger characteristics for ride comfort in autonomous wheelchairs. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems, Yokohama, Japan, 16–19 October 2017; pp. 102–107.
- Zschocke, A.K.; Albers, A. Links between subjective and objective evaluations regarding the steering character of automobiles. *Int. J. Automot. Technol.* **2008**, *9*, 473–481. [[CrossRef](#)]
- Chabrier, E.; Grima, M. Subjective and objective vehicle tests, two parallel vehicle handling evaluations. In Proceedings of the FISITA 2012 World Automotive Congress, Beijing, China, 27–30 November 2012; pp. 1767–1775.
- Furusho, Y.; Kotani, K. Objective and subjective evaluation models of pencil still drawings for art education. In Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, Sydney, Australia, 29 November–1 December 2017; pp. 1–5.
- Data, S.; Frigerio, F. Objective evaluation of handling quality. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2002**, *216*, 297–305. [[CrossRef](#)]
- Rothhämel, M.; Ijkema, J.; Drugge, L. A method to find correlations between steering feel and vehicle handling properties using a moving base driving simulator. *Veh. Syst. Dyn.* **2011**, *49*, 1837–1854. [[CrossRef](#)]

17. Nybacka, M.; He, X.; Gil Gómez, G.L.; Bakker, E.; Drugge, L. Links between subjective assessments and objective metrics for steering. *Int. J. Automot. Technol.* **2014**, *15*, 893–907. [[CrossRef](#)]
18. Nybacka, M.; He, X.; Su, Z.; Drugge, L.; Bakker, E. Links between subjective assessments and objective metrics for steering, and evaluation of driver ratings. *Veh. Syst. Dyn.* **2014**, *52*, 31–50. [[CrossRef](#)]
19. Gil Gómez, G.L.; Nybacka, M.; Bakker, E.; Drugge, L. Findings from subjective evaluations and driver ratings of vehicle dynamics: Steering and handling. *Veh. Syst. Dyn.* **2015**, *53*, 1416–1438. [[CrossRef](#)]
20. Gil Gómez, G.L.; Nybacka, M.; Bakker, E.; Drugge, L. Objective metrics for vehicle handling and steering and their correlations with subjective assessments. *Int. J. Automot. Technol.* **2016**, *17*, 777–794. [[CrossRef](#)]
21. Chen, G.; Zhang, W.; Gong, Z.; Sun, W. A new approach to vehicle shift quality subjective evaluation based on fuzzy logic and evidence theory. In Proceedings of the IEEE Conference on Industrial Electronics and Applications, Xian, China, 25–27 May 2009; pp. 2792–2795.
22. Liu, Y.; Liu, Q.; Lv, C.; Zheng, M.; Ji, X. A Study on objective evaluation of vehicle steering comfort based on driver's electromyogram and movement trajectory. *IEEE Trans. Hum.-Mach. Syst.* **2018**, *48*, 41–49. [[CrossRef](#)]
23. Varga, D. No-Reference Video Quality Assessment Based on the Temporal Pooling of Deep Features. *Neural Process. Lett.* **2019**, *50*, 1–14. [[CrossRef](#)]
24. Mahendran, N.; Vincent, D.R.; Srinivasan, K.; Chang, C.-Y.; Garg, A.; Gao, L.; Reina, D.G. Sensor-Assisted Weighted Average Ensemble Model for Detecting Major Depressive Disorder. *Sensors* **2019**, *19*, 4822. [[CrossRef](#)] [[PubMed](#)]
25. Weber, K.A.; Smith, A.C.; Wasielewski, M.; Egtesad, K.; Upadhyayula, P.A.; Wintermark, M.; Hastie, T.J.; Parrish, T.B.; Mackey, S.; Elliott, J.M. Deep Learning Convolutional Neural Networks for the Automatic Quantification of Muscle Fat Infiltration Following Whiplash Injury. *Sci. Rep.* **2019**, *9*, 7973. [[CrossRef](#)] [[PubMed](#)]
26. Yao, B.-G.; Peng, Y.-L.; Yang, Y.-J. Mechanical Measurement System and Precision Analysis for Tactile Property Evaluation of Porous Polymeric Materials. *Polymers* **2018**, *10*, 373. [[CrossRef](#)] [[PubMed](#)]
27. Gil Gómez, G.L.; Nybacka, M.; Drugge, L.; Bakker, E. Machine learning to classify and predict objective and subjective assessments of vehicle dynamics: The case of steering feel. *Veh. Syst. Dyn.* **2018**, *56*, 150–171. [[CrossRef](#)]
28. Mordvintsev, A.; Olah, C.; Tyka, M. DeepDream-A Code Example for Visualizing Neural Networks. Available online: <https://ai.googleblog.com/2015/07/deepdream-code-example-for-visualizing.html> (accessed on 10 September 2019).
29. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. SAE International Surface Vehicle Recommended Practice, Subjective Rating Scale for Vehicle Ride and Handling; SAE Standard J1441; SAE International: Warrendale, PA, USA, 2016.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Article

Developing a Neural–Kalman Filtering Approach for Estimating Traffic Stream Density Using Probe Vehicle Data

Mohammad A. Aljamal ¹, Hossam M. Abdelghaffar ^{2,3} and Hesham A. Rakha ^{1,*}

¹ Charles E. Via, Jr. Department of Civil and Environmental Engineering, Center for Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA

² Department of Computers Engineering and Systems, Engineering Faculty, Mansoura University, Mansoura, Dakahlia 35516, Egypt

³ Center for Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA

* Correspondence: hrakha@vt.edu

Received: 20 August 2019; Accepted: 2 October 2019; Published: 7 October 2019

Abstract: This paper presents a novel model for estimating the number of vehicles along signalized approaches. The proposed estimation algorithm utilizes the adaptive Kalman filter (AKF) to produce reliable traffic vehicle count estimates, considering real-time estimates of the system noise characteristics. The AKF utilizes only real-time probe vehicle data. The AKF is demonstrated to outperform the traditional Kalman filter, reducing the prediction error by up to 29%. In addition, the paper introduces a novel approach that combines the AKF with a neural network (AKFNN) to enhance the vehicle count estimates, where the neural network is employed to estimate the probe vehicles' market penetration rate. Results indicate that the accuracy of vehicle count estimates is significantly improved using the AKFNN approach (by up to 26%) over the AKF. Moreover, the paper investigates the sensitivity of the proposed AKF model to the initial conditions, such as the initial estimate of vehicle counts, initial mean estimate of the state system, and the initial covariance of the state estimate. The results demonstrate that the AKF is sensitive to the initial conditions. More accurate estimates could be achieved if the initial conditions are appropriately selected. In conclusion, the proposed AKF is more accurate than the traditional Kalman filter. Finally, the AKFNN approach is more accurate than the AKF and the traditional Kalman filter since the AKFNN uses more accurate values of the probe vehicle market penetration rate.

Keywords: real-time estimation; probe vehicle; traffic density; neural network; level of market penetration rate

1. Introduction

Real-time traffic state estimates have been increasingly recognized following the introduction of recent advanced technologies such as connected vehicle (CV) technologies. CVs aim to improve road safety by potentially reducing human errors, mitigating traffic congestion levels by offering alternative routes, and reducing on-road emissions and fuel consumption [1]. Nowadays, conducting research with limited probe vehicle data (e.g., CVs) is a challenge, especially when no additional data sources are provided. Hence, past research has utilized probe data in conjunction with existing detection systems to enhance proposed traffic models, despite the limitation that fixed detection techniques (e.g., loop detectors) always have some noise in their data [2–4].

A probe vehicle is defined as a vehicle that provides real-time information, such as its instantaneous position and speed. Several benefits of using probe vehicle data have been recognized; for example, the high quality of data compared with existing data sources (e.g., cameras and loop detectors), and data can be collected at any location inside the network, thus offering a clear picture about traffic behavior at any time. Therefore, transportation agencies are putting effort into facilitating the use of probe vehicle data.

Limited studies have used only information from probe vehicle data (e.g., Global Positioning Systems [GPSs]) to estimate the state of on-road traditional vehicles [5], such as traffic travel time, traffic density, traffic speed, and traffic volume. The real-time estimation of traffic density is important to achieving better traffic operations management in urban areas. This paper aims to estimate the total number of vehicles on signalized link approaches using only probe vehicle data. The estimate outcomes can be provided to traffic signal controllers to optimally determine the allocation of green time for each traffic signal phase [6,7], leading to better intersection performance measures such as intersection delays and vehicle crashes [8,9]. One concern with using probe vehicles is measuring their level of market penetration (LMP). The LMP is defined as the ratio of the total number of probe vehicles to the total number of vehicles. Providing accurate LMP estimates improves the estimation accuracy of the vehicle counts [5]. Therefore, in this paper, a machine-learning technique is developed to provide reliable LMP estimates.

2. Related Work

Different statistical tools have been used to estimate the total number of vehicles on arterial roads and freeways, such as the Kalman filter (KF) [10], Bayesian statistics [11], and Particle filter [12] approaches. The literature shows the benefits of using the KF technique in addressing different aspects of the traffic estimation problem. The KF has been used to estimate the traffic travel time [13,14], traffic speed [15,16], and traffic density [5,17]. Different detection techniques have been employed to estimate the number of vehicles, such as loop detectors, camera systems, and probe data. Two loop detectors, one at the entrance and the other at the exit of the link, are utilized to measure the total number of arrivals and departures, then the number of vehicles are simply obtained by applying the flow continuity equation [18]. A robust KF model with at least three loop detectors on the tested link was employed to estimate the number of vehicles on the link in [17]. The study derived the KF state equation from the flow continuity equation, while the measurement equation was derived from the relationship of the detector time occupancy and space occupancy; however, the cost of implementing such an algorithm in the field is high given the number of sensors needed. Another study employed the KF to estimate the number of vehicles on multi-section freeways. The state equation was derived from the flow continuity equation, while the measurement equation was derived from the hydrodynamic relationship between traffic speed and density [19]. Loop detectors were used in addition to speed sensors in the middle of the tested section. However, the proposed algorithm is hard to employ in the field due to the high cost of implementation. A video record, another detection technique, was used to estimate the traffic density for signalized links [20]. In that study, the authors used the space-mean speed rather than the traffic flow in the state equation due to high errors accompanied with sensor failures. Their argument takes into account that the space-mean speed is taken as an average quantity while the traffic flow is a cumulative quantity. They also demonstrated the importance of having knowledge about the system noise characteristics to improve the performance of the KF model. Consequently, the authors of this paper applied an adaptive Kalman filter (AKF) to enable real-time estimates of statistical parameters of the system noise rather than using predefined values for the entire simulation (as assumed in the traditional KF model).

As illustrated in the literature, stationary sensors, such as loop detectors and camera systems, suffer from poor detection accuracy and have high installation and maintenance costs. Advanced detection techniques such as GPS data have proven to be more accurate without the need to install additional

hardware. Consequently, recent studies have developed several traffic estimation models using fusion data (combination of two different data sources) to estimate the number of vehicles with the aim of achieving better accuracy than using only one source of data. In many of the works using fusion data, the KF technique was employed for estimating traffic density. One study achieved accurate estimated traffic density results using the traffic flow values measured from a video detection system and the travel time obtained from vehicles equipped with GPS devices [2]. The proposed estimation approach in this study differs in two significant ways from the proposed AKF model, namely only probe vehicle data are used with a variable time interval rather than a fixed value (the updating time interval was 1 min in [2]), and the proposed estimation approach uses the AKF to allow for real-time estimates of statistical parameters of the state and measurement noise.

Reviewing the literature, the KF model has proven its ability to address estimation research problems for different traffic applications. However, it is hard to implement in real-world applications due to hard estimates of statistical characteristics of the system noise (mean and variance). Consequently, researchers have developed the AKF to solve this issue and make field implementation possible. Chu et al. proposed an AKF model to estimate freeway travel time using both loop detectors and probe data [21]. They presented the estimation method for noise statistic parameters that was proposed in [22]. This estimation method of statistical parameters is known for its simplicity in handling errors and its fast processing time. Hence, in this study, the estimation of the statistical parameters uses the same estimation procedure as in Chu et al.'s study. It should be noted that the main difference between the proposed estimation approach and Chu et al.'s approach is that our model uses only probe vehicle data.

In a recent study, the KF model was proposed to estimate the number of vehicles on signalized link approaches using only probe vehicle data [5]. The KF state equation was based on the traffic flow continuity equation and thus one value of probe vehicle LMP (ρ), for the entire link, is used to scale up the probe measurements to reflect the total flow in the second term of the flow continuity equation as presented in Equation (1). It was found that using two LMP values (at the entrance and the exit of the link) produce more accurate vehicle count estimates, especially when dealing with low LMPs, as described later in Section 4.3. In Equation (1), $N(t)$ is the number of vehicles traversing the link at time (t), Δt is the variable duration of the updating time interval, $N(t - \Delta t)$ is the number of vehicles traversing the link in the previous interval, q^{in} and q^{out} are the probe flows entering and exiting the link between $(t - \Delta t)$ and (t) , respectively, and ρ is the LMP of probe vehicles.

$$N(t) = N(t - \Delta t) + \frac{\Delta t}{\rho} [q^{in}(t) - q^{out}(t)] \quad (1)$$

Machine learning has proven its ability to provide accurate estimates for different traffic characteristics [23–28]. Traffic speed and density have been estimated using an artificial neural network (ANN) model [23]. Video and Bluetooth data were used to build the ANN model. The traffic flow data were manually extracted from the video records, while the speed data were constructed from the collected Bluetooth travel time data. The neural network model (NN) is able to address the research problem if a good quantity of training data is accessible. Another study conducted several machine learning techniques such as k-means clustering, k-nearest neighbor classification, and locally weighted regression to estimate traffic speed [24] using archived data of speeds, counts, and densities. They found that machine learning models can improve the accuracy of speed estimation. Khan et al. [25] used artificial intelligence to classify the level of service in a freeway segment based on traffic density values. They used loop detectors and CV data to develop support vector machine and k-nearest neighbor classification. Results indicated higher accuracy from the support vector machine algorithm than the k-nearest neighbor classification algorithm. Estimating hourly traffic volumes between sensors was addressed using an NN model in the Maryland highway network [27], deploying both probe vehicles and automatic traffic recording station data to construct the

NN model. A comparison was also made between linear regression, k-nearest neighbor, support vector machine with linear kernel, random forest, and NN models, concluding that the NN model performed the best. The proposed approach produced 24% more accurate estimates than current volume profiles.

In this research study, an AKF technique was applied to estimate real-time vehicle counts along signalized link approaches using only probe vehicle data. The study then considers the recommendation of Aljamal et al.'s study [5] by using two LMP values at the entrance and the exit of the tested link. To achieve this task, an NN model was developed to provide real-time estimates of the LMP values to improve the accuracy of the proposed AKF model. After that, the paper develops the new AKFNN approach after combining the AKF with the developed NN models. The proposed study extends the state-of-the-art in vehicle count estimates by making four major contributions:

- The study tests the proposed AKF model using only probe vehicle data. The approach was evaluated considering different probe vehicle LMPs ranging from 10% to 90% at increments of 10%.
- The study develops an NN model to estimate the LMP of probe vehicles at the exit of the link to reflect the total vehicle departures.
- The study tests the developed AKFNN approach by using a fusion of probe and single-loop detector data. A comparison between the traditional KF, AKF, and AKFNN models is presented.
- The study examines the impact of the initial conditions on the AKF estimation model. Three initial condition parameters are tested: the initial vehicle count estimate, the initial mean estimate of the state noise errors, and the a priori initial covariance of the state system.

This paper is organized as follows. The first section describes the development of the simulation data. The second section describes the estimation models and the problem formulation for the KF, AKF, and AKFNN models. The third section discusses the results of the new proposed models. The fourth section provides the conclusions of the study and recommended future work.

3. Development of Simulation Data

This paper relies on the INTEGRATION traffic simulation model [29] to validate and test the accuracy of the proposed models. The INTEGRATION software has been extensively validated and demonstrated to replicate empirical observations [30–35]. Specifically, INTEGRATION was used to create synthetic data for conditions not observed in the field to quantify the sensitivity of the proposed method to the link length and traffic demand level. The selected tested link is located in downtown Blacksburg, Virginia, with an approximate length of 102 m based on ArcGis software, and connects two signalized intersections. The link characteristics were calibrated to local conditions using typical values, which included a free-flow speed of 40 (km/h), a speed-at-capacity of 32 (km/h), a jam density of 160 (veh/km/lane), and a base saturation flow rate of 2100 (veh/h/lane), which resulted in a roadway capacity of 700 (veh/h) given the cycle length and green times of the traffic signal. The traffic signal cycle length is 75 s and it has four phases with the following displayed green times: 5, 25, 5, and 28 s. The tested link here is assigned with a displayed green time of 25 s. These values were consistent with what was coded in the field.

The INTEGRATION simulation model was used to ease the generation of probe vehicle data as real probe data are not easy to access. For each LMP, a total of 50 scenarios were generated with different random seeds as conducted in [25]. Forty-nine scenarios were used to train and validate the proposed NN model, and scenario number 50 was considered the testing data set. The INTEGRATION model generates a “time-space” file which provides some information about the probe vehicles during their trips for every second. The time-space file records the instantaneous position, speed, and spacing for each probe vehicle. In addition to that, a loop detector is installed at the entrance of the tested link to create a detector output

file which provides some data about the simulation behavior such as speed, traffic volume, and occupancy at the detection location.

4. Estimation Models

This section first summarizes some crucial points regarding estimating the vehicle count as discussed in the authors' last research study [5]. In addition, this section describes the proposed AKF estimation model for estimating the vehicle count along signalized link approaches, and demonstrates the difference of the state-of-the-art KF model in [5] and the new proposed AKF model. Finally, an NN model is developed to provide estimates of the probe vehicle LMPs to be used in the proposed AKF model equations to attain higher accuracy. Two vehicle count estimation models are described in this section: (1) the AKF model, which uses only probe vehicle data; and (2) the AKFNN model, which fuses probe and single-loop detector data. The single-loop detector data were mainly used to develop the NN model.

4.1. Summary of the Developed KF Model

In a previous study [5], the authors developed a KF model to produce reliable vehicle count estimates using only probe vehicle data. In that study, the authors introduced a novel variable estimation time interval as opposed to the traditional fixed time interval. The estimation time interval was defined as the time when exactly n probe vehicles traversed the tested link. It was proven that the variable time interval, compared to a fixed time interval (e.g., 20 s), led to improved estimation accuracy. An illustrative example to show the benefits of using the variable time interval. If the approach's LMP is 10%, the number of probe vehicles will obviously be low. If we treat the problem using a fixed estimation interval, then the probability of observing zero probe vehicles within an interval will be high for short estimation time intervals, making the estimation inefficient and inaccurate. Accordingly, low LMPs require long intervals (e.g., 300 s) to ensure that at least one probe vehicle is on the approach. In contrast, approaches with high LMPs can use short estimation intervals (e.g., 20 s). Consequently, treating the estimation time interval as a variable produces an efficient and convenient way of determining the duration of the estimation period. For more details, readers may refer to [5].

One concern about the KF model is the use of predefined fixed values of the statistical parameters, mean and variance, of the KF state and measurement errors. Applying the KF model in real-world problems is limited since the statistical parameters are assumed to be known [21]. The mean and variance entities are known as variable rather than fixed values. To produce a flexible model, this study employs the AKF model to provide real-time estimates of the statistical parameters of the KF state and measurement errors as described in the following section.

4.2. Adaptive Kalman Filter (AKF)

The traditional KF technique is utilized with predefined error values of the state and measurement noise; these error values remain constant for the entire simulation. However, these values are hard to obtain in the field and they are always changing with time. Hence, an AKF is developed to overcome this issue and to dynamically estimate the error values in the state and measurement estimates. The AKF is comprised of two equations: (a) state equation and (b) measurement equation. The state equation is derived from the traffic flow continuity equation as defined in Equation (2). The state equation computes the number of vehicles by continuously adding the difference in the number of vehicles entering and exiting the section to the previously computed cumulative number of vehicles traveling along the section. This integral results in an accumulation error which requires fixing, and thus the measurement equation is needed. In Equation (2), the ρ value can be observed from historical data.

$$N(t) = N(t - \Delta t) + \frac{\Delta t}{\rho} [q^{in}(t) - q^{out}(t)] \quad (2)$$

The state equation produces accurate results if the scaled traffic flows (q^{in}/ρ_{in} and q^{out}/ρ_{out}) are accurate [5], as shown in Section 4.3. The total counts can be extracted from traditional loop detectors or video detection systems. We should note here that the ρ value in Equation (2) plays a major role in delivering accurate outcomes. ρ is defined as the ratio of the number of probe vehicles (N_{probe}) to the total number of vehicles (N_{total}), as shown in Equation (3). For instance, if ρ is equal 0.1, and the number of probe vehicles is 5, then the expected total number of vehicles is 50.

$$\rho = N_{probe} / N_{total} \quad (3)$$

Equation (4) describes the hydrodynamic relationship between the macroscopic traffic stream parameters (flow, density, and space-mean speed),

$$q = k u_s \quad (4)$$

where q is the traffic flow (vehicles per unit time), k is the traffic stream density (vehicles per unit distance), and u_s is the space-mean speed (distance per unit time). The u_s can be represented as shown in Equation (5),

$$u_s = D / TT \quad (5)$$

where D is the link length and TT is the average vehicle travel time. Since probe vehicles can share their instantaneous locations every Δt , the travel time of each probe vehicle can be computed for any road section. Thus, the probe vehicle travel time is used in the measurement equation, using Equations (4) and (5). The measurement equation can be written as shown in Equation (8):

$$TT(t) = D \times \frac{k(t)}{\bar{q}(t)} \quad (6)$$

$$TT(t) = \frac{1}{\bar{q}} [k(t) \times D] = \frac{1}{\bar{q}(t)} N(t) \quad (7)$$

$$TT(t) = H(t) \times N(t) \quad (8)$$

where \bar{q} is the average traffic flow entering and exiting the link, and $H(t)$ is a transition vector that converts the vehicle counts to travel times, and is the inverse of the average flow (i.e., the first term of Equation (7)), as shown in Equation (9).

$$H(t) = \frac{1}{\bar{q}(t)} = \frac{2 \times \rho}{q^{in}(t) + q^{out}(t)} \quad (9)$$

The system state and measurement equations can be written as in Equations (10) and (11), considering the errors (noise). The term $u(t)$ is the given inputs for the system. The vector $H(t)$ is used to convert the vehicle counts to travel times. The vector $w(t - \Delta t)$ is the state noise and is assumed to be Gaussian noise with the mean of $m(t)$ and variance of $M(t)$. The measurement noise $v(t)$ is assumed to be Gaussian noise with the mean of $r(t)$ and variance of $R(t)$.

$$\text{State Equation : } N(t) = N(t - \Delta t) + u(t) + w(t - \Delta t) \quad (10)$$

$$u(t) = \frac{\Delta t}{\rho} [q^{in}(t) - q^{out}(t)]$$

$$\text{Measurement Equation : } TT(t) = H(t) \times N(t) + v(t) \tag{11}$$

$$H(t) = \frac{1}{\bar{q}(t)} = \frac{2 \times \rho}{q^{in}(t) + q^{out}(t)}$$

The proposed AKF estimation model can be solved using the following equations:

$$\hat{N}^-(t) = \hat{N}^+(t - \Delta t) + u(t) + m(t - \Delta t) \tag{12}$$

$$\hat{P}^-(t) = \hat{P}^+(t - \Delta t) + M(t - \Delta t) \tag{13}$$

$$G(t) = \hat{P}^-(t)H(t)^T [H(t)\hat{P}^-(t) H(t)^T + R(t)]^{-1} \tag{14}$$

$$\hat{N}^+(t) = \hat{N}^-(t) + G(t) [TT(t) - H(t)\hat{N}^-(t) - r(t)] \tag{15}$$

$$\hat{P}^+(t) = \hat{P}^-(t) \times [1 - H(t) G(t)] \tag{16}$$

where \hat{N}^- is the a priori estimate of the vehicle counts calculated using the measurement prior to instant t , and \hat{P}^- is the a priori estimate of the covariance error at instant t . The Kalman gain (G) is demonstrated in Equation (14). The posterior state estimate (\hat{N}^+) and the posterior error covariance estimate (\hat{P}^+) are updated as shown in Equations (15) and (16), considering the probe vehicle travel time measurements. In the next section, the estimation steps of the noise statistical parameters (m, M, r, R) are described.

4.2.1. Online Estimation of Noise Statistics

An online estimate is conducted to optimally find the errors in the state and the measurement variables, to make the KF more efficient and applicable in real-world applications. As pointed out in the literature, the traditional KF assumes predefined errors in the system, which is not the case in real applications. A set of unknown noise statistical parameters, (m, M, r, R), needs to be estimated at every estimation step. The online estimate procedure follows the same procedure presented in [21].

The mean (m) and variance (M) of the state noise are shown in Equations (17) and (18), respectively.

$$m = \frac{1}{n} \sum_{t=1}^n m(t), \quad \text{where } m(t) = \hat{N}^+(t) - \hat{N}^+(t - \Delta t) - u(t) \tag{17}$$

$$M = \frac{1}{n-1} \sum_{t=1}^n [(m(t) - m) \cdot (m(t) - m)^T - (\frac{n-1}{n})\hat{P}^+(t - \Delta t) - \hat{P}^+(t)] \tag{18}$$

where $m(t)$ is the state noise at time t , the first term of Equation (18) is the covariance of w at time t , n is the number of state noise samples.

The mean (r) and variance (R) of the measurement noise are shown in Equations (19) and (20), respectively.

$$r = \frac{1}{n} \sum_{t=1}^n r(t), \quad \text{where } r(t) = TT(t) - H(t) \hat{N}^-(t) \tag{19}$$

$$R = \frac{1}{n-1} \sum_{t=1}^n [(r(t) - r) \cdot (r(t) - r)^T - (\frac{n-1}{n})H(t)\hat{P}^-(t)H^T(t)] \tag{20}$$

where $R(t)$ is the observation noise at time t . The first term of Equation (20) is the covariance of v at time t , and n is the number of measurement noise samples. As a summary, the KF and AKF models use the same equations except for the fact that the AKF model estimates the statistical parameters of the noise for every estimation step using Equations (17) to (20).

As found in our previous study [5], providing the system equations real-time estimates of ρ_{in} and ρ_{out} should improve the estimation accuracy. In this study, a single-loop detector was installed at the entrance of the tested link to produce real-time estimates of ρ_{in} . In contrast, in the next section, an NN model is developed to obtain real-time estimates for the ρ_{out} values.

4.3. Neural Network

NN is a machine learning technique that aims to recognize relationships between vast amounts of data by employing a certain number of neurons in every single hidden layer to achieve better accuracy [36]. The network consists of three main layers: the input layer, the hidden layer, and the output layer. This section takes into account the recommendation of using two market penetration rates (at the entrance and exit of the link) rather than one market penetration rate along the tested link in the KF equations [5]. Accordingly, the state equation and the H vector in the measurement equation are revised as presented in Equations (21) and (22). ρ_{in} and ρ_{out} are the probe LMP at the entrance and the exit of the link, respectively.

$$N(t) = N(t - \Delta t) + \Delta t \left[\frac{q^{in}(t)}{\rho_{in}(t)} - \frac{q^{out}(t)}{\rho_{out}(t)} \right] \tag{21}$$

$$H(t) = \frac{1}{\bar{q}(t)} = \frac{2}{\frac{q^{in}(t)}{\rho_{in}(t)} + \frac{q^{out}(t)}{\rho_{out}(t)}} \tag{22}$$

A single-loop detector was installed at the entrance of the link to measure ρ_{in} and also use as an input to the NN model. Accordingly, this study develops an NN model to estimate ρ_{out} . The tested link is shown in Figure 1. The next section describes the selected inputs (features) and the output variables of the NN model.

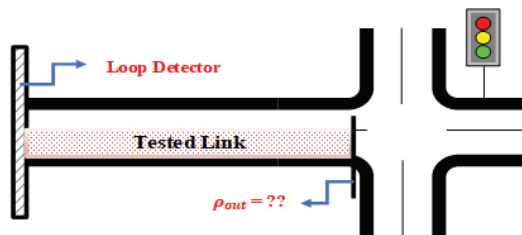


Figure 1. Tested link.

Characteristics of the NN: Input and Output Variables

Previous research has used different features to build machine learning models [23–26]. Fusing video and Bluetooth data was used to estimate traffic density and speed. The traffic flow was manually extracted from the video records, while the speed data were constructed from the collected Bluetooth travel time data [23]. Another study relied on archived data of traffic speeds, counts, and density to estimate traffic

speed [24]. Distance headway, number of stops, and speed data were identified as useful features to achieve accurate density estimates [25]. They employed loop detectors and CV data. In a recent study, Sekula et al. used probe and automatic traffic recording station data to extract the features of the NN model [27]. The selected features were the (1) speed of probe vehicles, (2) weather data such as temperature, visibility, precipitation, and weather status, (3) infrastructure data (speed limits, number of lanes, class of the road, and type of the road), (4) temporal data such as the day of the week, and (5) volume profiles based on historical data. The literature showed that the traffic speed is always used as a model feature, especially when probe vehicle data are used. In contrast, the traffic flow is always used when stationary sensors (e.g., loop detector) are used.

In this paper, a fusion of probe and single-loop detector data is utilized to produce the model features. The single-loop detector was installed at the entrance of the link and thus ρ_{in} can be computed directly using Equation (3). The ρ_{out} variable is calculated from the NN (the NN output). Seven possible inputs (features) were considered in the NN model, as defined in Table 1. Conducting a feature selection technique to validate the importance of each feature for the NN model, the number of the model features was dropped to five features. It should be noted that the selected model inputs can be easily extracted when probe vehicles are on the link. ρ_{out} can be expressed as a function of the selected inputs, as presented in Equation (23).

$$\rho_{out} = f(A_t, A_p, u_s, S_1, S_2) \quad (23)$$

The ρ_{out} values vary between 0 and 1, the 0 value means that no probe vehicles were observed at the exit of the link, while the value of 1 means that the D_p value is the same as the D_t . The selected inputs must be relevant to the model output ρ_{out} to allow the NN model to build a strong relationship between the model inputs and outputs, and therefore produce high estimation accuracy. For instance, in our case, the ρ_{out} value decreases as A_t and A_p increase. For instance, a high value of A_t means that the link is more congested and thus the number of departures (D_t) is expected to be high. The ρ_{out} value also decreases with increasing speed (S_1 , S_2 , and u_s). The speed is an indicator of the congestion level of the link; for instance, if the speed is low, then more vehicles are expected to be on the link, leading to higher values of D_t .

Table 1. Definition of the NN model inputs.

| Input Symbol | Definition | Unit |
|--------------|---|------|
| A_t | Total number of arrivals obtained from the single-loop detector | veh |
| A_p | Total number of probe arrivals | veh |
| D_p | Total number of probe departures | veh |
| S_1 | Average speed for probe vehicles at link entrance | km/h |
| S_2 | Average speed for probe vehicles at link exit | km/h |
| u_s | Space-mean speed for probe vehicles | km/h |
| u_t | Time-mean speed for probe vehicles | km/h |

A single hidden layer with one neuron, with a transfer function of hyperbolic tangent sigmoid, was used to build the NN model as shown in Figure 2. The Levenberg–Marquardt (LM) optimization has been proven in the literature to outperform the gradient decent and conjugate gradient methods for medium-sized problems [37]. Furthermore, the LM is considered the fastest back-propagation algorithm and thus was implemented in the proposed approach. The weights and biases of the developed NN model are described below. w_1 depicts the weights between the input layer and the hidden layer, while w_2 represents the weight between the hidden layer and the output layer. b_1 and b_2 represent the biases at the

hidden and output layers, respectively. Figure 2 describes the proposed AKFNN approach, combining the AKF model with the NN model.

$$w_1 = [0.43 \quad 0.19 \quad -47.28 \quad 0.36 \quad -0.43], \quad w_2 = [1.70], \quad b_1 = [-46.62], \quad b_2 = [0.95]$$

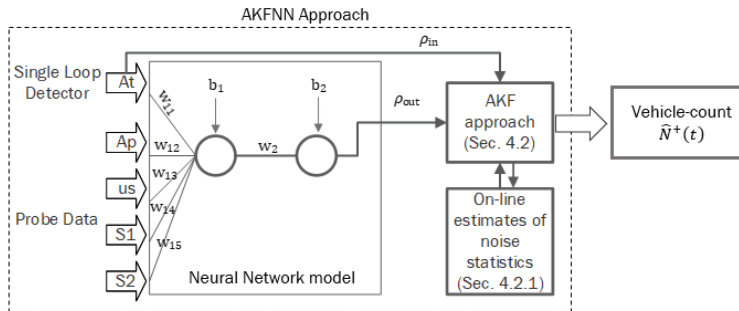


Figure 2. Flowchart for adaptive Kalman filter with a neural network (AKFNN) approach.

5. Results

This section evaluates the performance of the proposed models. The first subsection evaluates the performance of the AKF model and then compares the AKF with the KF model (Section 5.1). The second subsection presents the performance of the NN model used for estimating the LMP of probe vehicles at the exit of the link (ρ_{out}) (Section 5.2). The third subsection compares the performance of AKF with the AKFNN approach (Section 5.3). The fourth subsection investigates the sensitivity of the AKF estimation model to the initial conditions (Section 5.4). The accuracy of the proposed models was evaluated based on the root mean square error (RMSE) as shown in Equation (24). The RMSE has been frequently used in the literature to measure the difference between the model estimates and the actual values.

$$RMSE \text{ (veh)} = \sqrt{\sum_{t=1}^n [\hat{N}^+(t) - N(t)]^2 / n} \tag{24}$$

where $\hat{N}^+(t)$ represents the estimated vehicle count values, $N(t)$ represents the actual vehicle count values, and n is the total number of estimations. All simulation scenarios start with the following initial conditions: an initial vehicle count estimate of zero ($\hat{N}^+(0) = 0$ veh), which is the same value of the actual vehicle count, and initial mean and the prior covariance estimates of the state system ($m(0) = 2$ veh and $\hat{P}^-(0) = 75$ veh²) if the LMP scenario is less than or equal 60%, and ($m(0) = 9$ veh $\hat{P}^-(0) = 120$ veh²) if the LMP scenario is greater than 60%. The proposed models were evaluated using different probe vehicle LMPs, including 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%. For each scenario, a Monte Carlo simulation was conducted to create 300 random samples of probe vehicles from the full data set.

5.1. Comparison of the KF and the AKF Models

This section evaluates the proposed AKF model with real-time estimates of the error statistical parameters for the state and the measurement. This section also compares the proposed AKF model with the developed KF model in [5], as shown in Table 2. Results show that the AKF outperforms the KF model in most scenarios except for the scenarios with high LMPs (i.e., LMP of 80% and 90%). Results demonstrate the need to provide real-time estimates for the mean and variance error values in the state

and measurement when dealing with low/medium LMPs. This happened due to high error in the fixed ρ value that was used, which then produced high error in the vehicle count estimate. The AKF improved the traditional KF vehicle-count estimation accuracy by up to 29%. In contrast, for high LMPs, the user may proceed with predefined statistical values for the state and measurement (mean and variance error values), due to low errors in the vehicle count estimates (low error in the ρ value). In conclusion, a simple KF can be used with high LMPs without the need to change statistical noise parameters at every estimation step.

Table 2. Root mean square error (RMSE) values using the Kalman filter (KF) and the adaptive Kalman filter (AKF) models.

| LMP (%) | RMSE (veh) | | |
|---------|------------|-----|-----------------|
| | KF | AKF | Improvement (%) |
| 10 | 6.0 | 4.3 | 29 |
| 20 | 5.6 | 4.0 | 28 |
| 30 | 5.0 | 3.8 | 23 |
| 40 | 4.6 | 3.6 | 22 |
| 50 | 4.1 | 3.6 | 11 |
| 60 | 3.6 | 3.2 | 11 |
| 70 | 3.0 | 3.0 | 0 |
| 80 | 2.3 | 2.6 | −13 |
| 90 | 1.6 | 2.0 | −25 |

5.2. Developed NN Model

The NN model was employed to predict the (ρ_{out}) value, which is used to reflect the total number of vehicle departures from the given number of probe vehicle departures. The data set was divided into 70% for training, 15% for validation, and 15% for testing. The validation data set is used to measure network generalization and to avoid any over fitting problems [38]. The developed NN performance is shown in Table 3. The mean square error (MSE) is 0.01 and the R value is close to 1.0. The R value measures the correlation between model outputs and desired outputs. A value close to 1.0 means that the model outputs are very close to desired outputs. Figure 3 shows the error histogram for the training, validation, and testing data and their deviations from the zero error bar. Most of the errors lie around the zero error bar, which means that the developed NN model appropriately addressed the research goal (i.e., estimating ρ_{out}). Figure 4 presents the predicted and actual values for the ρ_{out} at different LMPs.

Table 3. Developed neural network (NN) model performance measures for the training, validation, and testing data set.

| Data Set | Samples | MSE | R |
|------------|---------|--------|-------|
| Training | 346,881 | 0.0171 | 0.872 |
| Validation | 74,331 | 0.0170 | 0.872 |
| Testing | 74,331 | 0.0173 | 0.871 |

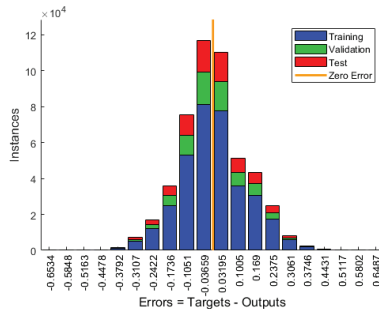


Figure 3. Error histogram for the training, validation, and testing data set.

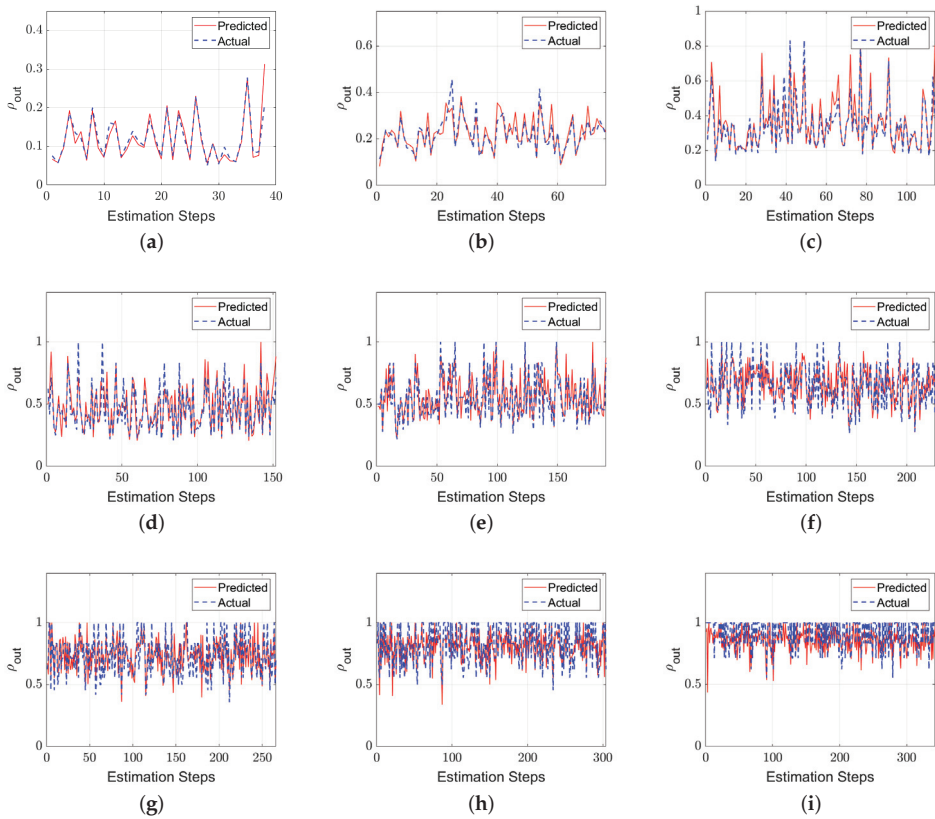


Figure 4. Actual and estimated values of ρ_{out} for different level of market penetration (LMP) scenarios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, (e) 50%, (f) 60%, (g) 70%, (h) 80%, and (i) 90% LMP.

5.3. Comparison of the AKF and the AKFNN Models

This section demonstrates the impact of using two ρ values rather than using one predefined ρ value. The average predefined ρ value is defined as the value for the entire tested link. The average ρ value remains constant for the entire simulation for each LMP scenario. For instance, if the scenario of 10% LMP is tested, the ρ value in both the state and measurement is treated as a value of 0.1. In this study, the authors proposed the use of two ρ values; one at the entrance and one at the exit of the link to reflect the total number of arrivals and departures from the given total number of probe arrivals and departures, respectively.

ρ_{in} is measured directly using the installed loop detector at the entrance of the link. The developed NN model is used to predict the ρ_{out} values (Section 5.2). Then, the ρ_{in} and ρ_{out} values are utilized in the AKF equations. Recall that the AKF model relies only on probe vehicle data, while the AKFNN model uses a fusion of probe vehicle and single-loop detector data.

In Table 4, the RMSE values using the AKF and the AKFNN models are presented. The results demonstrate the benefits of using the AKFNN approach rather than the AKF approach, where the estimation accuracy is improved by up to 26%. This finding proves what was recommended by Aljamal et al.'s previous study [5] to consider two ρ values rather than one value. As a result, the proposed AKFNN approach is robust and produces reasonable errors even with low LMPs. For instance, the estimated vehicle count values are off by 3.7 veh when the LMP is equal to 10%. Figure 5 presents the vehicle count estimation for different LMPs using the proposed AKFNN Approach.

Table 4. RMSE values using the AKF and the AKFNN models.

| LMP (%) | RMSE (veh) | | |
|---------|------------|-------|-----------------|
| | AKF | AKFNN | Improvement (%) |
| 10 | 4.3 | 3.7 | 13 |
| 20 | 4.0 | 3.6 | 11 |
| 30 | 3.8 | 3.5 | 9 |
| 40 | 3.6 | 3.3 | 8 |
| 50 | 3.6 | 2.7 | 26 |
| 60 | 3.2 | 2.4 | 25 |
| 70 | 3.0 | 2.4 | 20 |
| 80 | 2.6 | 2.3 | 12 |
| 90 | 2.0 | 1.8 | 10 |

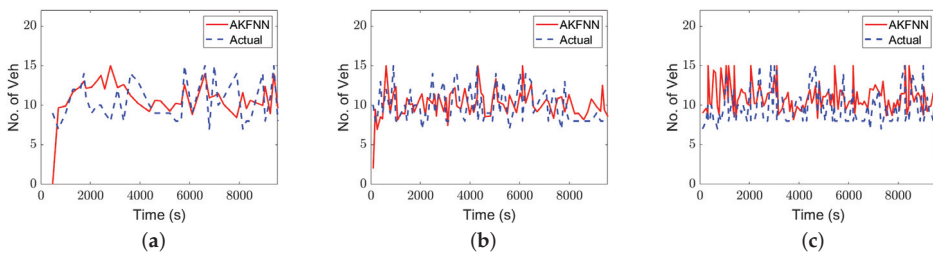


Figure 5. Cont.

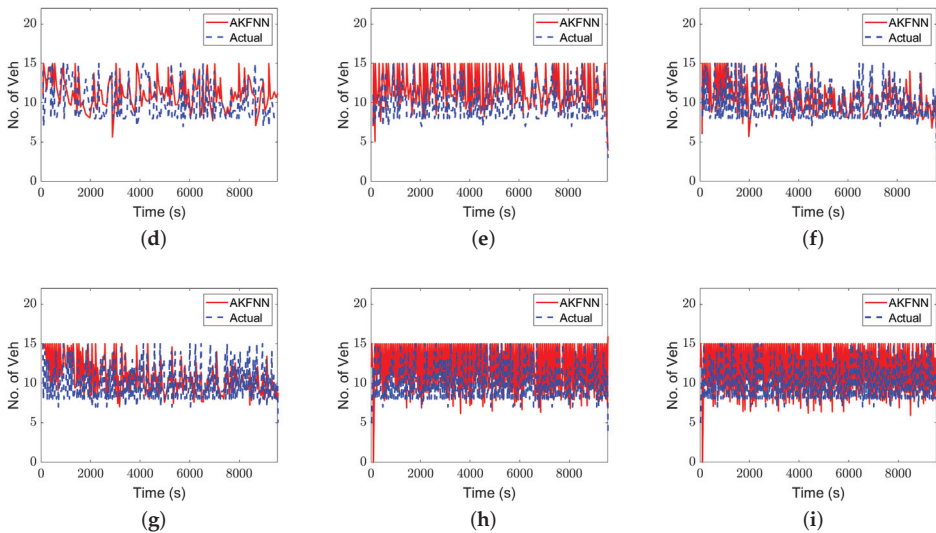


Figure 5. Actual and estimated vehicle counts over estimation intervals for different LMP scenarios: (a) 10%, (b) 20%, (c) 30%, (d) 40%, (e) 50%, (f) 60%, (g) 70%, (h) 80%, and (i) 90% LMP.

5.4. Impact of the Initial Conditions on the AKF Model

The KF model, traditional and adaptive, is sensitive to the initial condition parameters, such as the posterior state estimate ($N_i = \hat{N}^+(0)$), the mean of state noise ($m_i = m(0)$), and the prior error covariance estimate ($P_i = \hat{P}^-(0)$). These parameters are tuned by a trial-and-error technique to find the best initial condition values for seeking better KF estimation outcomes. However, in real applications, trial-and-error is not realistic and not easy to achieve. Hence, this section investigates the impact of initial conditions on the accuracy of the vehicle count estimation.

5.4.1. Impact of Initial Estimate of the Vehicle Count (N_i)

For the initial estimate value of the vehicle count (N_i), different values were evaluated (ranges from 0 to 10 at increments of 1). In this study, remember that all simulation scenarios start with an initial estimate of zero ($N_i = 0$ veh), which is the same value as the actual vehicle count. Figure 6a presents the RMSE values for different N_i values for the scenario of 10% LMP. As shown in the figure, the values of 8 and 10 produce the lowest RMSE. The RMSE value is equal to 4.3 veh when N_i is equal to 0. In contrast, the RMSE value is equal to 3.9 veh when N_i is equal to 8. As a result, starting the AKF model with the best initial estimate (e.g., $N_i = 8$ veh) would reduce the errors and therefore improve the estimation accuracy.

5.4.2. Impact of Initial Mean Estimate of the State System (m_i)

Another critical initial parameter in the AKF model is m_i . This parameter represents the mean value of the noise in the state equation. This paper tests 16 different m_i values (i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15). Figure 6b presents the vehicle count estimation RMSE values for different m_i values. The RMSE value is equal to 4.7 veh when the simulation starts with a 0 value of m_i . In contrast, the RMSE value is 3.9 veh when the value of m_i is equal to 11.

5.4.3. Impact of Initial Prior Covariance Estimate of the State System (P_i)

The last parameter tested in this study is the initial prior estimate of error covariance P_i . The error covariance parameter describes the accuracy of the state system. For instance, if the covariance value is low, then the state outcome is accurate and close to the actual value. As stated in the literature, the initial parameters should always be tuned to achieve accurate estimation accuracy. Thirteen different P_i values were tested (i.e., 5, 10, 15, 20, 25, 50, 75, 100, 120, 150, 200, and 250). Figure 6c presents the RMSE values using different P_i values. The P_i value of 150 veh² produces the lowest RMSE values.

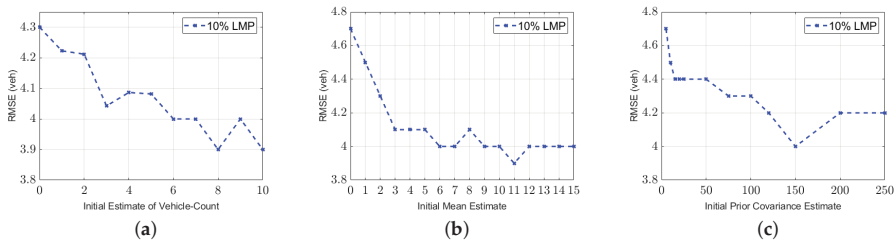


Figure 6. Impact of the initial conditions on the AKF model: (a) Initial estimate values N_i , (b) Initial mean estimate values m_i , and (c) Initial covariance estimate values P_i .

The research presented in this study evaluates the proposed approaches as they should be in real-world applications. Therefore, the trial-and-error technique was avoided since it is not a valid solution in the field. However, it was noticed that previous research always tunes the initial parameters to determine the best initial conditions when testing their estimation approaches [2,3,17]. If that is the case, let us assume that the proposed AKFNN approach always starts with the best initial value of P_i , which would produce less errors. Table 5 presents the RMSE when considering the trial-and-error technique (Tuned AKFNN). The AKFNN and the Tuned AKFNN approaches used the same values of N_i and m_i , but they used different P_i values. N_i is assumed to be zero, while m_i has two values based on the tested scenario: a value of 2 veh when low LMP scenarios are tested (LMP \leq 60%), and a value of 9 veh with high LMP scenarios (LMP $>$ 60%). From the table, tuning the P_i value significantly improves the estimation accuracy for all scenarios (by up to 27%). For instance, at 10% LMP, the estimation error dropped from 3.7 to 3.3 vehicles. On the other hand, the estimated vehicle count values are off by 2.8 vehicles instead of 3.6 vehicles for the scenario of 20% LMP.

Table 5. Impact of applying the trial-and-error technique for the initial value of covariance P_i .

| LMP (%) | RMSE (veh) | | |
|---------|------------|-------------|-----------------|
| | AKFNN | Tuned AKFNN | Improvement (%) |
| 10 | 3.7 | 3.3 | 11 |
| 20 | 3.6 | 2.8 | 22 |
| 30 | 3.5 | 2.7 | 23 |
| 40 | 3.3 | 2.4 | 27 |
| 50 | 2.7 | 2.1 | 22 |
| 60 | 2.4 | 2.1 | 13 |
| 70 | 2.4 | 2.1 | 13 |
| 80 | 2.3 | 1.8 | 22 |
| 90 | 1.8 | 1.5 | 17 |

In conclusion, the AKF model was proven to be very sensitive to the initial conditions (N_i, m_i, P_i). Hence, starting the simulation with good assumptions of the initial conditions can significantly improve the estimation accuracy, as shown in Table 5. Finally, Table 6 presents the performance of the models discussed in the paper.

Table 6. RMSE values for the KF, the AKF, the AKFNN, and the tuned AKFNN models.

| LMP (%) | RMSE (veh) | | | |
|---------|------------|-----|-------|-------------|
| | KF | AKF | AKFNN | Tuned AKFNN |
| 10 | 6.0 | 4.3 | 3.7 | 3.3 |
| 20 | 5.6 | 4.0 | 3.6 | 2.8 |
| 30 | 5.0 | 3.8 | 3.5 | 2.7 |
| 40 | 4.6 | 3.6 | 3.3 | 2.4 |
| 50 | 4.1 | 3.6 | 2.7 | 2.1 |
| 60 | 3.6 | 3.2 | 2.4 | 2.1 |
| 70 | 3.0 | 3.0 | 2.4 | 2.1 |
| 80 | 2.3 | 2.6 | 2.3 | 1.8 |
| 90 | 1.6 | 2.0 | 1.8 | 1.5 |

6. Conclusions

The research proposed a novel AKF model for estimating the number of vehicles on signalized approaches using only probe vehicle data. An AKF model was developed to provide real-time estimates of the statistical properties (mean and variance) for the state and measurement errors. The state equation is derived from the traffic flow continuity equation, while the measurement equation is constructed using the traffic hydrodynamic equation. Results show that the proposed AKF model outperforms the traditional KF model (improves the estimation accuracy by up to 29%), demonstrating the need to use real-time values of the statistical noise parameters in the KF model.

Two estimation models were presented, namely (a) the AKF and (b) the AKFNN. The AKF model uses only probe vehicle data assuming a fixed LMP value that is obtained from historical data, while the AKFNN uses a fusion of probe and single-loop detector data with real-time estimates of the LMP values (ρ_{in} and ρ_{out}). In this paper, a robust NN model was developed to provide accurate real-time estimates of the ρ_{out} values. The selected features of the NN model are A_t (observed from the single-loop detector), A_p , u_s , S_1 , and S_2 (observed from probe vehicles).

The AKF and the NN models were combined to develop the novel AKFNN approach. Results demonstrate that the AKFNN approach significantly improves the vehicle count estimation accuracy since the ρ_{in} and ρ_{out} values are estimated better. Subsequently, the paper compared the AKF with the AKFNN models, showing that the AKFNN model outperforms the AKF model, enhancing the estimation accuracy by up to 26%.

Finally, the study investigated the impact of the initial conditions (N_i , m_i , and P_i) on the AKF performance. Results show that the AKF model is very sensitive to the initial conditions. For instance, starting the simulation with an N_i value of 8 instead of 0 improves the estimation accuracy by 10%. In addition, starting the simulation with an m_i value of 11 instead of 2 enhances the estimation accuracy by up to 10%. For the P_i parameter, an improvement of 7% could occur if the simulation starts with an initial value of 150 instead of 75 veh². The study also tested the accuracy of the AKFNN estimation by allowing the P_i parameter to be tuned (Tuned AKFNN approach), showing that more improvement could be achieved. Specifically, the Tuned AKFNN improves the accuracy by up to 27%.

In conclusion, both models (AKF and AKFNN) produce high estimation accuracy when compared with the state-of-the-art KF model. Proposed future work entails testing traffic signal performance using the estimates of the total number of vehicles as inputs to the traffic signal controller.

Author Contributions: The work described in this article is the collaborative development of all authors, conceptualization, M.A.A., H.M.A., and H.A.R.; methodology, M.A.A., H.M.A., and H.A.R.; software, M.A.A., H.M.A., and H.A.R.; validation, M.A.A., H.M.A., and H.A.R.; formal analysis, M.A.A., H.M.A., and H.A.R.; investigation, M.A.A., H.M.A., and H.A.R.; writing—review and editing, M.A.A., H.M.A., and H.A.R.

Funding: This research effort was sponsored by the University Mobility and Equity Center (UMEC).

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this article.

References

1. Zmud, J.; Goodin, G.; Moran, M.; Kalra, N.; Thorn, E. *Advancing Automated and Connected Vehicles: Policy and Planning Strategies for State and Local Transportation Agencies*; National Academies Press: Washington, DC, USA, 2017.
2. Anand, R.A.; Vanajakshi, L.; Subramanian, S.C. Traffic density estimation under heterogeneous traffic conditions using data fusion. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 31–36.
3. Anand, A.; Ramadurai, G.; Vanajakshi, L. Data fusion-based traffic density estimation and prediction. *J. Intell. Transp. Syst.* **2014**, *18*, 367–378. [[CrossRef](#)]
4. Badillo, B.E.; Rakha, H.; Rioux, T.W.; Abrams, M. Queue length estimation using conventional vehicle detector and probe vehicle data. In Proceedings of the 2012 IEEE Conference on Intelligent Transportation Systems (15th IEEE ITSC), Anchorage, AK, USA, 16–19 September 2012; pp. 1674–1681.
5. Aljamal, M.A.; Abdelghaffar, H.M.; Rakha, H.A. Kalman filter-based vehicle count estimation approach using probe data: A multi-lane road case study. In Proceedings of the 22st International Conference on Intelligent Transportation Systems (ITSC), Auckland, New Zealand, 27–30 October 2019.
6. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. Isolated traffic signal control using Nash bargaining optimization. *Global J. Res. Eng.* **2017**, *16*, pp. 27–36.
7. Abdelghaffar, H.M.; Rakha, H.A. A novel decentralized game-theoretic adaptive traffic signal controller: Large-scale testing. *Sensors* **2019**, *19*, 2282. [[CrossRef](#)] [[PubMed](#)]
8. Rakha, H.; Van Aerde, M. REALTRAN: An off-line emulator for estimating the effects of SCOOT. *Transp. Res. Rec.* **1995**, *1494*, 124–128.
9. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. Isolated traffic signal control using a game theoretic framework. In Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1496–1501.
10. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
11. Press, S.J.; Press, J.S. *Bayesian Statistics: Principles, Models, and Applications*; Wiley: New York, NY, USA, 1989.
12. Del Moral, P. Non-linear filtering: Interacting particle resolution. *Markov Processes Relat. Fields* **1996**, *2*, 555–581.
13. Yang, J.S. Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 2128–2133.
14. Chen, M.; Chien, S.I. Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based. *Transp. Res. Rec.* **2001**, *1768*, 157–161. [[CrossRef](#)]
15. Guo, J.; Xia, J.; Smith, B.L. Kalman filter approach to speed estimation using single loop detector measurements under congested conditions. *J. Transp. Eng.* **2009**, *135*, 927–934. [[CrossRef](#)]
16. Ye, Z.; Zhang, Y.; Middleton, D.R. Unscented Kalman filter method for speed estimation using single loop detector data. *Transp. Res. Rec.* **2006**, *1968*, 117–125. [[CrossRef](#)]
17. Vigos, G.; Papageorgiou, M.; Wang, Y. Real-time estimation of vehicle-count within signalized links. *Transp. Res. Part C Emerg. Technol.* **2008**, *16*, 18–35. [[CrossRef](#)]
18. Roess, R.P.; Prassas, E.S.; McShane, W.R. *Traffic Engineering*; Pearson Education Inc.: Hoboken, NJ, USA, 2011.

19. Gazis, D.; Liu, C. Kalman filtering estimation of traffic counts for two network links in tandem. *Transp. Res. Part B Methodol.* **2003**, *37*, 737–745. [[CrossRef](#)]
20. Ajitha, T.; Vanajakshi, L.; Subramanian, S. Real-time traffic density estimation without reliable side road data. *J. Comput. Civil Eng.* **2013**, *29*, 04014033. [[CrossRef](#)]
21. Chu, L.; Oh, S.; Recker, W. Adaptive Kalman filter based freeway travel time estimation. In Proceedings of the 84th TRB Annual Meeting, Washington, DC, USA, 9–13 January 2005.
22. Myers, K.; Tapley, B. Adaptive sequential estimation with unknown noise statistics. *IEEE Trans. Autom. Control* **1976**, *21*, 520–523. [[CrossRef](#)]
23. Fulari, S.; Vanajakshi, L.; Subramanian, S.C. Artificial neural network-based traffic state estimation using erroneous automated sensor data. *J. Transp. Eng. Part A Syst.* **2017**, *143*, 05017003. [[CrossRef](#)]
24. Antoniou, C.; Koutsopoulos, H.N. Estimation of traffic dynamics models with machine-learning methods. *Transp. Res. Rec.* **2006**, *1965*, 103–111. [[CrossRef](#)]
25. Khan, S.M.; Dey, K.C.; Chowdhury, M. Real-time traffic state estimation with connected vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1687–1699. [[CrossRef](#)]
26. Wassantachatt, T.; Li, Z.; Chen, J.; Wang, Y.; Tan, E. Traffic density estimation with on-line SVM classifier. In Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Genova, Italy, 2–4 September 2009; pp. 13–18.
27. Sekuła, P.; Marković, N.; Vander Laan, Z.; Sadabadi, K.F. Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A Maryland case study. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 147–158. [[CrossRef](#)]
28. Jahangiri, A.; Rakha, H.A.; Dingus, T.A. Adopting machine learning methods to predict red-light running violations. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 650–655.
29. Aerde, M.V.; Rakha, H.A. *INTEGRATION Release 2.40 for Windows: User's Guide-Volume II: Advanced Model Features*; Technical Report; M. Van Aerde and Assoc., Ltd.: Kingston, ON, Canada, 2013.
30. Dion, F.; Rakha, H.; Kang, Y.S. Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. *Transp. Res. Part B Methodol.* **2004**, *38*, 99–122. [[CrossRef](#)]
31. Rakha, H.; Kang, Y.S.; Dion, F. Estimating vehicle stops at undersaturated and oversaturated fixed-time signalized intersections. *Transp. Res. Rec.* **2001**, *1776*, 128–137. [[CrossRef](#)]
32. Chamberlayne, E.; Rakha, H.; Bish, D. Modeling the capacity drop phenomenon at freeway bottlenecks using the INTEGRATION software. *Transp. Lett.* **2012**, *4*, 227–242. [[CrossRef](#)]
33. Rakha, H.; Pasumarthy, P.; Adjerid, S. *The INTEGRATION Framework for Modeling Longitudinal Vehicle Motion*; TRANSTEC: Athens, Greece, 2004.
34. Aljamal, M.A.; Rakha, H.A.; Du, J.; El-Shawarby, I. Comparison of Microscopic and Mesoscopic Traffic Modeling Tools for Evacuation Analysis. In Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems, Maui, HI, USA, 4–7 November 2018; pp. 2321–2326.
35. Rakha, H.; Zhang, Y. INTEGRATION 2.30 framework for modeling lane-changing behavior in weaving sections. *Transp. Res. Rec.* **2004**, *1883*, 140–149. [[CrossRef](#)]
36. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1994.
37. Roweis, S. Levenberg-marquardt optimization. *Notes Univ. Toronto* **1996**, 2321–2326.
38. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the 14th international joint conference on Artificial intelligence, Montreal, QC, Canada, 20–25 August 1995.





Article

A Novel Recurrent Neural Network-Based Ultra-Fast, Robust, and Scalable Solver for Inverting a “Time-Varying Matrix”

Vahid Tavakkoli *, Jean Chamberlain Chedjou and Kyandoghene Kyamakya

Institute for Smart Systems Technologies, University Klagenfurt, A9020 Klagenfurt, Austria; jean.chedjou@aau.at (J.C.C.); kyandoghene.kyamakya@aau.at (K.K.)

* Correspondence: vtavakko@edu.aau.at; Tel.: +43-463-2700-3540

Received: 16 August 2019; Accepted: 11 September 2019; Published: 16 September 2019

Abstract: The concept presented in this paper is based on previous dynamical methods to realize a time-varying matrix inversion. It is essentially a set of coupled ordinary differential equations (ODEs) which does indeed constitute a recurrent neural network (RNN) model. The coupled ODEs constitute a universal modeling framework for realizing a matrix inversion provided the matrix is invertible. The proposed model does converge to the inverted matrix if the matrix is invertible, otherwise it converges to an approximated inverse. Although various methods exist to solve a matrix inversion in various areas of science and engineering, most of them do assume that either the time-varying matrix inversion is free of noise or they involve a denoising module before starting the matrix inversion computation. However, in the practice, the noise presence issue is a very serious problem. Also, the denoising process is computationally expensive and can lead to a violation of the real-time property of the system. Hence, the search for a new ‘matrix inversion’ solving method inherently integrating noise-cancelling is highly demanded. In this paper, a new combined/extended method for time-varying matrix inversion is proposed and investigated. The proposed method is extending both the gradient neural network (GNN) and the Zhang neural network (ZNN) concepts. Our new model has proven that it has exponential stability according to Lyapunov theory. Furthermore, when compared to the other previous related methods (namely GNN, ZNN, Chen neural network, and integration-enhanced Zhang neural network or IEZNN) it has a much better theoretical convergence speed. To finish, all named models (the new one versus the old ones) are compared through practical examples and both their respective convergence and error rates are measured. It is shown/observed that the novel/proposed method has a better practical convergence rate when compared to the other models. Regarding the amount of noise, it is proven that there is a very good approximation of the matrix inverse even in the presence of noise.

Keywords: matrix inversion; time-varying matrix; noise problem in time-varying matrix inversion; recurrent neural network (RNN); RNN-based solver; real-time fast computing

1. Introduction

Matrix inversion is extensively used in linear algebra (e.g., for solving linear equations). Although matrix inversion is already referred to in very ancient books, tremendous attention has been devoted to it (by scientists) mainly since the 17th century. The interest devoted to matrix inversion has led to the development of various methods, concepts, and algorithms for solving linear equations [1]. Solving matrix inversion is very useful in engineering, physics, and other natural sciences [2]. Solving a real-time/online matrix inversion is part of mathematics and control theory. It finds important applications in various areas such as traffic simulation and/or online control in the frame of intelligent transportation systems, robotics (e.g., for kinematics and inverse kinematics), communications [3],

machine learning [4], smart/complex antennas (MIMO) [5–7], Field Programmable Gate Array (FPGA) [8,9], signal processing [10], image processing [11,12], and robotics [13–15], etc.

Also, the matrix inversion is very useful in several decision-making algorithms. There are plenty of different heuristics and goal-programming algorithms which are using linear relationships to solve problems and they often try to solve those problems via matrix inversion; this is the case, for example, in social media networks where it can help to speed up the ranking amongst different categories [16,17].

Matrix inversion is further widely used and is of critical importance in various processing contexts in smart sensors. In some cases, the accuracy of certain measurements can be significantly improved by involving matrix inversion. For example, a real-time matrix inversion is used to ensure high-precision in multi parameter sensing while using the so-called fiber Bragg grating (FBG)-based sensors [18]. Thereby, in that related approach, the sensor functionality is approximated with a linear system and the problem is thus solved through linear system of equations. Also, this last-mentioned approximation has other advantages like the capability of re-constructing lost information, a capability used, for example, in the context of so-called compressed sensing [19,20].

In general, the measurements of different physical quantities related to a dynamical system can be explained as linear measurement equations or multi-variate sensing processes [21]. The measurement of those/such mentioned quantities often requires a real-time computing of matrix inversions. Further, the relationship between different measurements can also be used for calibrating purposes for the related sensors [22].

Finally, matrix inversion used in sensors related processing processes does also provide the capability to significantly reduce noise in measurements. For example, just for illustration, matrix inversion is helping to ignore noisy measurement in spatially distributed sensors [23].

To fulfil the expected role in various applications, different methods have been developed to achieve both fast convergence and higher accuracy of the matrix inversion related calculations. Some of the most famous methods are the following: elimination of variables, Gaussian elimination (also known as row reduction), lower-upper (LU) decomposition, Newton's method, eigenvalue decomposition, Cholesky decomposition, and many other methods.

Generally, one can categorize matrix inversion methods into two different groups: (a) the recursive (or iterative) methods; and (b) the direct methods [7,24–26].

The first group encompasses methods like Gauss–Seidel or gradient descent. The initial condition (or starting point) is provided and each step uses the last value to calculate the new value. In each iteration, the solution approximation becomes better until the desired accuracy is reached [27,28].

On the other hand, direct methods like Cholesky or Gaussian elimination typically compute the solution in finite number of iterations. They can find the exact solution if there exists no rounding error [29]. Those analytical methods normally have a minimum arithmetic complexity of $O(n^3)$ (provided those algorithms are implemented on a single CPU). For parallel systems, this arithmetic complexity is different, as the algorithm should be changed in a way to fit for efficient processing on a parallel system architecture. Hereby, depending on the parallelizability potential of the algorithm, the present number of parallel cores will affect the final effective complexity. However, in real benchmarking implementations, most parallel implementations of algorithms do require the transfer of large amounts of data amongst processors and thereby this communication need does lead to a loss of efficiency of the algorithm with respect to speeding-up capability in presence of multiple cores. Therefore, implementing the same algorithm on a parallel system is very inefficient (i.e., speed-up). On the other hand, in practice, the noise problem is a very serious problem and the related denoising process is an expensive one, which can provoke a violation of the systems' real-time property. The need for developing a new concept to solve this concern is therefore sufficiently justified.

For answering the above described parallelizability problem, several parallel concepts have been introduced, one good example being the so-called cellular neural network (CNN). It has been introduced by Leon Chua in 1988 in the seminal paper entitled: "Cellular Neural Networks: Theory" [30]. In 1993, a further article entitled: "The CNN Universal Machine: An Analogic Array Computer", was published

by Tamas Roska and Leon Chua, where the first analog CNN processor was presented [31]. Since then many different articles have been published to show the applicability of CNN processors for various usages.

In this article, we use a more generic concept which is called recurrent neural network (RNN), which is basically a more general neural network family to which CNN belongs. Essentially, they (i.e., RNN) are building elements of the so-called dynamic neural network (DNN) [32]. The parallel computing nature of RNN and the inherent fast processing speed while solving problems make it a very good basic brick of a novel concept for efficiently solving differential problems on multiple cores [33].

An RNN processor, similarly to its ancestors (artificial neural networks), is a set of cells, which do have mathematical relations (i.e., coupling) with their respective neighbors. The dynamic property of each cell is expressed by one differential equation. Furthermore, the dynamic property of the network can be customized for solving a given ODE (ordinary differential equation) by appropriate values of ODE's parameter settings (or coefficients) expressed in the form of matrices called "templates".

Playing with templates does provide the possibility to solve different kind of problems without changing the physical property and/or the architecture of the RNN machine or processor.

The inherent flexibility and the fast processing speed while solving problems with RNN does provide two advantages to solve problems when compared to traditional or competing computational methods or concepts [34–36]. First, we can solve different kind of problems by changing templates. Second, the problem solving can be significantly accelerated (see speed-up) without losing accuracy.

The above-mentioned good features of RNN (e.g., flexibility, speed-up (in presence of multiple cores), etc.) motivate the use of this promising paradigm for an accelerated solving of linear algebraic equations on either one-core or multi-core platforms. This is not a simple conversion as we do face completely different hardware computing frameworks with different parameters to be taken care of. All RNN parameters should be adjusted such that the resulting new dynamical system does converge to the target solution of the problem.

In this paper, we do need to formulate the central problem (i.e., realizing a "matrix inversion") in a manner such that the final state of the RNN system is the solution of the target linear algebraic system of equations; see Equation (1).

$$M(t)X = I \quad (1)$$

In more detail, we define the linear algebraic equations system as Equation (1). Hereby, $M(t)$ is a $n \times n$ matrix of smooth time varying real numbers and X is a $n \times n$ matrix, I is identity matrix of size $n \times n$. We would like to find X such that Equation (1) is satisfied. Our target is to find corresponding RNN templates, which are such that for any initial value problem (IVP), after a finite number of iterations, our dynamic system (see Equation (2)) converges to a solution X^* , which approximates X .

$$\begin{aligned} \dot{X} &= F(X) \\ \forall X \in \mathbb{R}_{n \times n}, \lim_{t \rightarrow \infty} MX - I &= 0 \end{aligned} \quad (2)$$

In Equation (2), $\dot{X} = F(X)$ is showing an ODE in a general form. The second part of Equation (2) is showing our problem that we wish to convert into the form of an ODE solving.

There exist several published works, in which one has tried to solve similar problems with dynamical systems like recurrent neural networks (RNN) [13,37–43] or artificial neural networks (ANN) [44–47], etc. Although most of those works are also related to RNN, our work and the novel concepts presented in this paper have more potential to reliably provide faster convergence towards the solution of Equation (1).

One of the main differences between our method and other related works, especially RNN concepts, is that the proposed model does need training like this required for most of RNN networks required. Our model is converging to the problem's solution if and only if a correct selection/setting

of internal dynamic parameters is done. Therefore, this functionality makes our model completely unique amongst other types of RNN models, which are usually used in machine learning [48].

For training, most of the published related works use common traditional methods like gradient descent [39] to create dynamical systems and do then customize them for an implementation on a target platform.

In this paper we do compare the performance of the own novel method developed with those already implemented traditional methods from the relevant literature. This does provide the basis for a fair judgement of both advantages and disadvantages of each method, old ones versus our new one.

The overall structure of this paper is as follows. Section 2 contains both background and a comprehensive overview of previous dynamic system models used for matrix inversion. Besides, related requirements with respect to the target platform for implementing the dynamic system (RNN, ANN ...) are discussed. In Section 3, we do formulate our problem (i.e., inverting a time-varying matrix) with required restrictions related to RNN. The effectiveness of the developed model for solving the matrix inversion problem will be demonstrated in Section 4. The main proof and interesting result are the one showing the global (fast) convergence of the RNN-based dynamic system to the final solution of Equation (1).

In Section 5, the new method will be used to create a new dynamical system, i.e., an RNN model. The performance of this novel RNN will be compared to that of previous/original concepts.

In the last section (i.e., Section 7), some concluding remarks summarize the quintessence of the key achievements in this work.

2. Related Works of Dynamical Neural Networks

Solving Equation (1) using any dynamic method like the so-called dynamic neural network (DNN) requires creating a dynamic system with an attractor in that system (if and only if Equation (1) has a real solution), which is a fix point equal to the solution of Equation (1). Such systems can be implemented by different ways, but the most famous way of creating such dynamic systems is to use either the gradient descent method, the Zhang dynamics, or the Chen dynamics. All three named methods do essentially use the same concept, but both the second and the third named methods do use smaller step sizes and therefore more memory is required. This can improve/speed-up the convergence of the algorithms. We will be providing (in the next sections) a full explanation of the advantages of each method; afterwards a generic new method for solving Equation (1) will be provided.

2.1. The Gradient Method

This method involves a first-order optimization technique for finding the minimum point of a function. This technique takes steps in the direction of the negative gradient. It is based on the observation of where the multivariate function $F(X)$ decreases the fastest if we choose the negative gradient of $F(X)$ at point a : $-\nabla F(a)$.

$$b = a - \gamma \nabla F(a), \forall \gamma \in \mathbb{R} \text{ and } \gamma > 0 \quad (3)$$

γ is the step size for updating decision neurons. For small γ , $F(a) \geq F(b)$; with this remark, we can extend the observations by Equation (4).

$$X_{n+1} = X_n - \gamma \nabla F(X_n) \quad (4)$$

That means, by iterating Equation (4), $F(X)$ will be converging to the minimum point of the function F .

The gradient descent (see Equation (5)) can also be described as the “Euler method” for solving ordinary differential equations.

$$\dot{X}(t) = -\gamma \nabla F(X(t)) \quad (5)$$

Gradient descent can be used to solve linear equations. In that case, the problem is reformulated as the quadratic minimization of a function which is then solving it with gradient descent method.

$$F(X) = \frac{1}{2} \|MX - I\|^2 \tag{6}$$

Then, based on Equation (6), we have:

$$\nabla F(X) = M^T(MX - I) \tag{7}$$

By substituting Equation (7) into Equation (5) the following dynamic model is obtained:

$$\dot{X}(t) = -\gamma M^T MX + \gamma M^T I \tag{8}$$

The exact analytical solution of Equation (8) is expressed as the following:

$$X(t) = (X(0) - M^{-1}) e^{-\gamma M^T M t} + M^{-1} \tag{9}$$

Whereby the first derivative of $X(t)$ denoted by $\dot{X}(t)$ will be:

$$\dot{X}(t) = -\gamma (X(0) - M^{-1}) (M^T M + \dot{M}^T M t) e^{-\gamma M^T M t} \tag{10}$$

Equations (8) and (1) converge to the same solution described by the point M^{-1} .

$$\forall \gamma, a_{i,j} \in \mathbb{R} \text{ and } \gamma > 0, \text{ then } \gamma M M^T > 0$$

This method is also called “gradient-based” dynamics. It can be designed by norm-based energy functions [45,49]. The advantage of this model is its easiness of implementation, but due to the effective factor of convergence which can be seen in Equation (9) it takes (a long) time to converge to the solution of the problem, and this convergence rate has an effect on noise sensitivity of the model as it makes the model more sensitive to the noises. Also, this model is not appropriate for real-time matrixes.

2.2. Zhang Dynamics

There exists another method to create [13,50] a dynamic system converging to the required solution. In this method, the error function is defined in the following way:

$$E(t) = M(t)X(t) - I \tag{11}$$

This means the error will be increasing proportionally to the amount of divergence of X from its true solution. Larger values of the difference will create larger values of the error.

Equation (11) is changing over time in a way such that the result will be the same as Equation (2). This requires that one moves against errors to come closer to the solution. Therefore, one can define the derivation of this function as following:

$$\dot{E}(t) = -\gamma E(t) \tag{12}$$

The solution of Equation (12) can be expressed as Equation (13). This equation is showing how the error function is decreasing exponentially in a reverse direction to the errors. This will force the function to converge to the solution. Using this dynamic system is helping to create a new dynamic system for our problem.

$$E(t) = C e^{-\gamma t} \tag{13}$$

By substitution of $E(t)$ and $E'(t)$ into Equation (12) we will obtain a new dynamic system [11].

$$M\dot{X} + \dot{M}X = -\gamma(MX - I) \quad (14)$$

Whereby the solution of Equation (14) expressed as follows:

$$X(t) = Ce^{-\gamma t} + M^{-1} \quad (15)$$

When we make a first derivation of Equation (15) we will have:

$$\dot{X}(t) = -C\gamma e^{-\gamma t} \quad (16)$$

By combining Equations (15) and (16) we derive Equation (14). It is observed again that by increasing t to infinite our dynamic system will be converging to the solution of Equation (1). This model has a very good convergence rate and this will solve the problem of noise sensitivity. On the other hand, it is a model which is much more difficult to implement.

2.3. Chen Dynamics

This method is a combination of the two previously described methods. It assumes matrix M is not changing during time; therefore, the time-derivation of M is zero. If we multiply Equation (8) by M and then sum it with Equation (14) we will obtain a new dynamic system [39]:

$$M\dot{X}(t) = -\gamma M^T M (MX(t) - I)M\dot{X}(t) = -\gamma(M^T M + I)(MX - I) \quad (17)$$

Solving Equation (17) lead to the following solution:

$$X(t) = -C M^{-1} e^{-(M^T M + I)t} + M^{-1} \quad (18)$$

A derivation of Equation (18) leads to following dynamical system:

$$\dot{X}(t) = C (M^T + M^{-1}) e^{-(M^T M + I)t} \quad (19)$$

By combining Equation (17) in Equation (18) leads to Equation (19).

If $M^T M > 0$ we can state and confirm that this method has a better convergence rate than the first and second above mentioned ones [39]. As we see in Equation (19) t is multiplied with $(M^T M + I)$ in the exponent term, which produces a larger exponent value than the ones for the previous two methods. This model is better than the previous model. It has a very good convergence rate and its sensitivity to noise is very low. On the other hand, its implementation is difficult as it has more coefficient to calculate with respect to the other methods and it does not fit for a real-time matrix inversion (i.e., for inverting a time-varying matrix).

2.4. Summary of the Main Previous/Traditional Methods

By comparing the properties of the above presented methods, there is one big difference amongst them. Table 1 shows the major differences between those models by using 4 different criteria. The convergence rate refers to the convergence during time during the solving process. The "time-varying time matrix inversion" criteria refers to the ability of a given model to be usable for the case of the inversion of a time-varying matrix. The "implementation" criteria refer to how easy it is to implement a given model on RNN machines/processors. And the last criteria in Table 1 refers to how far a given model is sensitive to noise that is present in the time-varying matrix values. This last-mentioned criterion is very important because it does express the resilience of a given model to noise, which is always present in analog computing signals. Although noise is relatively low in digital systems, digital

systems do however introduce a noise-equivalent signal distortion originating from computational rounding of numbers as they are digitally represented and processed with in a fix-size digital arithmetic.

The Zhang model does have a fixed convergence rate over to time. But the gradient descent model does contain a coefficient which can be changed to influence (increase or decrease) the convergence rate. However, the Chen model is much better than the two previous ones as it does potentially provide a much better convergence rate.

The convergence rate has a direct impact on the noise sensitivity of a given model. Indeed, an increase of the convergence rate does decrease the noise sensitivity. Therefore, the Chen model has highest level of stability with respect to noise, although it is the more complex to implement.

Furthermore, amongst the 3 models listed in Table 1, only the Zhang model offers the capability of solving a time-varying matrix (i.e., a real-time matrix inversion).

Table 1. Comparison of different types of DNN (dynamic neural network) concepts (the traditional ones) for matrix inversion.

| Criteria/Method | Gradient NN | Zhang NN | Chen |
|-------------------------------|-----------------------|-----------|---------------------------|
| Convergence rate | $e^{-\gamma M^T M t}$ | e^{-t} | $e^{-\gamma(M^T M + I)t}$ |
| Time-varying matrix inversion | not available | Available | not available |
| Implementation | Easy | Hard | very hard/difficult |
| Noise sensitivity | High | Low | very low |

3. Our Concept: The Novel RNN Method

According to Chen [39], his model is converging to the solution of Equation (1) under any initial value. One can however re-formulate the Chen model [39] as result of following goal function:

$$\min \{Z = \|(X - A^{-1})\|^2 + \|(AX - I)\|^2\} \tag{20}$$

We can add another positive statement to Equation (20); thus, we multiply the last term of Equation (20), i.e., the term $((AX - I)^2)$, with the matrix A and we add it again to the function Z .

$$\min \{Z = \|(X - A^{-1})\|^2 + \|(AX - I)\|^2 + \|(A^T AX - A^T)\|^2\} \tag{21}$$

After adding that new term to the right side of Equation (20) and solving Z (according to Equation (21), one does find/obtain the following dynamical system:

$$M\dot{X}(t) = -\gamma((M^T M)^2 + M^T M + I)(MX - I) - \dot{M}X \tag{22}$$

The solution of this equation (see Equation (22)) can be expressed as follows:

$$X(t) = -C.M(t)^{-1}e^{-\gamma \int_0^t ((M^T M)^2 + M^T M + I)dz} + M(t)^{-1} \tag{23}$$

In Equation (23), when times goes to infinite the limit of X will converge to $M(t)^{-1}$ and it thereby provides the solution for Equation (1). C is a constant value (a matrix), which is added during/while solving Equation (22); see Table 2 for an illustration. Newly added terms $(M^T M)^2 + M^T M$ produce a better convergence rate. The main reason of this convergence rate is the positive value of the integral and it provides additional factors when compared to the previous time-varying models. Therefore, by adding more coefficients to the right-hand side of Equation (22), we can obtain the following equation:

$$M\dot{X}(t) = -\gamma \left(\sum_{i=0}^n (M^T M)^i \right) (MX - I) - \dot{M}X \tag{24}$$

Equation (24) is more general and can create the model of Equation (22), which is a specific configuration of it. For this, we just need to set the value of parameter n to 2.

Theorem 1. For any given nonsingular matrix $M \in \mathbb{R}^{n \times n}$, the state matrix $X(t) \in \mathbb{R}^{n \times n}$, while starting from any (initial value problem) IVP $X(0) \in \mathbb{R}^{n \times n}$, Equation (24) will achieve global convergence to $X^*(t) = M^{-1}(t)$.

Proof of Theorem 1. Let define $E(t) = X(t) - X^*(t)$ be the error value during the process for finding the solution. If this equation is multiplied by M , it does lead to $M(t)E(t) = M(t)X(t) - M(t)X^*(t)$ or $M(t)E(t) = M(t)X(t) - I$. Thus, a derivation of the error function will lead to $M\dot{E}(t) + \dot{M}E(t) = M\dot{X}(t) - \dot{M}X(t)$. By replacing this in Equation (24) we obtain the following expression:

$$M\dot{E}(t) = -\gamma \left(\sum_{i=0}^n (M^T M)^i \right) M E(t) - \dot{M} E(t) \tag{25}$$

Let us define the Lyapunov function $\epsilon(t) = ME(t)$, which is always a positive function. The derivative of this function can be obtained as follows:

$$\dot{\epsilon}(t) = E(t)^T M^T M \cdot \frac{dE(t)}{dt} + E(t)^T M^T \frac{dM(t)}{dt} E(t) \tag{26}$$

By replacing Equation (25) into Equation (26) it leads to the following:

$$\dot{\epsilon}(t) = -\gamma E(t)^T \left(\sum_{i=0}^n (M^T M)^{i+1} \right) E(t) \tag{27}$$

Hence:

$$\dot{\epsilon}(t) = -\gamma E(t)^T M^T \left(\sum_{i=0}^n (M^T M)^i \right) ME(t) \tag{28}$$

One can replace middle term $\sum_{i=0}^n (M^T M)^i$ with large enough value of μ therefore:

$$\leq -\gamma \mu \|ME(t)\| \leq 0 \tag{29}$$

Thus, it appears that $\dot{\epsilon}(t)$ is always negative; furthers, $\dot{\epsilon}(t) = 0$ if and only if $X(t) = X(t)^*$ is satisfied. Therefore, our differential equation globally converges towards a point (matrix), which is the equilibrium point for this function. □

Equation (30) is the result of an analytical solving of Equation (24). Increasing t in this equation will lead to the solution of the algebraic equation (i.e., of Equation (1)).

$$X(t) = -C \cdot M^{-1} e^{-\gamma \sum_{i=0}^n \int_0^t (M^T M)^i dz} + M^{-1} \tag{30}$$

In this equation, C is a constant value (matrix) and it is added during solving differential equation. Obviously, this equation has a much better rate of convergence when compared to the previous implementations and, like previous solutions/concepts, the minimum value of the eigenvectors of $M^T M$ should be positive.

Also, according to the Chen model, if Equation (24) is extended by introducing a monotonically increasing function F where $F(0) = 0$, here again our system will be converged to solution of

Equation (1). Thus, by introducing a function F in the Equation (24) the following new equation will be obtained, see Equation (31):

$$M\dot{X}(t) = -\gamma\left(\sum_{i=0}^n (M^T M)^i\right)F(MX - I) - \dot{M}X \tag{31}$$

Theorem 2. For any given nonsingular matrix $M \in \mathbb{R}^{n \times n}$, the state matrix $X(t) \in \mathbb{R}^{n \times n}$, while starting from any IVP (initial value problem) $X(0) \in \mathbb{R}^{n \times n}$ and with a monotonically increasing function F where $F(0) = 0$, Equation (31) will achieve global convergence to $X^*(t) = M^{-1}(t)$.

Proof of Theorem 2. Let define $E(t) = X(t) - X^*(t)$ for the error value during the process for finding the solution. If this equation is multiplied by M, it does lead to $M(t)E(t) = M(t)X(t) - M(t)X^*(t)$ or $M(t)E(t) = M(t)X(t) - I$. Thus, a derivation of the error function will lead to $M\dot{E}(t) + \dot{M}E(t) = M\dot{X}(t) - \dot{M}X(t)$. By replacing this in Equation (31), we obtain the following expression:

$$M\dot{E}(t) = -\gamma\left(\sum_{i=0}^n (M^T M)^i\right)F(ME(t)) - \dot{M}E(t) \tag{32}$$

Let's define the Lyapunov function $\epsilon(t) = ME(t)$, which is always a positive function. The derivative of this function can be obtained as follows:

$$\dot{\epsilon}(t) = E(t)^T M^T M \cdot \frac{dE(t)}{dt} + E(t)^T M^T \frac{dM(t)}{dt} E(t) \tag{33}$$

By replacing Equation (33) into Equation (32) it does lead to:

$$\dot{\epsilon}(t) = -\gamma E(t)^T M^T \left(\sum_{i=0}^n (M^T M)^i\right) F(ME(t)) \tag{34}$$

Hence:

$$\dot{\epsilon}(t) = -\gamma E(t)^T M^T \left(\sum_{i=0}^n (M^T M)^i\right) F(ME(t)) \tag{35}$$

One can replace middle term $\sum_{i=0}^n (M^T M)^i$ with large enough value of μ therefore:

$$\leq -\gamma\mu E(t)^T M^T F(ME(t)) \leq 0 \tag{36}$$

In the last equation, $E(t)^T M^T F(ME(t))$ is always positive because if $ME(t)$ becomes negative, $F(ME(t))$ also becomes negative and vice-versa.

Thus, it appears that $\dot{\epsilon}(t)$ is always negative; and $\dot{\epsilon}(t) = 0$ if and only if $X(t) = X(t)^*$ is satisfied. Therefore, our differential Equation (31) or (32) globally converges towards a point (matrix), which is the equilibrium point for this function. □

By choosing different forms of the function F, one can create various dynamical properties to be expressed by this model.

Examples of functions for F are: sigmoid, linear, square, cubic, arcs, etc. All these functions are suitable to be used in Equation (31) as all of them are increasing monotonic functions and they do all satisfy the $F(0) = 0$ condition (see Figure 1).

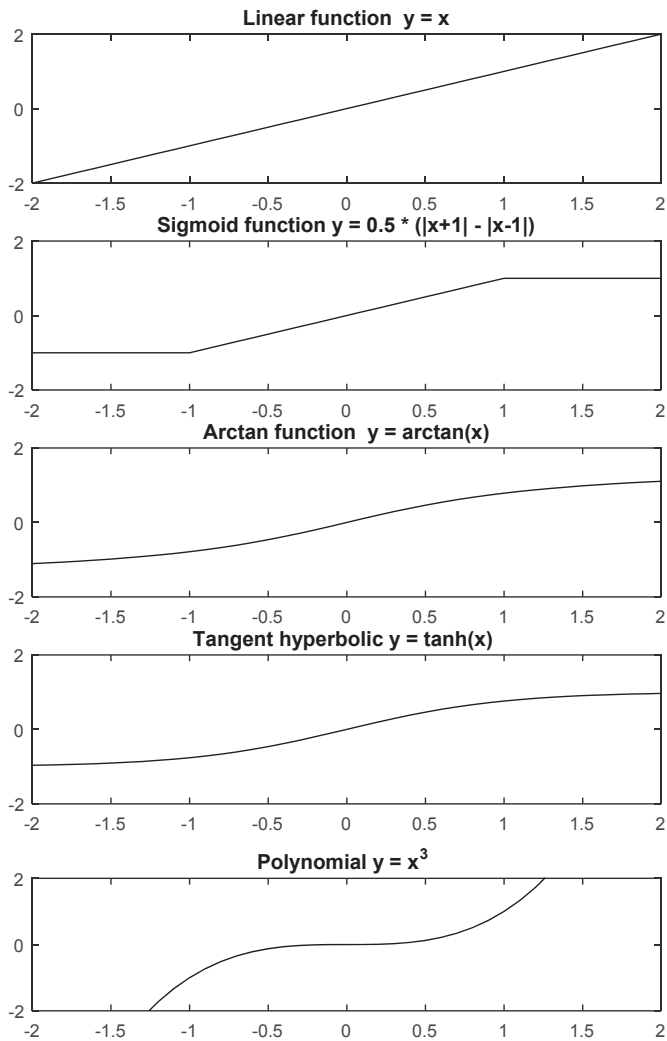


Figure 1. Illustrative examples of monotonic functions which can be used for solving the inversion of a time-varying matrix through Equation (31).

4. Model Implementation in SIMULINK

Equation (24) or Equation (31) can be implemented directly into SIMULINK (see Figure 2). This dynamic model has the components shown in Table 2.

If M is not a time-varying matrix we can simply put zero values in the M' matrix, otherwise M' model should be a corresponding derivative of matrix M . Executing the model will result in the following output in SIMULINK, see Figure 2, Figure 3, and Table 2, which is the solution of Equation (1). Therefore, it works well as we expected and gives the solution of Equation (24).

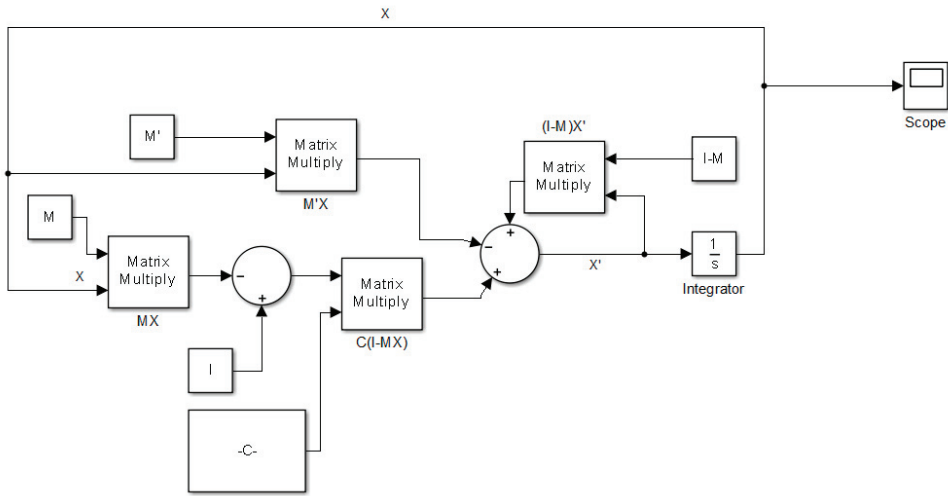







Figure 2. The RNN block diagram corresponding to Equation (24). Note that the matrix to be inverted is M.

Table 2. Components of SIMULINK model to implement Equation (24).

| Component | Description |
|---|---|
|  | $\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 2 & 6 & 1 \end{bmatrix}$ |
|  | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ |
|  | I-M |
|  | Time-derivative of M; if M is constant, just put zero values in the matrix M': $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |
|  | C is a weight value matrix equal to $\sum_{i=0}^n (M^T M)^i$ (see Equations (24) and (31)) For the setting n=2, we do have the following value for C: $M^T M M^T M + M^T M + I$ Note: If the matrix M is time-varying, C will no more be constant and will also be time-varying. |

| | | |
|------|----------|---------|
| -1 | 0.4444 | -0.1111 |
| 0.25 | -0.1389 | 0.2222 |
| 0.5 | -0.05555 | -0.1111 |

Figure 3. Result of model simulation (just for illustration) for the matrix M indicated in Table 2 (see first row of Table 2), SIMULINK output.

5. Illustrative Examples

In this section we consider some numerical examples to demonstrate the performance of our RNN implementation for solving a system of linear algebraic equations.

5.1. Illustrative Example 1

Let us define M and V as follows: Here, instead of using the identity matrix in Equation (1) we use a vector V. This does result in a linear system of equations.

$$M = \begin{bmatrix} -3.0755 & 0.5004 & 3.8135 \\ 0.0739 & -2.2382 & -4.7532 \\ -4.7576 & 1.9624 & -1.5879 \end{bmatrix}, V = \begin{bmatrix} 7.920 \\ 7.983 \\ 9.633 \end{bmatrix} \tag{37}$$

Thus, one can find X as following:

$$X = M^{-1}V = \begin{bmatrix} -3.330 \\ -3.305 \\ -0.175 \end{bmatrix} \tag{38}$$

This above given value of X is the target value (ground truth). In the following, we shall see how far how fast the models developed (Equation (24) and/or Equation (31)) do reach well this target value.

M is not singular and therefore can be inverted. Thus, our system of equations has one unique solution. C, as explained in the previous section, is a weight values-matrix which is calculated using the following formula:

$$C = \sum_{i=0}^n (M^T M)^i \tag{39}$$

Increasing the value of n will increase the convergence rate of Equation (24) or Equation (31) or (29). Here, we choose the value n = 3. The corresponding RNN parameters are defined as follows (see Figure 3):

$$C = M^T M M^T M M^T M + M^T M M^T M + M^T M + I$$

$$C = 10,000 \times \begin{bmatrix} 4.6283 & -2.2035 & -2.7439 \\ -2.2035 & 1.3283 & 2.5763 \\ -2.7439 & 2.5763 & 7.5178 \end{bmatrix} \tag{40}$$

$$X_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

These parameters are implemented in the previously described model in Simulink (see Figure 3) for numerical simulations and F (for simplicity) is taken as a linear function (Figure 1).

Figure 4 is showing the system simulation results during the time interval $t = [0, 0.5]$. The output (value of state vector) at the end of this time interval is the following:

$$X(0.5) = \begin{bmatrix} -3.329 \\ -3.304 \\ -0.1749 \end{bmatrix} \quad (41)$$

This above obtained result shows that the output of our system model is very close to the analytical solution and the difference is small and approximately 0.001, which can furthermore be corrected but adjusting the step size.

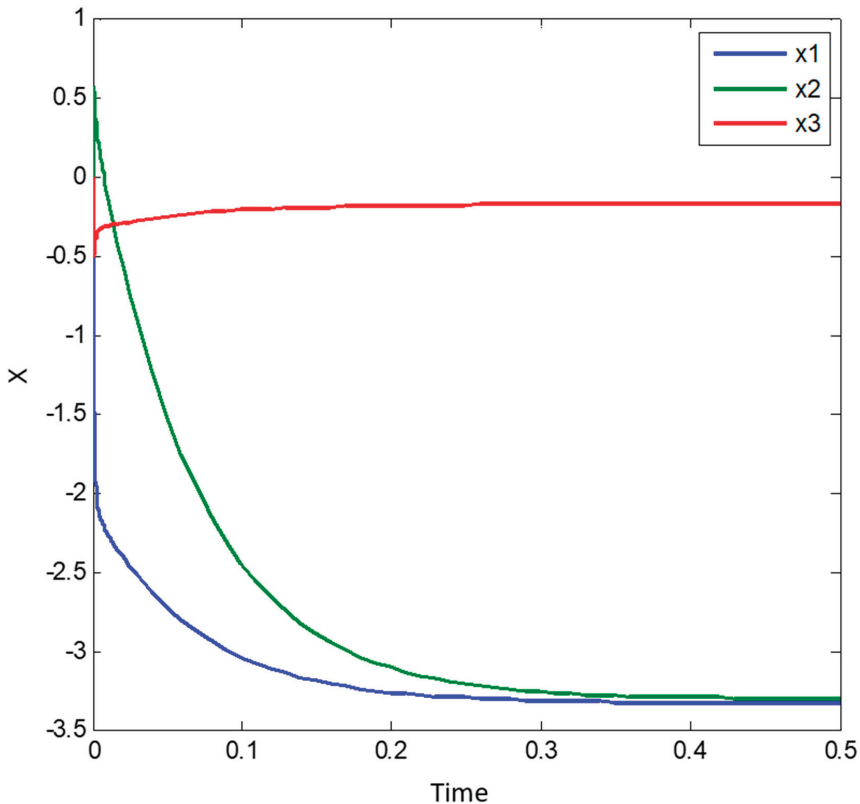


Figure 4. Result of our model simulation for solving the illustrative example with respect to solving an algebraic system of linear equations. Please note in this case we have taken $n = 3$ in Equation (24) or Equation (31).

5.2. Illustrative Example 2

Now we do the same experience for calculating the inverse of a time-varying matrix M . The matrix M is defined as follows:

$$M = \begin{bmatrix} \sin(t) & -\cos(t) \\ \cos(t) & \sin(t) \end{bmatrix} \quad (42)$$

the matrix M is always invertible and it does satisfy the conditions of Equation (1). Therefore, we can use it for testing our dynamic system model.

$$M^T = M^{-1} = \begin{bmatrix} \sin(t) & \cos(t) \\ -\cos(t) & \sin(t) \end{bmatrix} \tag{43}$$

Also, we define F as a linear function (Figure 1).

As we have explained previously (see Table 2), we calculate the weight C again for $n = 3$ as the following:

$$C = M^T M M^T M M^T M + M^T M M^T M + M^T M + I \tag{44}$$

Now all parameters are introduced in the Simulink model to simulate our system model (see Equation (24)). The simulation is done for $t = [0, 2.0]$. Figure 5 does contain benchmarking results for the first element of the time-varying matrix (i.e., the matrix element in first row, first column). Here (see Figure 5) the speed of convergence is compared to that of the original Zhang model. Thereby, for our model we consider different values of the parameter n . One can clearly see that our novel model strongly outperforms the original Zhang model. Table 3 shows the difference very clearly: Here, it is clear that by choosing $n = 3$ our model will converge to the problem solution 4 times faster than the ZNN model; notice that the ZNN model corresponds to $n = 0$ in Table 3. Amount of memory for saving this model is very small as we need to save main templates parameter of dynamic model in Equation (24) or Equation (31). Therefore, as see in Table 3, the memory usage is very small and it can be implemented on small computers.

Table 3. Comparison of performance of time varying matrix inversion with change of parameter n of Equation (24) or Equation (31). Please note $n = 0$ is ZNN model.

| Criteria/n | 0 | 1 | 2 | 3 |
|--|-----|-----|------|-------|
| Convergence Time to MSE 0.01 | 8.4 | 3.8 | 2.8 | 1.9 |
| MSE in $t = 2.0$ | 3.4 | 0.4 | 0.06 | 0.008 |
| Approximated estimation of memory usage in Bytes | 96 | 128 | 128 | 128 |

5.3. Illustrative Example 3

In the last illustrative experiment, we try to test and analyze the effect of changing the function F in Equation (31) on the convergence rate of our model.

For this test, we take the same parameter setting as in the first experience (see illustrative example 1), but we change the function F and calculate coefficients for $n = 3$. The selected functions for this test are the following ones: linear function ($Y = X$), sigmoid function ($Y = \frac{1}{2}(|X + 1| - |X - 1|)$), arctangent function ($Y = \arctan(X)$), tangent hyperbolic function ($Y = \tanh(X)$), and polynomial function ($Y = X^3$). All other parameters are fixed such to show the correct properties of the functions. As we can see in the results of this simulation experiment (see Figure 6) the polynomial function is showing the very best convergence amongst all considered functions. Therefore, it is evident that selecting the appropriate function can help our model to converge faster to the solution of the matrix inversion problem. It is also evident that higher values of n will perform much better.

Table 4 is showing our simulation result until $t = 0.05$ for this illustrative example 3. It does clearly show the effect of using different functions on the system/model convergence. Both polynomial and linear functions are the best functions for this task, while both sigmoid and tanh functions are the worst functions to be used in this context for matrix inversion.

Table 4. Performance comparison of different function types used in Equation (31), whereby all are monotonic increasing functions. The polynomial function shows the better convergence compared to the other functions.

| Criteria/ Function used in Equation (31) with $n = 3$ | Linear | Sigmoid | Arctan | Tanh | X^3 |
|--|--------|---------|--------|------|-------|
| MSE at $t = 0.05$ | 0.075 | 7.90 | 4.58 | 7.83 | 0.062 |

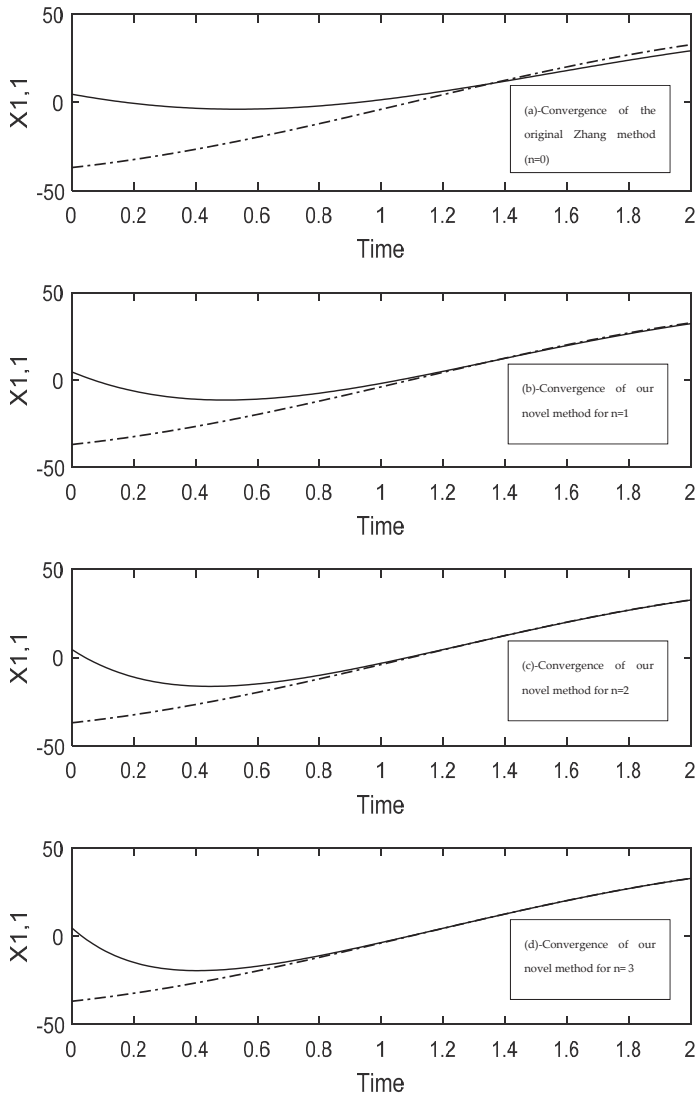


Figure 5. Showing the difference in convergence speed between the original Zhang method and our new developed method (see Equation (24)). The solid lines are the one obtained from the model solving and the dashed lines are the target values. (a) convergence of the original Zhang model; (b) until (d): convergence profiles of our novel method for different values of n . All graphs (a) until (d) are plotted for the first element of the time-varying matrix M (i.e., first row, first column).

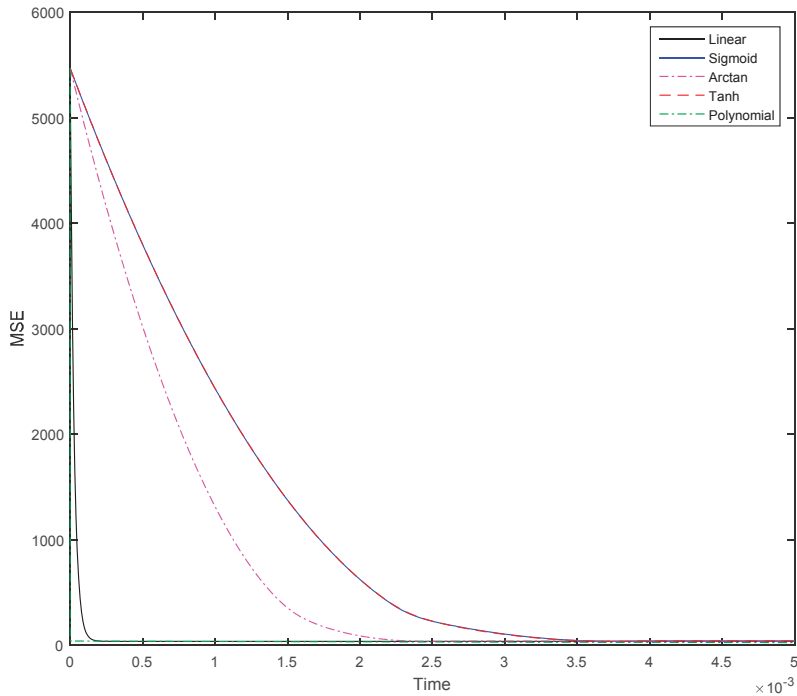


Figure 6. Showing differences in convergence speed while considering different function types for F in Equation (31). Please note that n is 3 for the polynomial function F .

The superiority with respect to convergence speed of the polynomial function F is very evident.

6. Comparison of Our Novel Method with Previous Studies

In this section we compare our novel model with the following related methods which are well-known from the relevant literature: gradient descent method, Zhang dynamics, and Chen dynamics. Hereby, 1000 different static random matrices are generated for use in the experiment. Finally, we then sum up the results obtained as shown in Figure 7. Figure 7 does display how the error converges towards zero when the different models are executed over time; the intention is to hereby illustrate the convergence speed of all considered models. Hereby, the error function is calculated by using the formula $\|MX - I\|$.

By comparing the different methods with our method (Figure 7), the fastest convergence of our method to the exact solution is observed. Specifically, our method is at least 3 times faster than the Chen method while implemented on CPU (the results of Figure 7 were obtained on CPU). An implementation on a multiple-core platform should display a much higher relative and comparative speed-up performance of our model. Obviously, by using more coefficients (i.e., higher values of n) in Equation (24) we can reach a much better convergence rate.

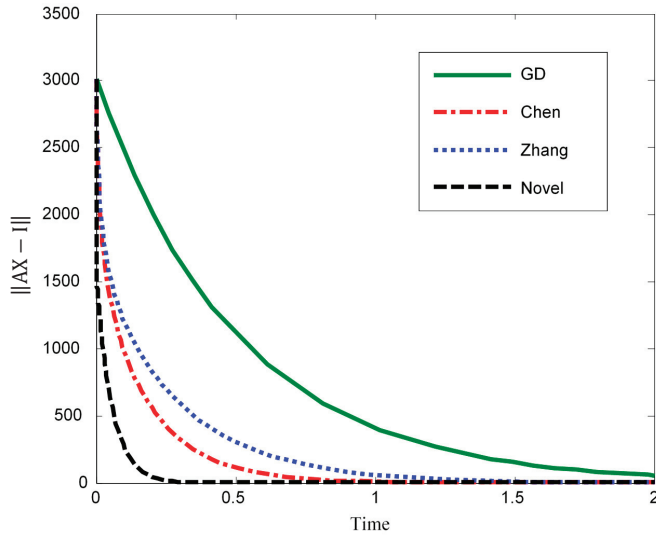


Figure 7. Comparison of different DNN models with respect to convergence speed under the same conditions—benchmarking. In this figure, GD refers to the “gradient descent” method. Our novel method (see novel in the figure; Equation (24)) is calculated for $n = 3$. The fast convergence of our method to the exact solution is clearly demonstrated. It is also clear that this convergence would be much faster if higher values of n are taken ($n > 3$).

7. Conclusions

A novel method for solving the very important “matrix inversion” problem has been developed and validated in this paper. Consequently, it can also solve linear algebraic systems of equations. The concept developed is essentially an RNN-based analog computing machine.

We have compared this novel method with other dynamical systems methods from the relevant literature. It has been demonstrated that our novel method does theoretically have an exponential convergence rate towards the exact solution of the problem for any IVP (initial value problem). Also, the convergence rate of this method is much higher than those of other related competing concepts.

It is further possible to customize this novel model in order to enable its implementation on a cellular neural network processor machine. This has previously been done in our previous works.

For validation we have extensively compared our method with other relevant competing methods like gradient descent, Zhang, and Chen methods in one large simulation experiment. Hereby, we have considered the following scenarios: 1000 random matrixes of M are generated and applied in the different dynamical system models for matrix inversion. For each time-point, errors reached are summed-up using the formula $\|MX - I\|$.

It has been clear that while using our novel method we do visibly reach a significant speed-up even on one single CPU, this compared to the other methods. A use of more coefficients (i.e., higher values for n in model Equations (24) and (31)) will surely result in an even much higher convergence rate. On the other hand, however, if we use more coefficients, we would need more preparation time to create the templates and consume more computing resources for running the (our novel) RNN processor dynamical model. Therefore, it is recommended to reach a balance/tradeoff between convergence rate and number of required coefficients (i.e., value of n).

To finish, the last experiments have demonstrated that using the polynomial function for F in Equation (31) does lead to a clearly much higher convergence rate when compared to the other types of function. Further, the higher the polynomial order, the best.

Author Contributions: Conceptualization, V.T. and K.K.; Methodology, J.C.C. and K.K.; Software, V.T.; Validation, V.T., J.C.C. and K.K.; Formal Analysis, V.T.; Investigation, V.T.; Resources, V.T.; Data Curation, V.T.; Writing—Original Draft Preparation, V.T.; Writing—Review & Editing, J.C.C. and K.K.; Visualization, V.T.; Supervision, J.C.C. and K.K.; Project Administration, K.K.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Lumpkin, B. *Algebra Activities from Many Cultures*; J. Weston Walch: Portland, ME, USA, 1997.
2. Song, W.; Wang, Y. Locating Multiple Optimal Solutions of Nonlinear Equation Systems Based on Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 414–431. [[CrossRef](#)]
3. Wang, Y.; Leib, H. Sphere Decoding for MIMO Systems with Newton Iterative Matrix Inversion. *IEEE Commun.* **2013**, *17*, 389–392. [[CrossRef](#)]
4. Gu, B.; Sheng, V. Feasibility and Finite Convergence Analysis for Accurate On-Line ν -Support Vector Machine. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1304–1315.
5. Eilert, J.; Wu, D.; Liu, D. Efficient Complex Matrix Inversion for MIMO Software Defined Radio. In Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007.
6. Wu, M.; Yin, B.; Vosoughi, A.; Studer, C.; Cavallaro, J.R.; Dick, C. Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink. In Proceedings of the ISCAS2013, Beijing, China, 19–23 May 2013; pp. 2155–2158.
7. Ma, L.; Dickson, K.; McAllister, J.; McCanny, J. QR Decomposition-Based Matrix Inversion for High Performance Embedded MIMO Receivers. *IEEE Trans. Signal Process.* **2011**, *59*, 1858–1867. [[CrossRef](#)]
8. Arias-García, J.; Jacobi, R.P.; Llanos, C.H.; Ayala-Rincón, M. A suitable FPGA implementation of floating-point matrix inversion based on Gauss-Jordan elimination. In Proceedings of the VII Southern Conference on Programmable Logic (SPL), Cordoba, Argentina, 13–15 April 2011; pp. 263–268.
9. Irturk, A.; Benson, B.; Mirzaei, S.; Kastner, R. An FPGA Design Space Exploration Tool for Matrix Inversion Architectures. In Proceedings of the Symposium on Application Specific Processors, Anaheim, CA, USA, 8–9 June 2008; pp. 42–47.
10. Benesty, J. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *J. Acoust. Soc. Am.* **2001**, *107*, 384–391. [[CrossRef](#)]
11. Warp, R.J.; Godfrey, D.; Dobbins, J.T. Applications of matrix inversion tomosynthesis. *Med. Imaging* **2000**, 3977. [[CrossRef](#)]
12. Godfrey, D.; McAdams, H.P.; Dobbins, J.T. The effect of averaging adjacent planes for artifact reduction in matrix inversion tomosynthesis. *Med Phys.* **2013**, *40*, 021907. [[CrossRef](#)]
13. Zhang, Y.; Ge, S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* **2005**, *16*, 1477–1490. [[CrossRef](#)]
14. Guo, D.; Zhang, Y. Zhang neural network, Getz–Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots’ kinematic control. *Neurocomputing* **2012**, *97*, 22–32. [[CrossRef](#)]
15. Guo, D.; Li, K.; Yan, L.; Nie, Z.; Jin, F. The application of Li-function activated RNN to acceleration-level robots’ kinematic control via time-varying matrix inversion. In Proceedings of the Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 3455–3460.
16. Amato, F.; Moscato, V.; Picariello, A.; Sperli, G. Recommendation in Social Media Networks. In Proceedings of the IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19–21 April 2017; pp. 213–216.
17. Amato, F.; Castiglione, A.; Moscato, V.; Picariello, A.; Sperli, G. Multimedia summarization using social media content. *Multimed. Tools Appl.* **2018**, *77*. [[CrossRef](#)]
18. Hopf, B.; Dutz, F.; Bosselmann, T.; Willsch, M.; Koch, A.; Roths, J. Iterative matrix algorithm for high precision temperature and force decoupling in multi-parameter FBG sensing. *Opt. Express* **2018**, *26*, 12092–12105. [[CrossRef](#)] [[PubMed](#)]

19. Cheng, Y.; Tsai, P.; Huang, M. Matrix-Inversion-Free Compressed Sensing with Variable Orthogonal Multi-Matching Pursuit Based on Prior Information for ECG Signals. *IEEE Trans. Biomed. Circuits Syst.* **2016**, *10*, 864–873. [[CrossRef](#)] [[PubMed](#)]
20. Ji, S.; Dunson, D.; Carin, L. Multitask Compressive Sensing. *IEEE Trans. Signal Process.* **2009**, *57*, 92–106. [[CrossRef](#)]
21. Bicchi, A.; Canepa, G. Optimal design of multivariate sensors. *Meas. Sci. Technol.* **1994**, *5*, 319–332. [[CrossRef](#)]
22. Mach, D.; Koshak, W.J. General matrix inversion technique for the calibration of electric field sensor arrays on aircraft platforms. *J. Atmos. Ocean. Technol.* **2007**, *24*, 1576–1587. [[CrossRef](#)]
23. Liu, S.; Chepuri, S.P.; Fardad, M.; Maşazade, E.; Leus, G.; Varshney, P.K. Sensor Selection for Estimation with Correlated Measurement Noise. *IEEE Trans. Signal Process.* **2016**, *64*, 3509–3522. [[CrossRef](#)]
24. Zhang, Y.; Chen, K.; Tan, H. Performance Analysis of Gradient Neural Network Exploited for Online Time-Varying Matrix Inversion. *IEEE Trans. Autom. Control* **2009**, *54*, 1940–1945. [[CrossRef](#)]
25. Zhang, Y.; Ma, W.; Cai, B. From Zhang Neural Network to Newton Iteration for Matrix Inversion. *IEEE Trans. Circuits Syst.* **2009**, *56*, 1405–1415. [[CrossRef](#)]
26. Kandasamy, W.; Smarandache, F. *Exploring the Extension of Natural Operations on Intervals, Matrices and Complex Numbers*; ZIP Publishing: Columbus, OH, USA, 2012.
27. Chen, Y.; Yi, C.; Qiao, D. Improved neural solution for the Lyapunov matrix equation based on gradient search. *Inf. Process. Lett.* **2013**, *113*, 876–881. [[CrossRef](#)]
28. Yi, C.; Chen, Y.; Lu, Z. Improved gradient-based neural networks for online solution of Lyapunov matrix equation. *Inf. Process. Lett.* **2011**, *111*, 780–786. [[CrossRef](#)]
29. Wilkinson, J. Error Analysis of Direct Methods of Matrix Inversion. *JACM* **1961**, *8*, 281–330. [[CrossRef](#)]
30. Chua, L.; Yang, L. Cellular Neural Networks: Theory. *IEEE Trans. Circuits Syst.* **1988**, *35*, 1257–1272. [[CrossRef](#)]
31. Roska, T.; Chua, L. The CNN universal machine: An analogic array computer. *IEEE Trans. Circuits Syst.* **1993**, *40*, 163–173. [[CrossRef](#)]
32. Endisch, C.; Stolze, P.; Hackl, C.M.; Schröder, D. Comments on Backpropagation Algorithms for a Broad Class of Dynamic Networks. *IEEE Trans. Neural Netw.* **2009**, *20*, 540–541. [[CrossRef](#)]
33. Potluri, S.; Fasih, A.; Kishore, L.; Machot, F.A.; Kyamakya, K. CNN Based High Performance Computing for Real Time Image Processing on GPU. In Proceedings of the Nonlinear Dynamics and Synchronization (INDS) & 16th Int'l Symposium on Theoretical Electrical Engineering (ISTET), Klagenfurt, Austria, 25–27 July 2011; pp. 1–7.
34. Mainzer, K. CNN and the Evolution of Complex Information Systems in Nature and Technology. In Proceedings of the 7th IEEE Cellular Neural Networks and Their Applications, Frankfurt, Germany, 24 July 2002; pp. 480–485.
35. Chedjou, J.; Kyamakya, K.; Khan, U.; Latif, M. Potential Contribution of CNN-based Solving of Stiff ODEs & PDEs to Enabling Real-Time Computational Engineering. In Proceedings of the 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010), Berkeley, CA, USA, 3–5 February 2010; pp. 1–6.
36. Chedjou, J.; Kyamakya, K. A Universal Concept Based on Cellular Neural Networks for Ultrafast and Flexible Solving of Differential Equations. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 749–762. [[CrossRef](#)]
37. Stanimirovic, P. Recurrent Neural Network for Computing the Drazin Inverse. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2830–2843. [[CrossRef](#)]
38. Wang, J. Recurrent Neural Networks for Computing Pseudoinverses of Rank-Deficient Matrices. *Siam J. Sci. Comput.* **1997**, *5*, 1479–1493. [[CrossRef](#)]
39. Chen, K. Recurrent Implicit Dynamics for Online Matrix Inversion. *Appl. Math. Comput.* **2013**, *219*, 10218–10224. [[CrossRef](#)]
40. Feng, F.; Zhang, Q.; Liu, H. A Recurrent Neural Network for Extreme Eigenvalue Problem. In Proceedings of the ICIC 2005, Fuzhou, China, 20–23 August 2015; pp. 787–796.
41. Tang, Y.; Li, J. Another neural network based approach for commuting eigenvalues and eigenvectors of real skew-symmetric matrices. *Comput. Math. Appl.* **2010**, *60*, 1385–1392. [[CrossRef](#)]
42. Xu, L.; King, I. A PCA approach for fast retrieval of structural patterns in attributed graphs. *IEEE Trans. Syst. Man Cybern.* **2001**, *31*, 812–817.

43. Liu, Y.; You, Z.; Cao, L. A recurrent neural network computing the largest imaginary or real part of eigenvalues of real matrices. *Comput. Math. Appl.* **2007**, *53*, 41–53. [[CrossRef](#)]
44. Bouzerdorm, A.; Pattison, T. Neural Network for Quadratic Optimization with Bound Constraints. *IEEE Trans. Neural Netw.* **1993**, *4*, 293–304. [[CrossRef](#)]
45. Zhang, Y. Towards piecewise-linear primal neural networks for optimization and redundant robotics. In Proceedings of the IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 23–25 April 2006.
46. Hopfield, J.J.; Tank, D. Neural Computation of Decisions in Optimization Problems. *Cybernetics* **1984**, *52*, 141–152.
47. Kennedy, M.P. Neural Networks for Nonlinear Programming. *IEEE Trans Circuits Syst.* **1988**, *35*, 554–562. [[CrossRef](#)]
48. He, Z.; Gao, S.; Xiao, L.; Liu, D.; He, H.; Barber, D. Wider and deeper, cheaper and faster: Tensorized LSTMs for sequence learning. In Proceedings of the Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 1–11.
49. Jang, J.-S.; Lee, S.-Y.; Shin, S.-Y. An Optimization Network for Matrix Inversion. In Proceedings of the 1987 IEEE Conference on Neural Information Processing Systems, Denver, CO, USA, 8–12 November 1987.
50. Zhang, Y.; Li, Z.; Li, K. Complex-valued Zhang neural network for online complex-valued time-varying matrix inversion. *Appl. Math. Comput.* **2011**, *217*, 10066–10073. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Artificial Neural Networks for Forecasting Passenger Flows on Metro Lines

Mariano Gallo ^{1,*}, Giuseppina De Luca ¹, Luca D'Acerno ² and Marilisa Botte ²¹ Department of Engineering, University of Sannio, piazza Roma 21, 82100 Benevento, Italy² Department of Civil, Architectural and Environmental Engineering, Federico II University of Naples, via Claudio 21, 80125 Naples, Italy

* Correspondence: gallo@unisannio.it; Tel.: +39-0824-305565

Received: 1 July 2019; Accepted: 1 August 2019; Published: 5 August 2019

Abstract: Forecasting user flows on transportation networks is a fundamental task for Intelligent Transport Systems (ITSs). Indeed, most control and management strategies on transportation systems are based on the knowledge of user flows. For implementing ITS strategies, the forecast of user flows on some network links obtained as a function of user flows on other links (for instance, where data are available in real time with sensors) may provide a significant contribution. In this paper, we propose the use of Artificial Neural Networks (ANNs) for forecasting metro onboard passenger flows as a function of passenger counts at station turnstiles. We assume that metro station turnstiles record the number of passengers entering by means of an automatic counting system and that these data are available every few minutes (temporal aggregation); the objective is to estimate onboard passengers on each track section of the line (i.e., between two successive stations) as a function of turnstile data collected in the previous periods. The choice of the period length may depend on service schedules. Artificial Neural Networks are trained by using simulation data obtained with a dynamic loading procedure of the rail line. The proposed approach is tested on a real-scale case: Line 1 of the Naples metro system (Italy). Numerical results show that the proposed approach is able to forecast the flows on metro sections with satisfactory precision.

Keywords: artificial neural networks; metro; transportation; user flow forecast

1. Introduction

Knowledge of user flows on transportation systems is crucial for implementing control and management policies. In this context, monitoring systems assume a central role and are widely used in many road and rail networks: they are one of the most important (and necessary) components of Intelligent Transport Systems (ITSs).

Monitoring systems are based on sensors (or detectors) that measure some characteristics of flows and transmit them to a control room, where the relevant data are used for implementing control and management strategies. For instance, knowing traffic flows on a road network is useful for implementing flow-responsive traffic-signal systems, while the user loads on public transport vehicles can be used for real-time scheduling/rescheduling tasks.

Despite their usefulness, monitoring systems are not always provided on road and rail networks or, sometimes, implemented systems do not have enough sensors to collect the data required for control and management strategies. Indeed, the costs of these systems require significant investments from public administration or public transport firms which are often inadmissible.

In this paper, we focus on metro lines, where access is controlled by turnstiles which can count the passengers entering each station with or without identifying their direction (in most of the existing stations, except terminals, passengers can board trains in different directions). These data can be easily collected every few minutes (e.g., fifteen-minute intervals) without installing new sensors. The objective

of this paper is to propose a method based on Artificial Neural Networks (ANNs) for estimating the number of passengers on each segment of a metro line using data obtained from turnstiles. Indeed, on-board data are rarely available in real time because they require infrared scanners, or alternatively weight-based, image-based or photocell-based sensors that are seldom installed on coaches.

The paper is organised as follows: Section 2 provides the background; Section 3 describes the problem to solve and the ANN approach; the method adopted for generating the data used in numerical tests is reported in Section 4; Section 5 describes the case study and the results; Section 6 concludes and identifies some research prospects.

2. Background

2.1. Artificial Neural Networks

The Artificial Neural Network (ANN) is a mathematical method that is widely used for reproducing several physical phenomena and forecasting the results of some actions on (or variations of) the parameters/variables of the system. ANNs are considered to be black-boxes since the functions and the relationships between inputs and outputs are hidden, not known and, generally, not interpretable.

Both the strengths and weaknesses of ANNs are related to their black-box approach. ANNs can reproduce a phenomenon or approximate a function without making the parameters explicit; moreover, once trained, they are able to give the results rapidly. On the other hand, trained ANNs are not extendible even to similar cases and work only if the boundary conditions do not change significantly.

ANNs have been widely studied elsewhere; they were initially introduced in [1–3] and then developed in other pioneering contributions [4–8]. Many general books focus on ANNs; here we refer to [9–14].

Literature reviews have been proposed in several papers. Scarselli and Tsoi [15] presented a review of studies that used Feedforward Neural Networks to approximate some functions, examining computational aspects, structures of the network (hidden layers and neurons), and training algorithms. They also proposed two training algorithms. Baptista and Morgado-Dias [16] examined the numerous software tools available, with the intention to help choose the most appropriate tool while considering its features (operating system, minimum hardware, kind of licence, algorithms implemented, and so on). Timotheou [17] reviewed random neural networks and their application to several problems. Extreme learning machines were reviewed in [18], while reviews on deep learning in neural networks can be found in [19,20]. Finally, here we refer to Yao [21] for an appraisal of evolutionary artificial neural networks.

2.2. Road Traffic Flow Forecasting

Two main types of transportation flow forecasting problems can be identified: (i) short-term forecasting and (ii) traffic data spatial extension. Some papers that applied ANNs to these problems were reviewed in [22].

The first problem aims to forecast the traffic flows (or user flows) that use a road section (or a transit line) in a future time interval, using the data measured in the previous time intervals in the same road section. This problem has been widely studied elsewhere, and a complete review would deserve a specific paper; here we refer to [23,24]. For solving this problem, several methods were proposed; in this paper, we focus on most of the literature that has used ANNs.

Kirby et al. [25] discussed the use of ANNs for forecasting traffic flows on motorways up to an hour ahead and compared this approach with other statistical models. Smith and Demetsky [26] compared the performances of ANNs with traditional methods for solving the short-term traffic flow prediction problem, such as data-based algorithms and time-series models; they found that the back-propagation neural network model was able to predict future traffic flows on highways better than the other models. In the same research field, ANNs were used for modelling freeway traffic in a macroscopic environment [27]; the authors found that the neural network model was able to

capture the traffic dynamics quite closely and was “computationally efficient for real-time implementation”. ANNs were proposed as tools for predicting congestion and forecasting flows in [28]; the authors also discussed whether ANNs were able to estimate parameters that cannot be directly measured with road sensors. Park et al. [29] proposed a radial basis function neural network for short-term forecasting on freeways; they tested the method with real observations and compared it with other approaches such as Taylor series, single and double exponential smoothing methods, and back-propagation neural networks. Zheng et al. [30] proposed a Bayesian combined neural network approach for short-term forecasting on freeways, while a binary neural network was presented in [31]. Another application on a highway can be found in [32], while applications in urban environments can be found in [33,34]. Park et al. [35] used feedforward multilayer neural networks for estimating link travel times on freeways. Other applications of ANNs to short-term forecasting can be found in [36–40]. Ledoux [41] proposed the use of ANNs within an urban traffic flow model, while Florio and Mussone [42] studied traffic flow stability on freeways with neural network models.

Traffic data spatial extension problems have received less attention in the literature. Lin et al. [43] used a macroscopic model for short-term forecasting, which is also able to predict flows on other links. Zheng Zhu et al. [44] used ANNs for spatial extension of traffic flows at road intersections. Gallo and De Luca [45] proposed the use of ANNs for estimating traffic flows on some links of an urban road network according to the flows measured on other links.

Recently, deep learning methods have been proposed in the field of traffic prediction; a survey can be found in [46]. The authors identified four main deep learning models: deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and deep reinforcement learning (referring, in particular, to the Deep Q-Network [47]). In the ITS field, among others, DNN, CNN and RNN are useful for time series prediction. However, the above methods have not yet been used for the spatial extension of traffic or passenger data.

Short-term traffic forecasting with deep learning was studied in [48], where a long short-term memory (LSTM) network was proposed; the method was tested on a case study in Beijing, showing promising forecast accuracy compared with other approaches. The same approach has also been used for traffic flow prediction with missing data in [49].

Temporal CNN was proposed in [50] for short-term forecasting of passenger demand, outperforming other models in test cases. Support vector machines and data denoising schemes were combined in [51] for traffic flow prediction; the proposed denoising algorithms improved the results in this hybrid model, compared to other approaches without a denoising strategy. Short-term travel speed prediction was studied in [52–55].

2.3. Metro Passenger Flow Forecasting

The specific problem tackled in this paper entails metro passenger flow prediction. The literature review in this field presents some interesting contributions.

Deep learning methods were proposed in [56] and tested on a Bus Rapid Transit (BRT) system. The proposed model forecasts the hourly flow, adopting a three-stage deep learning architecture. This paper also analyses the literature, identifying four different approaches: (1) traditional classical algorithms; (2) regressive models; (3) machine learning-based models, including ANNs; (4) hybrid models. All studied cases, however, refer to short-term or long-term time periods, without considering the spatial extension. Among them, cases reported in [57,58] are applied on railways and based on ANNs, focusing on short-term and long-term forecasting respectively. Short-term forecasting on urban metros was also studied along with other methods, such as Kalman filter [59] and ARIMA (autoregressive integrated moving average) models [60].

Li et al. [61] proposed a multiscale radial basis function (MSRBF) for forecasting short-term metro passenger flows on special occasions, such as sporting events, concerts, and so on. In this case, passenger flow is very irregular and predictions are more difficult to obtain. Ling et al. [62] used smart-card data for predicting passenger flows in the subway of Shenzhen (China); they analysed

four predictive models: a historical average model, ANN, regression model and a gradient-boosted regression tree model. Liu et al. [63] proposed a deep learning method for short-term forecasting of metro inbound/outbound passenger flows, while Wang et al. [64] proposed a Novel Markov-Grey model for solving the same problem.

2.4. Contribution of the Paper

ANNs have been widely used in numerous scientific fields since the 1950s/1960s and in traffic engineering since the 1990s. Most applications in traffic engineering have focused on the temporal extent of data (more frequently short-term or, sometimes, long-term predictions) and road environments; fewer cases refer to transit systems. The spatial extent of data has been less widely studied and, to our best knowledge, the use of ANNs for the specific problem tackled in this paper has not been proposed elsewhere. Therefore, the originality of our contribution does not so much concern the method used, which is indeed consolidated, as the problem dealt with and the procedure used to construct the training datasets. Other more advanced methods, such as deep learning, will be the subject of further research, as will be discussed in the conclusions. In this paper, the performance of ANNs was not compared with other methods because there are no benchmarks. Indeed, almost all methods usually used as benchmarks in short-term forecasting problems are not applicable in our case, since they are time-series specific.

It is important to underline that the problem studied is relevant to the real-time management of metro lines. Data at turnstiles can be easily collected with methods that do not require significant additional investment, while data obtained through the above procedure (loads on line sections) are essential for service operators and, unlike the former, are not easily detectable in real time and continuously.

3. Problem Description and ANN Approach

We assume that turnstiles control all accesses to a metro line: each user, entering a station, uses a ticket (or a pass) for crossing the turnstile. Moreover, the turnstiles are only able to count users entering the station without linking the origin of each trip with the corresponding destination. This situation is common to many metro lines, such as Line 1 of the Naples metro system (Italy) which will be the subject of the real-scale test. Indeed, turnstiles are often installed only for facilitating ticket control/validation and avoiding no-ticket trips, and, in urban contexts, the fare is the same regardless of the origin-destination pair. Below, we consider two cases: (a) turnstiles at the station entrance that measure only passengers entering, with no indication which direction they will follow; (b) turnstiles upon access to platforms that also give information on trip direction (see Figure 1).

The data collected by turnstiles can be used, with low technological investment, for implementing a monitoring system of the whole metro line, generating information about the passengers on each railway section (between two stations). Such information can be of great use to metro operators for implementing real-time strategies, like a frequency increase or reduction, determination of train composition (number of passenger carriages), the scheduling of additional runs, and so on.

The problem to solve is the estimation of loads on the line starting from turnstile data. For this purpose, we propose feedforward ANNs, which are suitable because (a) the relationships between inputs and outputs do not need to be explicitly known, (b) the results are obtained rapidly, and (c) the boundary conditions do not usually change so much as to invalidate forecasts.

The structure of the ANN provides an input layer with a node for each turnstile, an output layer with a node for each convoy load, and one or more hidden layers. The best structure of the ANN has to be designed for each specific problem. A crucial point concerns the dynamic nature of the problem: the train moves along the line, loading and unloading passengers at stations in different time intervals. Therefore, the number of onboard passengers between two stations at time t depends on the passengers loaded and unloaded at previous stations at different times ($<t$). Hence, ANN inputs

have to consider turnstile counts referring to several time intervals preceding those being forecast. The number of inputs will depend on the travel time duration between terminals.

The other crucial point is the training phase of ANNs; here we propose to use a supervised learning method, where the example datasets are generated through dynamic simulation models (see Section 4). Note that it is not possible to use real-world data for the training phase. Indeed, only the input data (on passengers entering the stations) are available while the output data (on-board passengers) can be known only if all coaches have sensors that are able to measure them; in this case, however, the proposed approach would not be useful.

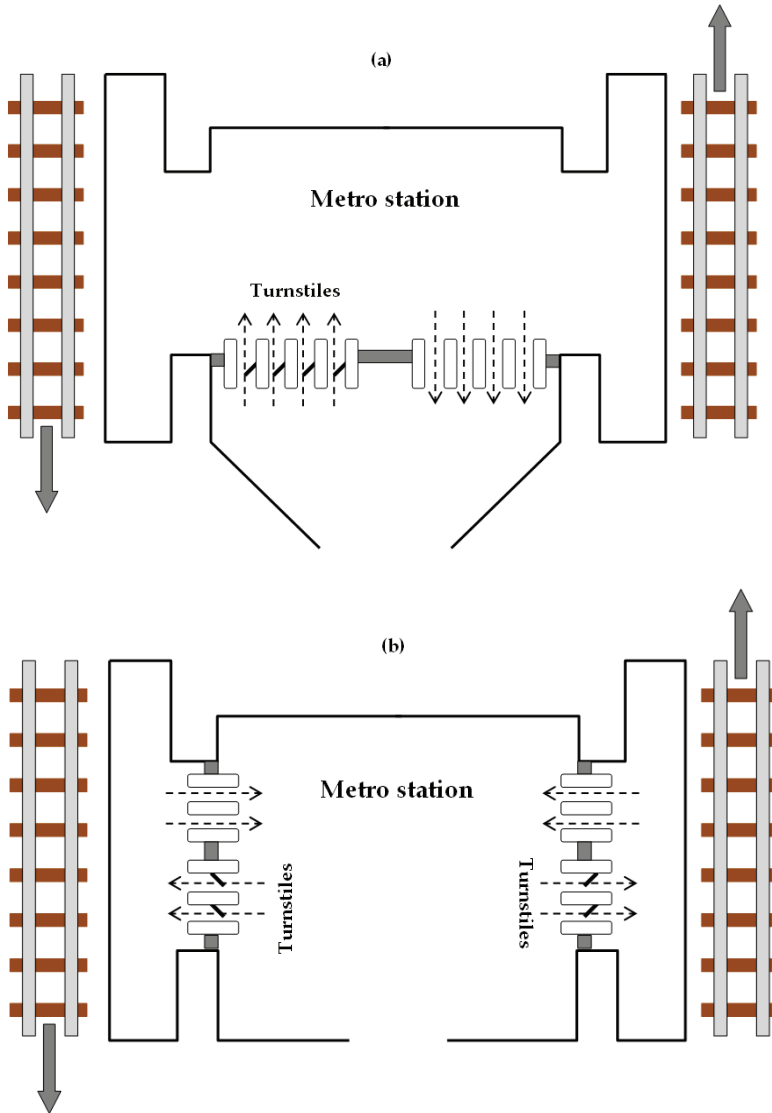


Figure 1. Types of metro stations: (a) turnstiles at the entrance; (b) turnstiles at platform accesses.

4. Generation of Training Datasets

To generate the training datasets, we used the simulation model proposed in [65]. This model assumes that:

1. platforms can accommodate all passengers (incoming, waiting and outgoing);
2. at each station and for each direction there is only one platform available;
3. the dwell time is constant and independent of the number of passengers alighting and boarding;
4. there is no interaction on the platform between alighting, boarding and waiting passengers;
5. the capacity of each train is fixed;
6. passengers are distributed uniformly among the train coaches;
7. there is no interaction in the train between alighting, boarding and onboard passengers.

In our test, we assume that the passengers follow a FIFO (First In-First Out) rule for boarding the convoy. The analytical details of the model can be found in [65].

Using this model, we generate the training datasets on the case study as follows: (a) numerous origin-destination (OD) matrices referring to 15' intervals are randomly generated starting from a base OD matrix; (b) four OD matrices, referring to four consecutive time intervals, are assigned to the metro line, yielding as results the passengers counted at turnstiles, for each interval, and the passengers onboard in each railway section in the next time interval; (c) the output data of the problem are the passengers on railway sections, while the input data are the passengers counted at turnstiles in four time intervals, corresponding to the four previous 15' periods (e.g., flows on railway sections between 10:15 and 10:30 are estimated according to the turnstile counts in the intervals: 9:15–9:30, 9:30–9:45, 9:45–10:00 and 10:00–10:15). Therefore, the structure of the training datasets is reported in Table 1, where the following notations are used:

- ds is the number of datasets;
- t is the period under analysis;
- $c^i_{j,t}$ is the passenger count at turnstile j in period t for dataset i ;
- $f^i_{k,t}$ is the load on railway section k in period t for dataset i .

Table 1. Structure of training datasets for period t .

| Dataset → | 1 | 2 | ... | ds |
|----------------------------------|----------------|----------------|-----|-------------------|
| <i>Input data</i> | | | | |
| Turnstile 1–Period $t - 1$ | $c^1_{1,t-1}$ | $c^2_{1,t-1}$ | ... | $c^{ds}_{1,t-1}$ |
| Turnstile 2–Period $t - 1$ | $c^1_{2,t-1}$ | $c^2_{2,t-1}$ | ... | $c^{ds}_{2,t-1}$ |
| ... | ... | ... | ... | ... |
| Turnstile ts –Period $t - 1$ | $c^1_{ts,t-1}$ | $c^2_{ts,t-1}$ | ... | $c^{ds}_{ts,t-1}$ |
| Turnstile 1–Period $t - 2$ | $c^1_{1,t-2}$ | $c^2_{1,t-2}$ | ... | $c^{ds}_{1,t-2}$ |
| Turnstile 2–Period $t - 2$ | $c^1_{2,t-2}$ | $c^2_{2,t-2}$ | ... | $c^{ds}_{2,t-2}$ |
| ... | ... | ... | ... | ... |
| Turnstile ts –Period $t - 2$ | $c^1_{ts,t-2}$ | $c^2_{ts,t-2}$ | ... | $c^{ds}_{ts,t-2}$ |
| Turnstile 1–Period $t - 3$ | $c^1_{1,t-3}$ | $c^2_{1,t-3}$ | ... | $c^{ds}_{1,t-3}$ |
| Turnstile 2–Period $t - 3$ | $c^1_{2,t-3}$ | $c^2_{2,t-3}$ | ... | $c^{ds}_{2,t-3}$ |
| ... | ... | ... | ... | ... |
| Turnstile ts –Period $t - 3$ | $c^1_{ts,t-3}$ | $c^2_{ts,t-3}$ | ... | $c^{ds}_{ts,t-3}$ |
| Turnstile 1–Period $t - 4$ | $c^1_{1,t-4}$ | $c^2_{1,t-4}$ | ... | $c^{ds}_{1,t-4}$ |
| Turnstile 2–Period $t - 4$ | $c^1_{2,t-4}$ | $c^2_{2,t-4}$ | ... | $c^{ds}_{2,t-4}$ |
| ... | ... | ... | ... | ... |
| Turnstile ts –Period $t - 4$ | $c^1_{ts,t-4}$ | $c^2_{ts,t-4}$ | ... | $c^{ds}_{ts,t-4}$ |
| <i>Output data</i> | | | | |
| Railway section 1–Period t | $f^1_{1,t}$ | $f^2_{1,t}$ | ... | $f^{ds}_{1,t}$ |
| Railway section 2–Period t | $f^1_{2,t}$ | $f^2_{2,t}$ | ... | $f^{ds}_{2,t}$ |
| ... | ... | ... | ... | ... |
| Railway section rs –Period t | $f^1_{rs,t}$ | $f^2_{rs,t}$ | ... | $f^{ds}_{rs,t}$ |

5. Case Study and Numerical Results

The proposed approach was tested on Line 1 of the Naples metro system. This line (see Figure 2) is 18 km long and has 18 stations; it connects high-density districts in Naples and is crucial infrastructure for urban mobility.

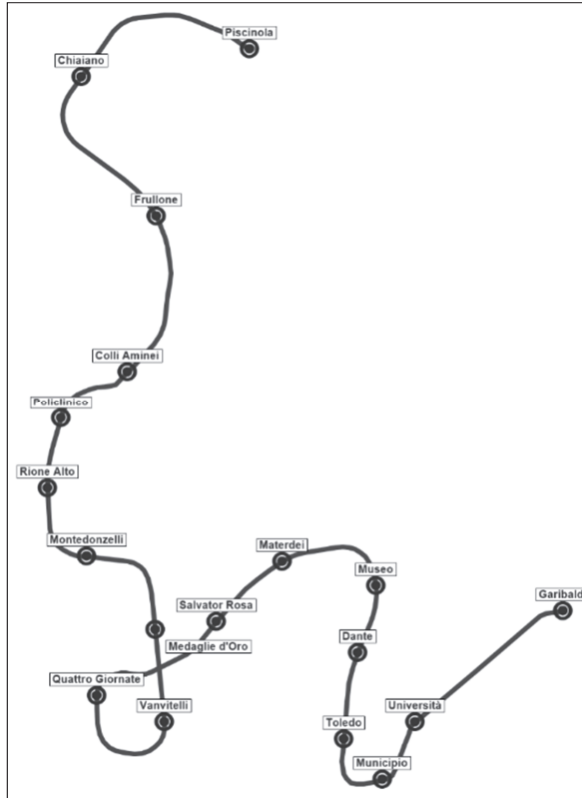


Figure 2. Line 1 route.

Considering these characteristics, we have 34 or 18 turnstiles, if we divide the passengers according to direction or not, and 34 mono-directional railway sections. The main features of the line are summarised in Table 2.

Table 2. Features of Line 1.

| | |
|---|-----------|
| Stations | 18 |
| Working day runs | 241 |
| Convoy capacity (pax/convoy) | 864 |
| Line length (km) (outward/return direction) | 18.8/18.6 |
| Headway (min) | 8–20 |

The training datasets were obtained by simulating 2500 OD matrices generated randomly. On eliminating some of them because their results were not feasible (too many passengers compared to the actual capacity), we generated 2279 training datasets. Of the latter datasets, 2229 were used for the whole training process of the ANNs with the software MatLab: 1561 training datasets (70%), 334 validation datasets (15%) and 334 testing datasets (15%). The remaining 50 datasets were used

to verify the goodness of the trained ANNs with examples that were not used before in the training process. We tested six ANN structures for both cases: (a) turnstiles at the station entrance (18 turnstiles), and (b) turnstiles at the access points to platforms (34 turnstiles). We thus trained and tested 12 ANNs, as reported in Table 3.

Table 3. ANN structures.

| Case | ANN | Input Nodes | Output Nodes | Hidden Layers | Neurons |
|------|--------|-------------|--------------|---------------|---------|
| a | a_1_6 | 72 | 34 | 1 | 6 |
| a | a_1_10 | 72 | 34 | 1 | 10 |
| a | a_1_20 | 72 | 34 | 1 | 20 |
| a | a_2_6 | 72 | 34 | 2 | 6/6 |
| a | a_2_10 | 72 | 34 | 2 | 10/10 |
| a | a_2_20 | 72 | 34 | 2 | 20/20 |
| b | b_1_6 | 136 | 34 | 1 | 6 |
| b | b_1_10 | 136 | 34 | 1 | 10 |
| b | b_1_20 | 136 | 34 | 1 | 20 |
| b | b_2_6 | 136 | 34 | 2 | 6/6 |
| b | b_2_10 | 136 | 34 | 2 | 10/10 |
| b | b_2_20 | 136 | 34 | 2 | 20/20 |

The training phase required computing times from 30 s (case a_1_6) to 8 min (case b_2_20), with a Personal Computer Hewlett Packard i7-7700HQ, 280 GHz, RAM 16 GB. In Table 4 we report the best and worst coefficients of determination (R^2) for each case, referring to the 50 datasets not used in the training phase, and the corresponding averages and variances. The datasets for which R^2 is lower than 0.9, 0.8, 0.7 and 0.6 are reported in Table 5. In these tables, the best values for each ANN are underlined.

Examining the results reported in Tables 4 and 5, we may identify as best ANN structures the one with one hidden layer and 20 neurons for case (a), and two hidden layers and 10 neurons for case (b). The corresponding dispersion diagrams in the cases of best and worst R^2 are reported in Figures 3 and 4.

Table 4. Coefficients of determination (R^2).

| ANN | Best | Worst | Average | Variance |
|--------|---------------|---------------|---------------|---------------|
| a_1_6 | <u>0.9946</u> | 0.5487 | 0.7984 | 0.0124 |
| a_1_10 | <u>0.9941</u> | 0.4990 | 0.8108 | 0.0160 |
| a_1_20 | 0.9931 | <u>0.5613</u> | <u>0.8332</u> | 0.0157 |
| a_2_6 | 0.9916 | <u>0.5353</u> | 0.7949 | <u>0.0121</u> |
| a_2_10 | 0.9930 | 0.4638 | 0.8136 | 0.0175 |
| a_2_20 | 0.9933 | 0.5342 | 0.8244 | 0.0162 |
| b_1_6 | 0.9875 | 0.5460 | 0.8016 | <u>0.0129</u> |
| b_1_10 | <u>0.9905</u> | 0.3505 | 0.8115 | 0.0202 |
| b_1_20 | <u>0.9882</u> | 0.4226 | <u>0.8291</u> | 0.0160 |
| b_2_6 | 0.9785 | <u>0.5119</u> | 0.7775 | 0.0132 |
| b_2_10 | 0.9889 | <u>0.4935</u> | 0.8221 | 0.0132 |
| b_2_20 | 0.9852 | 0.4492 | 0.8075 | 0.0245 |

Best values are underlined.

Table 5. Analysis of R² values.

| ANN | R ² < 0.9 | R ² < 0.8 | R ² < 0.7 | R ² < 0.6 |
|------------------------|----------------------|----------------------|----------------------|----------------------|
| Number of datasets | | | | |
| a_1_6 | 41 | 25 | 9 | 4 |
| a_1_10 | 36 | 22 | <u>7</u> | 4 |
| a_1_20 | <u>33</u> | <u>18</u> | <u>7</u> | <u>3</u> |
| a_2_6 | 43 | 25 | 10 | <u>3</u> |
| a_2_10 | 35 | 21 | 8 | 4 |
| a_2_20 | 34 | 19 | 9 | 5 |
| Percentage of datasets | | | | |
| a_1_6 | 82% | 50% | 18% | 8% |
| a_1_10 | 72% | 44% | <u>14%</u> | 8% |
| a_1_20 | <u>66%</u> | <u>36%</u> | <u>14%</u> | <u>6%</u> |
| a_2_6 | 86% | 50% | 20% | <u>6%</u> |
| a_2_10 | 70% | 42% | 16% | 8% |
| a_2_20 | 68% | 38% | 18% | 10% |
| b_1_6 | 74% | 48% | 20% | <u>4%</u> |
| b_1_10 | 68% | 46% | 16% | <u>10%</u> |
| b_1_20 | 64% | <u>38%</u> | 16% | 6% |
| b_2_6 | 86% | <u>50%</u> | 20% | 12% |
| b_2_10 | 70% | 40% | <u>14%</u> | <u>4%</u> |
| b_2_20 | <u>60%</u> | 42% | <u>18%</u> | <u>14%</u> |

Best values are underlined.

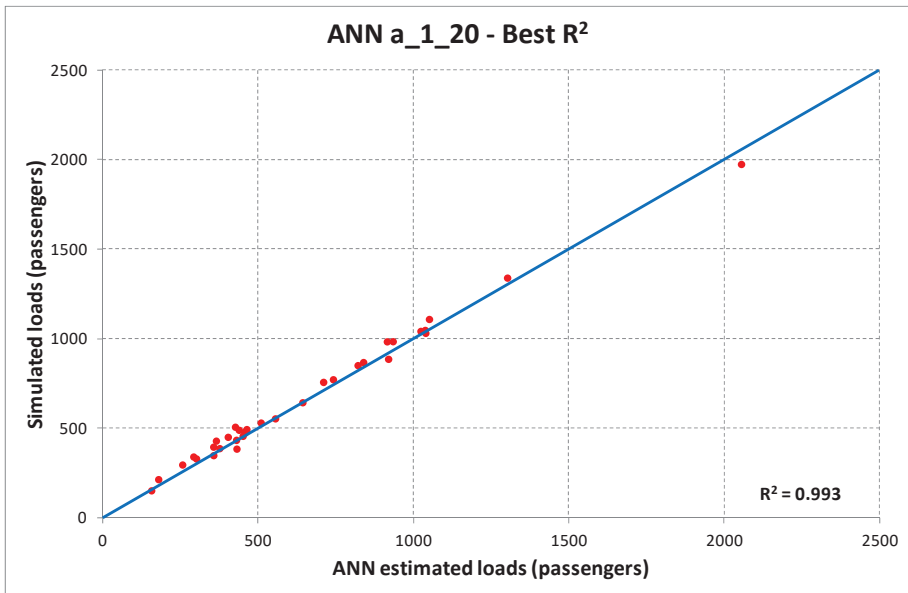


Figure 3. Cont.

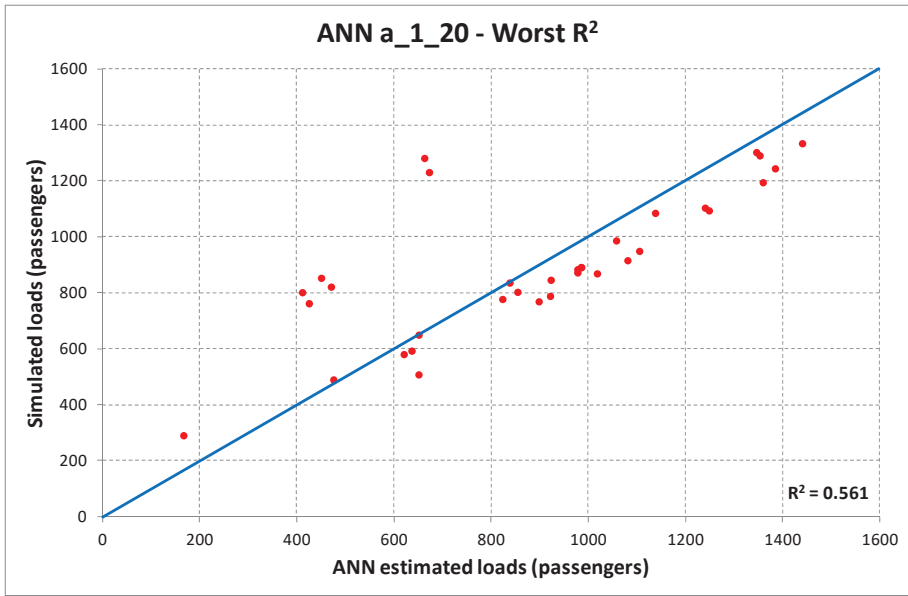


Figure 3. Dispersion diagrams ANN a_1_20 (best, upper diagram; worst, lower diagram).

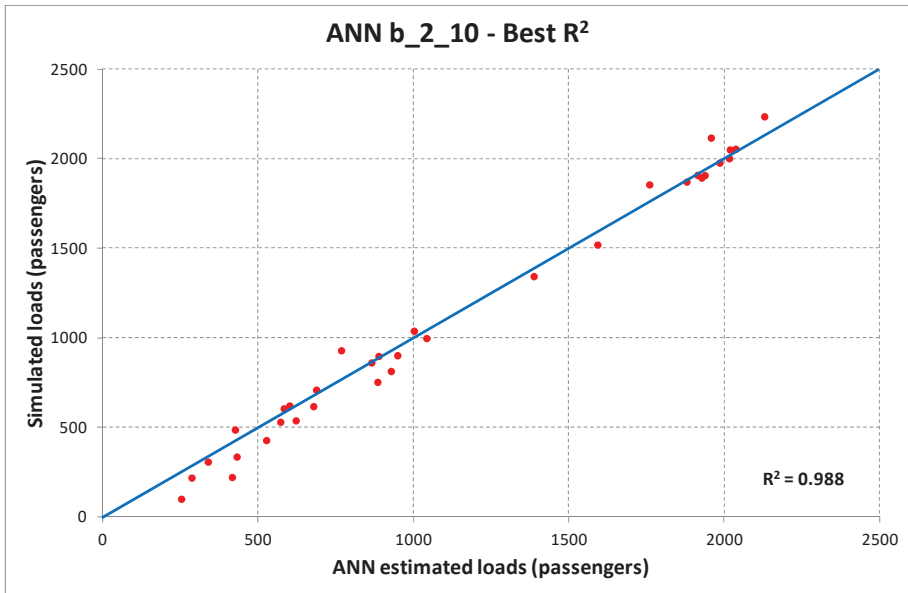


Figure 4. Cont.

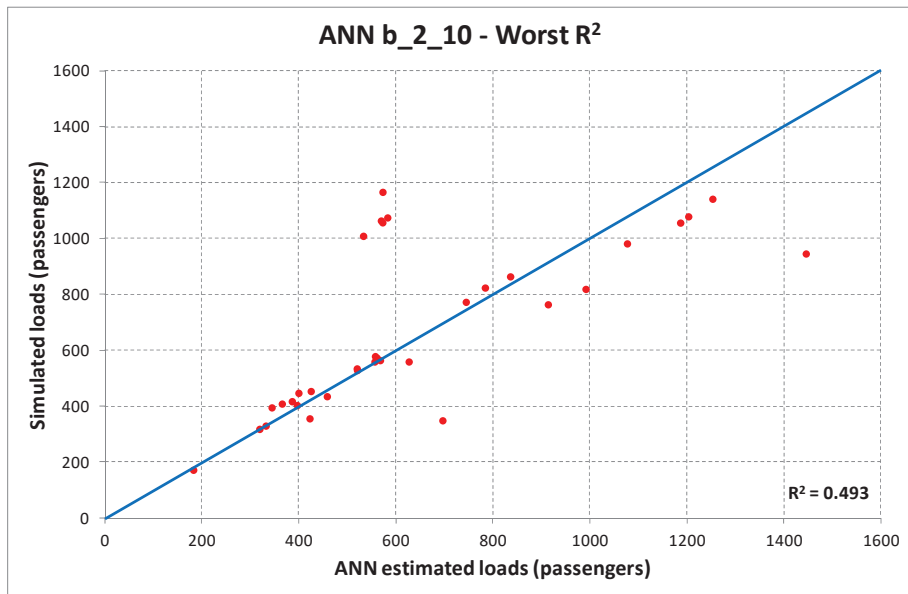


Figure 4. Dispersion diagrams ANN b_2_10 (best, upper diagram; worst, lower diagram).

6. Conclusions and Research Prospects

In this paper, we studied the problem of forecasting passenger flows on railway sections of a metro line starting from counts at turnstiles and proposed to use artificial neural networks (ANNs) for its solution. The training datasets were generated using a simulation model. We considered two cases: turnstiles at station entrances and turnstiles at platform accesses. For both, we designed and trained several ANNs.

The results showed a good capacity of ANNs to forecast the loads on railway sections. Our analysis allowed us to identify the best ANN structure for each case.

Future research could profitably lead in several directions. First of all, other ANN structures could be tested. Then the problem could be extended to more complex metro systems, including systems with more than one line. Finally, other methods could be investigated, as well deep-learning approaches that could be applied to this problem.

Author Contributions: Conceptualization, M.G. and L.D.; Methodology, M.G., G.D.L. and L.D.; Validation, M.B. and G.D.L.; Investigation, M.B. and G.D.L.; Resources, L.D. and M.B.; Data Curation, M.B. and G.D.L.; Writing—Original Draft Preparation, M.G. and L.D.; Writing—Review and Editing, M.G., G.D.L., L.D. and M.B.; Supervision, M.G.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to the anonymous reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

- McCulloch, W.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
- Hebb, D.O. *The Organization of Behaviour. A Neuropsychological Theory*; Wiley: Hoboken, NJ, USA, 1949.
- Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*; Spartan Books: Washington, DC, USA, 1962.

4. Minsky, M.; Papert, S. *An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 1969.
5. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69. [[CrossRef](#)]
6. Grossberg, S. *Neural Networks and Natural Intelligence*; MIT Press: Cambridge, MA, USA, 1988.
7. Minsky, M.L. Theory of Neural—Analog Reinforcement System and Its Application to the Brain—Model Problem. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1954.
8. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [[CrossRef](#)]
9. Judd, J.S. *Neural Network Modeling and Connectionism. Neural Network Design and the Complexity of Learning*; MIT Press: Cambridge, MA, USA, 1990.
10. Haykin, S. *Neural Networks: A Comprehensive Foundation*; McMaster University: Hamilton, ON, Canada, 1994.
11. Miller, W.T.; Werbos, P.J.; Sutton, R.S. *Neural Networks for Control*; MIT Press: Cambridge, MA, USA, 1995.
12. Rojas, R. *Neural Networks. A Systematic Introduction*; Springer: Berlin/Heidelberg, Germany, 1996.
13. Haykin, S.S. *Kalman Filtering and Neural Networks*; Wiley Online Library: Hoboken, NJ, USA, 2001.
14. Aggarwal, C.C. *Neural Networks and Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2018.
15. Scarselli, F.; Tsoi, A.C. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Netw.* **1998**, *11*, 15–37. [[CrossRef](#)]
16. Baptista, D.; Morgado-Dias, F. A survey of artificial neural network training tools. *Neural Comput. Appl.* **2013**, *23*, 609–615. [[CrossRef](#)]
17. Timotheou, S. The random neural network: A survey. *Comput. J.* **2010**, *53*, 251–267. [[CrossRef](#)]
18. Huang, G.; Huang, G.-B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *61*, 32–48. [[CrossRef](#)] [[PubMed](#)]
19. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
20. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [[CrossRef](#)]
21. Yao, X. A review of evolutionary artificial neural networks. *Int. J. Intell. Syst.* **1993**, *8*, 539–567. [[CrossRef](#)]
22. De Luca, G.; Gallo, M. Artificial Neural Networks for forecasting user flows in transportation networks: Literature review, limits, potentialities and open challenges. In Proceedings of the 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, Naples, Italy, 26–28 June 2017; pp. 919–923.
23. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we’re going. *Transp. Res. Part. C* **2014**, *43*, 3–19. [[CrossRef](#)]
24. Oh, S.; Byon, Y.J.; Jang, K.; Yeo, H. Short-term travel-time prediction on highway: A review of the data-driven approach. *Transp. Res. Rev.* **2015**, *35*, 4–32. [[CrossRef](#)]
25. Kirby, H.R.; Watson, S.M.; Dougherty, S. Should we use neural networks or statistical models for short-term motorway traffic forecasting? *Int. J. Forecast.* **1997**, *13*, 43–50. [[CrossRef](#)]
26. Smith, B.L.; Demetsky, M.J. Short-term traffic flow prediction: Neural network approach. *Transp. Res. Rec.* **1994**, *1453*, 98–104.
27. Zhang, H.; Ritchie, S.G.; Lo, Z.-P. Macroscopic modeling of freeway traffic using an artificial neural network. *Transp. Res. Rec.* **1997**, *1588*, 110–119. [[CrossRef](#)]
28. Dougherty, M.S.; Kirby, H.C. The use of neural networks to recognize and predict traffic congestion. *Traffic Eng. Control* **1998**, *346*, 311–314.
29. Park, B.; Carroll, J.M.; Urbank, T.I. Short-term freeway traffic forecasting using radial basis function neural network. *Transp. Res. Rec.* **1998**, *1651*, 39–46. [[CrossRef](#)]
30. Zheng, W.; Lee, D.-H.; Shi, Q. Short-term freeway traffic prediction: Bayesian combined neural network approach. *J. Transp. Eng.* **2006**, *132*, 114–121. [[CrossRef](#)]
31. Hodge, V.; Austin, J.; Krishnan, R.; Polak, J.; Jackson, T. *Short-Term Traffic Prediction Using a Binary Neural Network*; UTSG: Toronto, ON, Canada, 2011.
32. Kumar, K.; Parida, M.; Katiyar, V.K. Short term traffic flow prediction for a non urban highway using Artificial Neural Network. *Procedia Soc. Behav. Sci.* **2013**, *104*, 755–764. [[CrossRef](#)]
33. Zheng, F.; Van Zuylen, H. Urban link travel time estimation based on sparse probe vehicle data. *Transp. Res. Part. C* **2013**, *31*, 145–157. [[CrossRef](#)]

34. Csikos, A.; Viharos, Z.J.; Kisk, B.; Tettamanti, T.; Varga, I. Traffic speed prediction method for urban networks an ANN approach. In Proceedings of the Models and Technologies for Intelligent Transportation Systems, Budapest, Hungary, 3–5 June 2015.
35. Park, D.; Rilett, L.R. Forecasting Freeway Link Travel Times with a Multilayer Feedforward Neural Network. *Comput. Aided Civ. Infrastruct. Eng.* **1999**, *14*, 357–367. [[CrossRef](#)]
36. Yasdi, R. Prediction of road traffic using a neural network approach. *Neural Comput. Appl.* **1999**, *8*, 135–142. [[CrossRef](#)]
37. Li, R.; Lu, H. Combined neural network approach for short-term urban freeway traffic flow prediction. *Lect. Notes Comput. Sci.* **2009**, *5553*, 1017–1025.
38. Gao, Y.; Sun, S. Multi-link traffic flow forecasting using neural networks. In Proceedings of the 2010 Sixth International Conference on Natural Computation, Yantai, China, 10–12 August 2010; pp. 398–401.
39. Gao, J.; Leng, Z.; Qin, Y.; Ma, Z.; Liu, X. Short-term traffic flow forecasting model based on wavelet neural network. In Proceedings of the 25th Chinese Control and Decision Conference, Guiyang, China, 25–27 May 2013; pp. 5081–5084.
40. Goves, C. Short term traffic prediction on the UK motorway network using neural networks. In Proceedings of the European Transport Conference, Frankfurt, Germany, 28–30 September 2015.
41. Ledoux, C. An urban traffic flow model integrating neural networks. *Transp. Res. Part. C* **1997**, *5*, 287–300. [[CrossRef](#)]
42. Florio, L.; Mussone, L. Neural network models for classification and forecasting of freeway traffic flow stability. In *Transportation Systems: Theory and Application of Advanced Technology*; Liu, B., Blosseville, J.M., Eds.; Elsevier: Amsterdam, The Netherlands, 1995; pp. 773–784.
43. Lin, S.; Xi, Y.; Yang, Y. Short-term traffic flow forecasting using macroscopic urban traffic network model. In Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems (ITSC), Beijing, China, 12–15 October 2008.
44. Zheng Zhu, J.; Xin Cao, J.; Zhu, Y. Traffic forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. *Transp. Res. Part. C* **2014**, *47*, 139–154.
45. Gallo, M.; De Luca, G. Spatial extension of road traffic sensor data with Artificial Neural Networks. *Sensors* **2018**, *18*, 14. [[CrossRef](#)] [[PubMed](#)]
46. Wang, Y.; Zhang, D.; Liu, Y.; Dai, B.; Lee, L.H. Enhancing transportation systems via deep learning: A survey. *Transp. Res. Part. C* **2019**, *99*, 144–163. [[CrossRef](#)]
47. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
48. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [[CrossRef](#)]
49. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **2018**, *318*, 297–305. [[CrossRef](#)]
50. Zhang, K.; Liu, Z.; Zheng, L. Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning. *IEEE Trans. Intell. Transp. Syst.* **2019**. [[CrossRef](#)]
51. Tang, J.; Chen, X.; Hu, Z.; Zong, F.; Han, C.; Li, J. Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Physica A* **2019**. [[CrossRef](#)]
52. Zheng, L.; Zhu, C.; Zhu, N.; He, T.; Dong, N.; Huang, H. Feature selection-based approach for urban short-term travel speed prediction. *IET Intell. Transp. Syst.* **2018**, *12*, 474–484. [[CrossRef](#)]
53. Wang, J.; Chen, R.; He, Z. Traffic speed prediction for urban transportation network: A path based deep learning approach. *Transp. Res. Part. C* **2019**, *100*, 372–385. [[CrossRef](#)]
54. Yu, D.; Liu, C.; Wu, Y.; Liao, S.; Anwar, T.; Li, W.; Zhou, C. Forecasting short-term traffic speed based on multiple attributes of adjacent roads. *Knowl. Based Syst.* **2019**, *163*, 472–484. [[CrossRef](#)]
55. Zhang, K.; Zheng, L.; Liu, Z.; Jia, N. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing* **2019**. [[CrossRef](#)]
56. Liu, L.; Chen, R.-C. A novel passenger flow prediction model using deep learning methods. *Transp. Res. Part. C* **2017**, *84*, 74–91. [[CrossRef](#)]
57. Tsai, T.H.; Lee, C.K.; Wei, C.H. Neural network based temporal feature models for short-term railway passenger demand forecasting. *Expert Syst. Appl.* **2009**, *36*, 3728–3736. [[CrossRef](#)]

58. Li, J.T.; Yang, J.F. Prediction of Dalian station passenger based on RBF neural network. *J. Dalian Jiaotong Univ.* **2007**, *28*, 32–34.
59. Jiao, P.; Li, R.; Sun, T.; Hou, Z.; Ibrahim, A. Three revised kalman filtering models for short-term rail transit passenger flow prediction. *Math. Probl. Eng.* **2016**, *2016*, 1–10. [[CrossRef](#)]
60. Cai, C.; Yao, E.; Wang, M.; Zhang, Y. Prediction of urban railway station's entrance and exit passenger flow based on multiply ARIMA model. *J. Beijing Jiaotong Univ.* **2014**, *38*, 135–140.
61. Li, Y.; Wang, X.; Sun, S.; Ma, X.; Lu, G. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transp. Res. Part. C* **2017**, *77*, 306–328. [[CrossRef](#)]
62. Ling, X.; Huang, Z.; Wang, C.; Zhang, F.; Wang, P. Predicting subway passenger flows under different traffic conditions. *PLoS ONE* **2018**, *13*, 1–23. [[CrossRef](#)]
63. Liu, Y.; Liu, Z.; Jia, R. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part. C* **2019**, *101*, 18–34. [[CrossRef](#)]
64. Wang, Y.; Ma, J.; Zhang, J. Metro Passenger Flow Forecast with a Novel Markov-Grey Model. *Period. Polytech. Transp. Eng.* **2019**. [[CrossRef](#)]
65. D'Acerno, L.; Botte, M.; Montella, B. Assumptions and simulation of passenger behaviour on rail platforms. *Int. J. Transp. Dev. Integr.* **2018**, *2*, 123–135. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Spatio-Temporal Synchronization of Cross Section Based Sensors for High Precision Microscopic Traffic Data Reconstruction

Adrian Fazekas * and Markus Oeser

Institute for Highway Engineering, RWTH Aachen University, 52074 Aachen, Germany

* Correspondence: fazekas@isac.rwth-aachen.de

Received: 8 June 2019; Accepted: 17 July 2019; Published: 19 July 2019

Abstract: The next generation of Intelligent Transportation Systems (ITS) will strongly rely on a high level of detail and coverage in traffic data acquisition. Beyond aggregated traffic parameters like the flux, mean speed, and density used in macroscopic traffic analysis, a continuous location estimation of individual vehicles on a microscopic scale will be required. On the infrastructure side, several sensor techniques exist today that are able to record the data of individual vehicles at a cross-section, such as static radar detectors, laser scanners, or computer vision systems. In order to record the position data of individual vehicles over longer sections, the use of multiple sensors along the road with suitable synchronization and data fusion methods could be adopted. This paper presents appropriate methods considering realistic scale and accuracy conditions of the original data acquisition. Datasets consisting of a timestamp and a speed for each individual vehicle are used as input data. As a first step, a closed formulation for a sensor offset estimation algorithm with simultaneous vehicle registration is presented. Based on this initial step, the datasets are fused to reconstruct microscopic traffic data using quintic Beziér curves. With the derived trajectories, the dependency of the results on the accuracy of the individual sensors is thoroughly investigated. This method enhances the usability of common cross-section-based sensors by enabling the deriving of non-linear vehicle trajectories without the necessity of precise prior synchronization.

Keywords: sensor synchronization; microscopic traffic data; trajectory reconstruction; expectation maximization; vehicle matching

1. Introduction

As the future of road transportation is being shaped around the idea of autonomous mobility, new methods of data acquisition and processing are being developed. Especially in the field of driving assistance systems, there is much current research on detecting and localizing vehicles with many different sensors like radar, LiDAR, computer vision, and acoustics. On the other side, infrastructure-based Intelligent Transportation Systems (ITS) still need further development to exploit the full potential of the already existing sensors. This especially means increasing the level of detail reached with current ITS techniques, which often only deliver aggregated traffic data consisting of datasets with a time resolution of minutes covering traffic parameters like traffic flux (vehicles/hour), density (vehicles/km), and speed (km/hour). This kind of data is a limiting factor for comprehensive analysis on the interaction between individual vehicles. For such detailed analysis, traffic data are required on a microscopic scale, which includes the quasi-continuous trajectory of every vehicle on the road.

The analysis of vehicle-to-vehicle interactions is indispensable for a detailed traffic safety analysis also covering risky situations between vehicles beyond only counting traffic accidents [1]. Many of the surrogate safety indicators rely on such a level of detail [2]. In [3], it was shown that more safety-critical

interactions happen when traffic is fluent and vehicle speeds are still at least moderate. In this case, high speed differences lead to more critical time-to-collision values and required deceleration rates to avoid a crash. While analyzing non-fluent traffic can be interesting to calibrate microsimulation models, the lower speeds are less critical from the safety perspective. Beyond safety considerations, efficiency analysis and calibration of traffic flow models can greatly benefit from microscopic traffic data [4]. While these applications rely on the offline processing of the raw data, the work in [5] showed how microscopic traffic models can be integrated into implementing methods for collaborative self-driving vehicles. Similarly, the work in [6] showed how such data can be used to model unmanaged intersections analytically. All these model-based applications could lead to real-time solutions of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) collaboration to support self-driving vehicles in the future.

1.1. State-of-the-Art Data Acquisition

Several alternative techniques exist today, capable of acquiring real microscopic traffic data. Of course, one of the most straightforward methods is equipping a large number of vehicles with high-precision Global Navigation Satellite System (GNSS) sensors and recording data from each vehicle [7]. Such methods can also be enhanced by inertial systems [8] to increase positioning accuracy. Other in-car methods can also be used like computer vision systems [9] and fusion of different sensor technologies [10]. While delivering high accuracy and large road coverage, all these techniques share the disadvantage of low traffic coverage. This means that specifically looking at a critical road section of interest and analyzing the dynamics of most vehicles passing through the section is practically impossible. On the other hand, analyzing the dynamics of vehicles specifically equipped with sophisticated sensor systems could be ambiguous, as the behavior of the drivers could be far from natural due to the necessary equipment installed in the vehicles.

A better approach to recording traffic data with a large coverage is the use of infrastructure-based sensors. A very common technique is based on inductive loops [11,12]. They have great reliability and real-time capabilities and are thus widely used for both highway and urban traffic management. Similar to loop detectors, radar sensors work well in various weather and illumination conditions, enabling the detection of individual vehicles and measuring their speeds [13,14]. Other types of static sensors like laser [15], LiDAR [16], and acoustic sensors [17,18] are also capable of detecting vehicles with their respective speeds. Many of the existing commercial products are very easy to use, as they do not need to be installed at great heights. They can be deployed on the ground at the road side, without the necessity of lane closures. This also enables them to be supplied by a battery without the necessity of complicated cabling. The key common drawback of these sensors though is that they gather traffic data from cross-sections, by counting vehicles and measuring their speeds [19]. Thus, the standard use case of these sensors does not enable recording microscopic traffic data over an entire area.

While there are a few studies showing how radar- [20] and laser-based [21] sensors can deliver microscopic traffic data, the most common technique by far is by means of computer vision [22–24]. With the appropriate technique of back-projection from pixel coordinates to real-world coordinates, computer vision-based techniques are a cost-effective, non-intrusive, and flexible way of recording microscopic traffic data. Even so, in many cases, the area covered is limited by occlusion, while a dense deployment of cameras is also limited due to the difficulty of road side installation. More specifically, the deployment of surveillance cameras for automatic data acquisition needs thorough planning because they typically need to be deployed at a height where side poles or bridges are required. Installing the cameras at the required height often requires the use of trailer-mounted work platforms and lane closures. Furthermore, the consultation of road administrations, operators of the poles, and sometimes even public transport organizations is necessary. Many cameras additionally require power cabling, with the respective effort and cost, or the installation of batteries on the poles, which can be safety critical. Thus, reducing the number of required cameras by filling gaps for interim sections is of great interest.

An additional difficulty in using already deployed CCTV cameras for computer vision methods is measuring the exact sensor distance or accurately synchronizing the video streams. In our research project “Highly automated tunnel surveillance for catastrophe management and regular operation” (AUTUKAR) regarding automatic video analysis in tunnel surveillance systems, we can only rely on the existing planning documents of the tunnel to determine distances between cameras, which often do not have the desired accuracy. On the other hand, when using real-time video analysis based on video streams, network bandwidth limitation means that a very accurate time synchronization between the streams cannot be guaranteed.

Finally, using cameras is also often restricted by data privacy, a very delicate subject especially in European countries, which often makes application of these sensors impossible. Using sensors that are 100% compliant with privacy laws makes a huge difference in European countries.

1.2. Contribution

In this paper, we present a method of spatio-temporal synchronization (offset estimation) of sensors and an appropriate fusion technique to reconstruct microscopic traffic data. The sensors record limited traffic data, which are covered by common detection sensors at fixed cross-sections. Firstly, this results in an enhancement that enables radar and loop detectors to generate area-based microscopic traffic data, even when a precise sensor synchronization is not possible. Secondly, it allows the filling of gaps of data already covering a limited road section, thus reducing the sensor density requirement of computer vision-based systems. The method solves the following problems simultaneously:

- Spatio-temporal offset estimation when measuring the distances between sensors or an exact clock synchronization is not possible
- Vehicle registration without any specific identification like number plates
- Reconstruction of vehicle trajectories with acceleration/deceleration maneuvers only based on cross-section recordings.

1.3. Impact

Thus, the presented work has a great impact by enhancing the usability of cross-section-based vehicle detectors by enabling them to acquire microscopic traffic data based only on unsynchronized records of detected timestamps and respective speeds. As many of the cross-section-based sensors do not require recording images or videos, using this method enables the acquisition of vehicle trajectories without any privacy issues. The method also greatly reduces the financial, organizational, and maintenance effort for camera-based acquisition with the ability of filling interim gaps between covered road sections, thus leading to less sensors required to cover a given length. With the proposed method, simple and cost-effective sensors can be used for a detailed safety analysis. If such sensors are deployed along highway exit lanes, this method can be adopted to derive parameters such as individual deceleration rates along the exit lane and to support decision making in regards to appropriate traffic harmonization methods. Furthermore, surrogate safety indicators such as space headways, time-to-collision, or minimal deceleration rates to avoid a crash can be derived, to better understand the evolution of these potential conflicts over time. In addition to presenting the basic principles of the method, we will also analyze the limitations of the method with respect to traffic volume, cross-section distance and detection accuracy.

The remainder of the paper is organized as follows: Section 2 presents the general methodology of the vehicle matching (Section 2.1) and trajectory derivation (Section 2.2). Section 3.1 shows how the underlying data for our experiments have been recorded, while Sections 3.2 and 3.3 present the experimental validation based on synthetic and real data. In Section 4, we draw the conclusions and discuss potential future research based on our approach.

2. Methodology

As formulated in the previous section, we have three different problems to solve: the coordinate systems of the sensor data are not completely synchronized; there is no known matching between individual vehicles; and the dynamic behavior of the vehicles between the two measured cross-sections, which we call gap sections, are unknown. We solved the first two problems simultaneously by making an assumption about the vehicle dynamics, namely that the sensors are close enough and the traffic is fluent enough, so that on average, the drivers make limited deceleration or acceleration maneuvers. This assumption is feasible for the considered orders of magnitude for the distances considered in this work of 100 m–200 m, because in fluent traffic, such sections are covered within several seconds, which restricts the amount of maneuvers. We must also note that it is of great interest to reconstruct microscopic data of fluent traffic as this state leads to high-risk interactions between vehicles due to large speed differences. For non-fluent traffic, analytically determining more complex dynamic behaviors between cross-sections is impossible without the means of driving behavior modeling. Additionally, low traffic speeds also lead to less safety-critical interaction, so from the point of view of traffic data analysis on a microscopic scale, this state is less of an interest than fluent traffic.

We formulate the assumption by describing the vehicles' movement along the street axis (in one lane) as:

$$s_i(t) = s_i(t_0) + v_i(t_0)(t - t_0) + a_i(t - t_0)^2, \quad (1)$$

where $s_i(t)$ is the position of vehicle i at time t , $s_i(t_0)$ and $v_i(t_0)$ are its location and speed at the beginning of the gap section, and a_i is its time-constant acceleration.

From Equation (1), it can also easily be derived that:

$$s_i(t) = s_i(t_0) + \frac{v_i(t) + v_i(t_0)}{2}(t - t_0) \quad (2)$$

so that the position is the same as if the vehicle would have moved with the mean speed between t_0 and t .

2.1. Vehicle Matching and Offset Estimation

Based on the assumption formulated above, we treated the vehicle matching as a probability density estimation problem. To solve it, we applied the expectation maximization algorithm described in [25]. The algorithm aims at finding the maximum likelihood estimates from incomplete data by looping over two iterations:

- Estimation step (E-step): Find an estimate for the complete data sufficient statistics.
- Likelihood Maximization step (M-step): Determine the parameters of the distributions based on the estimated data.

This method was originally intended to determine the parameters of a few statistical distributions based on a larger set of incomplete data. We applied this method to our problem as follows: Suppose we have two sensors detecting vehicles with their time-stamps and speeds at two consecutive cross-sections. The set of M vehicles with their respective speeds from the second sensor form linear Gaussian kernels, while the N vehicles and speeds from the first sensor form the data upon which we are trying to find the distribution parameters. The mean of the individual Gaussian models is defined by the mean speed between a vehicle pair as a linear trajectory, as defined in (2). The standard deviation is defined as the distance of the data point to the trajectory line. The geometrical interpretation of this distance can be seen in Figure 1, where $y_m = (t_{m0}, s(t_{m0}))$ is the recorded time and location of a vehicle entry from one sensor. x_n is an entry from the dataset of the other sensor. We define the vector $w_{mn} = x_n - y_m$ and u_{mn} as being the unit vector of the linear vehicle trajectory calculated from the mean speed between the two entries. We calculate the perpendicular vector w_{mn}^\perp from the desired linear trajectory to x_n as follows:

$$\begin{aligned}
 w_{mn}^\perp &= -proj_{u_{mn}}(w_{mn}) + w_{mn} = -u_{mn} \frac{u_{mn}^T w_{mn}}{\|u_{mn}\|} + w_{mn} = \\
 &= w_{mn} - ((u_{mn}^T w_{mn})^T u_{mn}^T)^T = w_{mn} - (w_{mn}^T (u_{mn} u_{mn}^T))^T = \\
 &= w_{mn} - u_{mn} u_{mn}^T w_{mn} = (I - u_{mn} u_{mn}^T) w_{mn}
 \end{aligned}$$

Thus, the distance of the point to the trajectory line is given by:

$$\|w_{mn}^\perp\| = \|(I - u_{mn} u_{mn}^T) w_{mn}\| = \|(I - u_{mn} u_{mn}^T)(x_n - y_m)\|$$

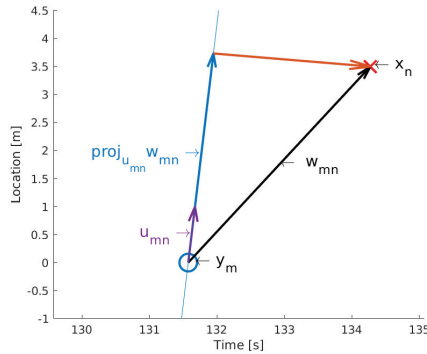


Figure 1. Geometric interpretation of the distance of a data point from the line trajectory of another data point. The data point y_m is the time and location of a vehicle detection in the first sensor, while x_n is a corresponding data entry from the second sensor. u_{mn} is the unit vector with the gradient calculated as the mean of the two speeds measured at the sensors. $proj_{u_{mn}} w_{mn}$ is the projection of w_{mn} onto this unit vector. The distance of the second data entry from the optimal linear trajectory corresponding to the mean speed can be derived by subtracting the projected vector from w_{mn} .

The Gaussian model of the vehicle pair (m, n) has the form:

$$p(x_n | \Theta_m) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\|(I - u_{mn} u_{mn}^T)(x_n - y_m)\|^2}{2\sigma^2}}, \tag{3}$$

In the rest of the paper, we will use the notation $U_{mn} = I - u_{mn} u_{mn}^T$ for the projection matrix used to calculate the perpendicular vectors on the unit vector u_{mn} . The Gaussian distribution of all vehicles use the same standard deviation, while they have equal membership probabilities $P(m) = \frac{1}{M}$. The mixture model is defined as:

$$p(x | \Theta) = \sum_{m=1}^M p(m) p(x | \Theta_m), \tag{4}$$

while the log likelihood function given the set of model parameters is:

$$\ln(p(\mathbf{x} | \Theta)) = \sum_{n=1}^N \ln \sum_{m=1}^M p(m) p(x_n | \Theta_m). \tag{5}$$

Figure 2 shows a probability distribution of a Gaussian mixture model defined by Equation (4) with two vehicle trajectories, with the first one having a higher speed than the second and the vehicles passing the detection sensor with a difference of one second.

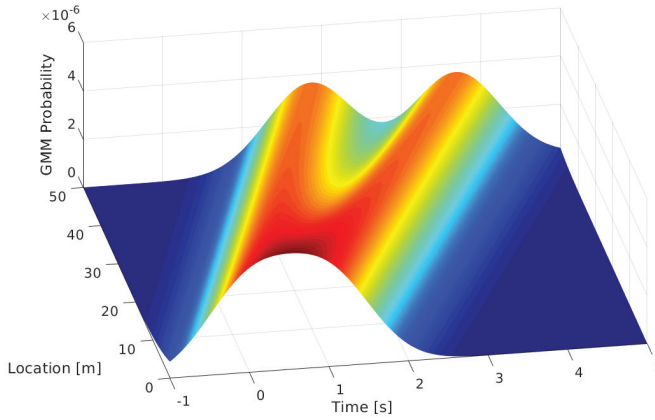


Figure 2. Distribution of a Gaussian mixture model consisting of two linear vehicle trajectories. If there is one data entry x with a specific speed and two possible data entries y , it results in two projected vectors $proj_{u_{mn}} w_{mn}$ and two Gaussian distributions from Equation (3). The Gaussian distributions depend on the orthogonal distance from the projected vector and add up to the Gaussian mixture model seen in this figure.

As described above, the maximization of this likelihood function is solved by looping through two steps. The first one is calculating the expected complete-data log likelihood function given by:

$$Q(\Theta, \Theta^{old}) = \sum_{n=1}^N \sum_{m=1}^M p(m|x_n, \Theta^{old}) \ln[p(m)p(x_n|\Theta_m)]. \tag{6}$$

The missing data consist of the soft matching of the detected vehicles so that (6) can be derived by determining the a posteriori probabilities of Gaussian model m matching vehicle n :

$$p(m|x_n, \Theta^{old}) = \frac{p(x_n|\Theta^{old})}{\sum_{m=1}^M p(x_n|\Theta^{old})} = \frac{\exp\left\{-\frac{\|U_{mn}(x_n - y_m)\|^2}{2\sigma^2}\right\}}{\sum_{m=1}^M \exp\left\{-\frac{\|U_{mn}(x_n - y_m)\|^2}{2\sigma^2}\right\}} \tag{7}$$

The second step consists of maximizing the Q function given by (6) with respect to the Gaussian parameters. At this point, we have to define the set of parameters Θ being considered here for optimization. As already stated above, one of the common parameters is the variance of the Gaussians given by σ^2 . As our initial problem was to determine the offset of the sensors, the other parameter is a translation, which moves the Gaussians along the space and/or time axes. As the set of Gaussians is based on vehicle data recorded from one sensor, it is feasible to use a common translation vector \mathbf{t} for all the Gaussians, as the offsets have to be consistent for all vehicles recorded within a sensor. After applying the logarithm to the exponential function, we get:

$$Q(\Theta, \Theta^{old}) = -N \ln(M\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^N \sum_{m=1}^M p(m|x_n, \Theta^{old}) \|U_{mn}(x_n - (y_m + \mathbf{t}))\|^2 \tag{8}$$

Note that in Equation (8), the first term is constant and can be neglected. Taking the partial derivative with respect to the translation parameter \mathbf{t} and equating it to zero give:

$$\mathbf{t} = \mathbf{U}^{-1} \left(\sum_{n=1}^N \sum_{m=1}^M p(m|\mathbf{x}_n, \Theta^{old}) U_{mn}(x_n - y_m) \right), \quad (9)$$

where $\mathbf{U} = \sum_{n=1}^N \sum_{m=1}^M p_{mn} U_{mn}$ and \mathbf{U}^{-1} is its inverse matrix. By substituting \mathbf{t} back into the function Q , we can maximize with respect to σ again by partial derivation and equalizing with zero:

$$\sigma^2 = \sum_{n=1}^N \sum_{m=1}^M p(m|\mathbf{x}_n, \Theta^{old}) \|U_{mn}(x_n - \hat{y}_m)\|^2, \quad (10)$$

where $\hat{y}_m = y_m - \mathbf{t}$. In accordance with the EM-algorithm, we cycle through the E- and M-step until the change in either the log-likelihood function or in the parameters of the Gaussian models (σ and \mathbf{t}) is small enough to consider that the algorithm has converged.

We summarize our approach as an algorithmic description in Algorithm 1. Here, the E-Step consists of recalculating the soft matching probabilities, in other words the probability that vehicle m from the first sensor matches vehicle n from the second one. The M-step is a spatio-temporal shift of all M vehicles from the second sensor with a common value \hat{t} and an update of the standard deviations of the Gaussians.

Algorithm 1: Vehicle registration using expectation maximization.

Data:

$x_n = (t_n, s_n)$ vehicle data from first sensor, $n \in (1, N)$

$y_m = (t_m, s_m)$ vehicle data from second sensor, $m \in (1, M)$

$v_{mn} = \frac{1}{2}(v_n + v_m) \leftarrow$ mean speed between sensors measurements

Initialization:

$\mathbf{t} = 0$

$\sigma^2 = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M \|U_{mn}(x_n - y_m)\|^2$

$u_{mn} = v_{mn} / \|v_{mn}\|$, $U_{mn} = (\mathbf{I} - u_{mn} u_{mn}^T)$

while $\Delta Q > T_Q$ & $(\Delta \sigma > T_\sigma \parallel \Delta \mathbf{t} > T_t)$ **do**

 E-Step:

$$\hat{p}_{mn} = \frac{\exp\left(-\frac{\|U_{mn}(x_n - y_m)\|}{2\sigma^2}\right)}{\sum_{m=1}^M \exp\left(-\frac{\|U_{mn}(x_n - y_m)\|}{2\sigma^2}\right)}$$

 M-Step:

$$\mathbf{U} = \sum_{M} \sum_N \hat{p}_{mn} U_{mn}$$

$$\hat{\mathbf{t}} = \mathbf{U}^{-1} \left(\sum_{M} \sum_N \hat{p}_{mn} U_{mn}(x_n - y_m) \right)$$

$$\hat{y}_m = y_m + \hat{\mathbf{t}}$$

$$\sigma^2 = \sum_{M} \sum_N p_{mn} \|U_{mn}(x_n - \hat{y}_m)\|^2$$

end

Result:

$p_{mn} \leftarrow$ probabilities of vehicle matches

$y_m = (t_m, s_m) \leftarrow$ corrected position and/or timestamps of the measurements for the second sensor

2.2. Microscopic Data Reconstruction

In this section, we will take the next step in reproducing microscopic traffic data, based on the results of the previous section. Let us suppose that with the means of the EM-algorithm, we found the spatio-temporal offset of the sensors, and we also have a correspondence matrix \mathbf{P} where the entries p_{mn} are the last computed prior distribution of the vehicle matching problem from Algorithm 1. We can compute the matrix \mathbf{M} with the entries $r_{mn} = \|U_{mn}(x_n - y_m)\|$ being the respective point to line distances.

Now, after we found the sensor offsets and the soft matching probabilities r_{mn} , we applied the Hungarian algorithm [26] to find the first correspondences between vehicles. The Hungarian algorithm is a polynomial time solution to the assignment problem of two datasets. We treated the vehicles of the first and second sensor as the two different datasets. The Hungarian algorithm uses a cost value for the assignment, which in our case was the distance from the linear trajectory from the already shifted entries of the second sensor to entries of the first one. Therefore, we used the same r_{mn} values as a cost value in the EM-algorithm and formed a matrix R with the number of rows and columns corresponding to the data entries of the first and second sensor. The overall procedure of the Hungarian algorithm is as follows:

- **Matrix reduction:** The cost matrix is reduced row-wise with the respective minimal value of the row. In cases of entire columns greater than zero, a column-wise reduction is also performed.
- **Line covering:** Cover resulting zeros with the minimum number of lines (horizontal and vertical). If the number of zeros, which are unique in the respective row and column, is equal to the smallest of the two matrix dimensions (number of vehicles in first or second sensor), an optimal matching has been found.
- **Additional reduction:** Reduce all non-covered elements by the minimal non-covered value and add that value to all elements covered by both horizontal and vertical lines. Continue with line covering again.

The result of the assignment problem is a set of pairs, where for each vehicle of the first sensor, a vehicle of the second sensor is matched. The assignment is only an interim step towards reconstructing the trajectories as it still assigns the vehicle pairs based on a linear trajectory based on mean speed between the vehicle pairs. However, the assignment is very robust, because the real trajectories of the vehicles are similar enough to the linear ones in order for the matching to be successful.

We also have to note at this point that the resulting assignments also hold assignments of outliers, as they have also been translated according to the EM-algorithm. Thus, both false positives and false negatives in the data of both sensors need to be considered. We applied a threshold of 3σ to the assigned values of r_{mn} . We considered all the assignments with costs above this threshold as outliers and removed them from the dataset, as they were misdetections (false positive or negative) of either sensor. A trajectory should not be reconstructed for such singular data entries. One important aspect here is that we have no possibility of identifying a false positive that fits well to a trajectory of a true positive detection. If a sensor delivers a false positive that fits a smoother trajectory than the true matching, that will influence the assignment of the Hungarian algorithm and also the prediction of the resulting trajectories. On the other hand, most of the sensors used in our experiments showed a tendency towards false negatives rather than false positives, so the chance of a false positive detection influencing the results was quite low. We will analyze the effect of false detection more thoroughly in the following section.

After finding appropriate assignments, we reconstructed the trajectory of the vehicles between the two sensors using curves, which can be parametrized to connect the two points while also following the detected speeds profile. In other words, the connecting curves $s(t)$ need to satisfy:

$$s(t_n) = s_n, \quad s(t_m) = s_m, \quad \frac{\delta s}{\delta t}(t_n) = v_n, \quad \frac{\delta s}{\delta t}(t_m) = v_m \quad (11)$$

where (t_n, s_n) and (t_m, s_m) are the time and positions of the assigned sensor detections, while v_n and v_m are the respective speeds. The polynomial curves used in this work were Bézier curves [27]. They have very useful properties, being easy to calculate and derive, so acceleration and deceleration rates can easily be analyzed. Thus, the use of Bézier curves enables the acquisition of a relatively high amount of dynamic behavior of road users, given the fact that only simple cross-section data were used and no complex driver behavior models were applied.

The general definition of Bézier curves is:

$$\mathbf{B}(u) = \sum_{i=0}^n \mathbf{b}_i B_{i,n}(u), \tag{12}$$

where:

$$B_{i,n}(u) = \begin{cases} \frac{n!}{(n-i)!i!} (1-u)^{n-i} u^i & , \text{if } 0 \leq i \leq n \\ 0 & , \text{otherwise} \end{cases} \tag{13}$$

called Bernstein polynomials of degree n . \mathbf{b}_i are the control points that define the curve, and as u increases within the interval $[0, 1]$, the Bézier curve is drawn from the first towards the last control point. From Equation (12), the general formulation of the r^{th} derivative of a Bézier curve of degree n can be deduced to:

$$\mathbf{B}^{(r)}(u) = \sum_{i=0}^{n-r} \mathbf{b}_i^{(r)} B_{i,n}(u), \tag{14}$$

where:

$$\mathbf{b}_i^{(r)} = n(n-1) \dots (n-r+1) \sum_{j=0}^r (-1)^{r-j} \binom{r}{j} \mathbf{b}_{i+j}. \tag{15}$$

We now proceed to define the control points based on the input data. The requirements defined by Equation (11) ensure that the positions and the speed values are respected by the trajectory. The disadvantage is that there is no continuity in the second derivative. In other words, a vehicle trajectory fulfilling those requirements will have an unrealistic speed change directly at the first and last control points. Thus, we add the continuity requirements of the second derivative in order to reconstruct the acceleration and deceleration in a more realistic manner. From Equations (12)–(15), it can be seen that the control points can be computed by setting the parameter u to zero and one using the derivatives and equating it with the values defined by the input data. In our case, we had six conditions in total (Equations (11) plus the continuity of the second derivative at both ends). These lead to a quintic Bézier curve, the control points of which can be derived as:

$$\mathbf{b}_0 = (t_n, s_n), \quad \mathbf{b}_5 = (t_m, s_m) \tag{16}$$

$$\mathbf{b}_1 = (t_n, s_n) + \frac{1}{5}(1, v_n), \quad \mathbf{b}_4 = (t_m, s_m) - \frac{1}{5}(1, v_m) \tag{17}$$

$$\mathbf{b}_2 = (t_n, s_n) + \frac{2}{5}(1, v_n), \quad \mathbf{b}_3 = (t_m, s_m) - \frac{2}{5}(1, v_m) \tag{18}$$

Figure 3 shows the iterative process of the proposed method. In the first image (a), the initial datasets are shown, where the crosses and circles are the timestamp and location entries of the first and second sensor, respectively. Note that all entries are at Location 0, as there is no prior knowledge of the location or time offsets between the sensors. In the second image (b), the Bézier curves of the matches are visualized. At the beginning of the algorithm, all pairs have similar matching probabilities, which is visualized by the similar opacity of the curves. The S-shape of the curve is originating from the condition of the speed continuity of the curve at the first and second detection with an unfinished spatial offset estimation. In the following plots, the spatio-temporal synchronization is shown by the shift of the circles (detections in the second sensor), while all the unrealistic curves get more transparent (less probable) until saturation. Please note that in this example, the reconstructed trajectory is the longitudinal one, so the resulting $s(t_n)$ is the vehicle position along the road. As we will see in later sections, this work was conducted with the use of sensors that only deliver the timestamps of vehicles passing, without measuring the lateral position on the road. Nevertheless, the method can easily be applied to the lateral position in the same way in order to deliver lane changing maneuvers.

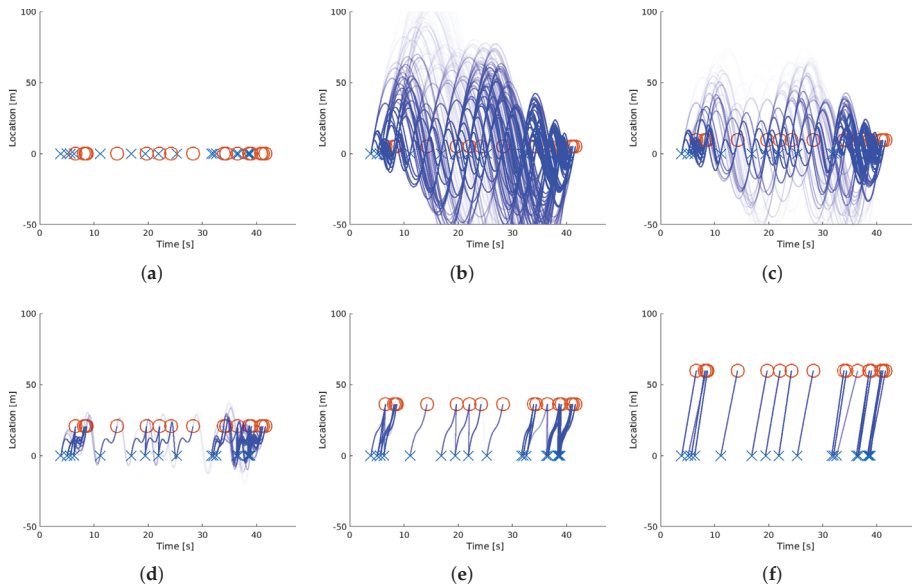


Figure 3. Iterative steps of the EM-algorithm with the visualization of the corresponding trajectories. The top left plot (a) shows the initial datasets, while the plots (b–f) are the visualizations of Iteration Steps 2, 4, 10, 24, and 100, respectively. In this figure, the opacity of the blue lines correspond to the results of the E-step, and the shift of the red circles corresponds to the M-Step of the EM approach, as described in Algorithm 1.

3. Experimental Results

In this section, we present the results of the behavior analysis of the described methods. In order to allow a comprehensive performance investigation, we used the different datasets consisting of synthetic, GNSS, and radar data. The validation steps applied were based on appropriate performance metrics for the problems presented in Section 2: matching, offset estimation, and trajectory reconstruction. The metrics will be discussed in more detail in this section.

3.1. Underlying Dataset

We based our validation on data recorded on the expressway “Pariser Ring” in Aachen on 30 November 2018 (Figure 4). The experiment was conducted with the use of four radar devices that detected passing vehicles and recorded their timestamp and speed at the cross-section of the device. The distances between consecutive devices were measured and consisted of 155 m, 35 m, and 35 m, respectively.

It is important to note that the radar devices had separate internal clocks with limited synchronization possibilities. The resolution of the recorded timestamp was only one second, while the resolution of the speed was 1 km h^{-1} . Another limitation of these devices is that they were installed and configured with a given angle to the street, and the exact cross-section of the detection was not known. We will analyze the effect of these limitations and also demonstrate that even under these circumstances, a reasonable reconstruction result is achievable.



Figure 4. The location of the data recording. Four radar devices were installed at the poles marked here with yellow. The red line shows a path recorded with an RTK-GNSS sensor mounted on a vehicle.

3.2. Synthetic Data Validation

The validation based on synthetic data was mainly motivated by the possibility of a systematic performance analysis. First, if necessary, we can easily control the different sources of error of the recording devices. We can, for example, eliminate the quantization of the recording resolutions or errors in the detection location. This gives the possibility to analyze specifically the effect of each error source separately, which cannot be done with real data. Secondly, we can apply a preconfigured range for measurement errors (“synthetic errors”) in order to analyze specifically their effect on the results. Finally, the underlying microscopic vehicle data can also be generated and used as a perfect reference. In comparison, real reference data measured in vehicles can never be completely errorless.

We based the synthetic data generation on the measurements of the first radar device, as it recorded all passing vehicles on three lanes and thus delivered the largest and densest data volume. We extracted the data of the morning rush-hour and overcame the limitation of time resolution by adding a uniformly-distributed number of milliseconds between zero and 1000 to each recorded entry. We kept the recorded speed values and chose a second cross-section at a specified distance d . Then, for each vehicle, we generated a normally-distributed acceleration value with mean m_a and standard deviation σ_a , and we calculated the time when the vehicle would reach the second sensor if it used the generated acceleration value. We then split the data, treating the start and end positions as the dataset of the first and second sensor accordingly. We of course neglected the location information from the datasets so that the original offset had to be reconstructed by our algorithm. In the different validation steps presented, we varied specific parameters like measurement errors and data volume. For each of the parameter values, we applied the EM algorithm a fixed number of times to have the mean and variance of the resulting output. We were also able to apply the algorithm with optimization in the space/time domain separately or with combined spatio-temporal optimization.

The considered performance metrics are defined as follows:

- **Offset reconstruction error:** the deviation between the sensor offset reference value and the reconstructed one. Depending on spatial or temporal optimization, this involves the location or time offset in meters or seconds.
- **Trajectory reconstruction error:** the deviation between reference vehicle trajectories and the reconstructed ones measured as a root-mean-squared error in meters.
- **Matching sensitivity:** the recall value between the correctly-matched pairs after reconstruction and the pair of the reference dataset.

- **Matching relevance:** the positive predictive value between the correctly-matched pairs after reconstruction and all the matched pairs.
- **Number of iterations:** the number of iterations the EM-algorithm requires to converge.

3.2.1. Errorless Data

In this first validation step, we considered the case where the timestamp and speed of the vehicles can be recorded without error at the exact cross-section of the sensor. We varied the number of vehicles from 5–400 using a sensor distance of 100 m. Additionally, we varied the distance between the sensors from 50–150 m using 200 vehicles. We used 50 runs for each setting, resulting in a total of 4000 runs for the analysis of the vehicle count and 500 runs for the analysis of sensor distance. We first used only spatial optimization. The results showed that for all the settings, our algorithm was able to reconstruct both the sensor offsets and the vehicle trajectories perfectly, meaning that all appropriate metrics had zero mean. The only metric varying was the number of iterations for convergence, which seemed to depend on the number of vehicles used, but not on the distance between the sensors (Figure 5). We also validated the spatio-temporal estimation when we altered the sensor time offset by shifting each time entry of one dataset by a fixed value between one and 10 s and using a location offset of 100 m and a data volume of 200 vehicles. Just as in the case of the distance variation, the algorithm perfectly estimated both spatial and temporal offsets.

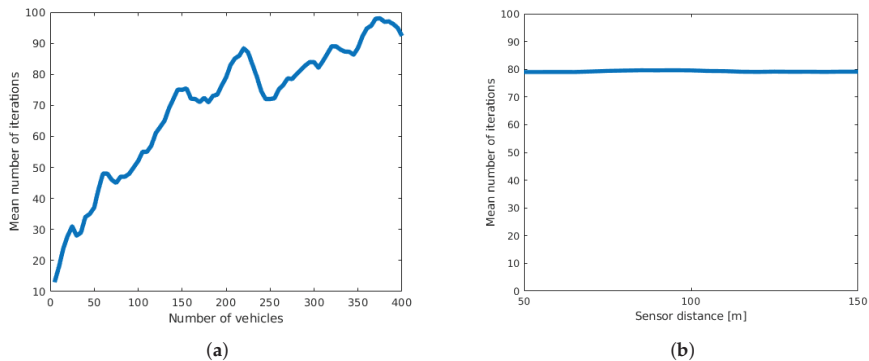


Figure 5. Number of iterations for algorithm convergence depending on (a) the number of vehicles and (b) the distance between sensors.

3.2.2. Variance in Detection Location

Next, we analyzed the effect of varying the detection location. This constituted a type of error resulting from the fact that a sensor will not detect a vehicle at the exact same cross-section as it is installed, although we treated the data as such. In most cases, each vehicle will be detected at an individual cross-section depending on the physical characteristics, which influence the detection. An example would be the reflectance of the vehicle chassis in the case of a radar sensor, which will influence how early a vehicle can be detected. We used a total number of 200 simulated vehicle trajectories for this step, and 75 m, 100 m, and 125 m were used for the sensor distance. Note that we were still assuming that the detector made no error in speed measurement and started with the sensors being perfectly time-synchronized, so we only applied optimization in the space dimension.

From Figure 6, it can be seen that the bias of the offset estimation remained low, while a small variation was present due to the fact that a limited number of measurements (runs) had been used. Both results were independent of the sensor distance. The standard deviation of the estimation error increased linearly with the standard deviation of the detection location, but remained at a much lower level. This can be explained by the fact that the location errors of individual vehicles compensated each other. The right plot of Figure 6 shows that the standard deviation decreased with a higher

number of vehicles. Thus, it is arguably sensible to use more vehicles for optimization to decrease the resulting error. On the other hand, a higher number of vehicles not only led to a higher number of required iterations (Figure 7), but additionally increased the computation time of each iteration as the dimensions of the matrices used in optimization increased.

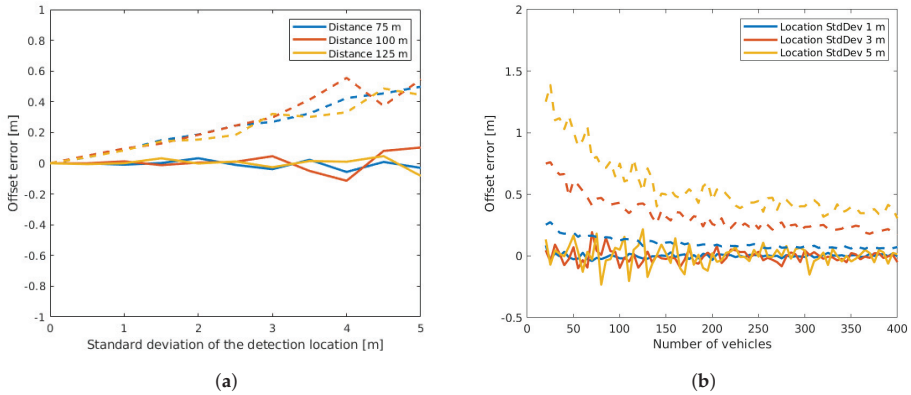


Figure 6. Offset estimation error depending on detection location (a) and depending on the number of vehicles (b). In both plots, the mean resulting error is plotted with a continuous line, while the standard deviation of the error is plotted with a dashed line.

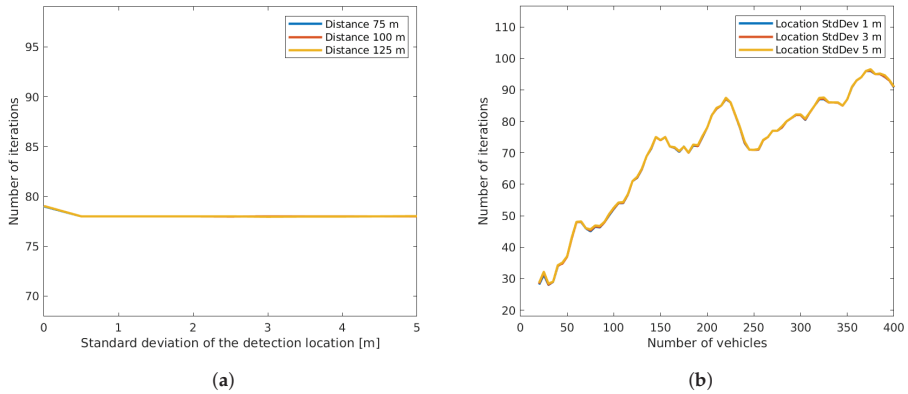


Figure 7. Number of iterations until convergence depending on detection location (a) and depending on the number of vehicles (b).

In contrast to the sensor offset estimation, the mean RMS error of the vehicle trajectories only depended on the variance of the detection location and remained constant with increasing number of vehicles (see Figure 8). The linearity between the standard deviation in detection and the RMS error came from the relation between the end point of the computed trajectory and the calculated curve, which will adapt the location curve accordingly. The standard deviation of the mean RMS error showed a slightly negative slope, due to the variance of the mean being the variance of the individuals divided by the sample size, a property of independent identically distributed data. Thus, as the number of vehicles (sample size) became larger, the variance of the mean error decreased.

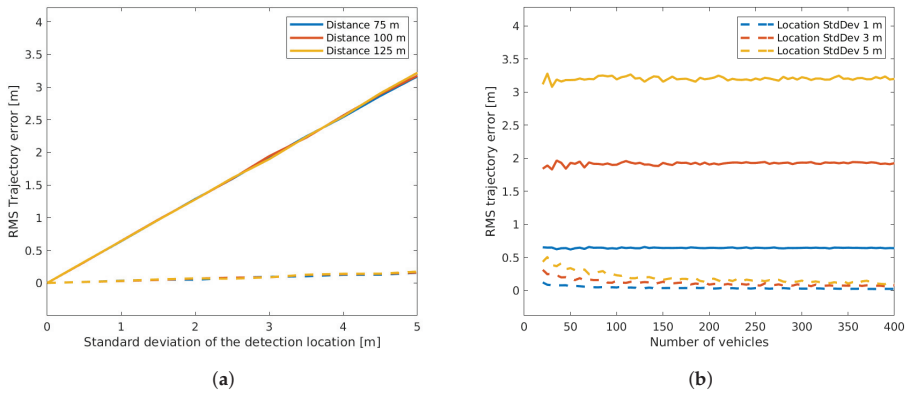


Figure 8. Trajectory RMS error depending on detection location (a) and depending on the number of vehicles (b). In both plots, the mean resulting error is plotted with a continuous line, while the standard deviation of the error is plotted with a dashed line.

We go on to examine the performance of the spatio-temporal offset estimation. Obtaining a time offset of zero is of course expected as we did not specifically add any time shift to the dataset. When looking at the results, the error values showed that while the time offset estimation mean was indeed at zero, the location estimation error significantly increased. Figure 9 shows a scatter plot between the time and space offset errors, where we observed that the values were distributed like a bivariate Gaussian, which seemed to be aligned with the mean speed of the vehicles. Indeed, deriving the eigenvectors of the covariance matrix showed that the principal axes of the Gaussian fitted to the data had a slope almost equal to the mean speed. Thus, when we only used the spatial optimization, the result was spread like a conditional Gaussian distribution at the vertical axes with the time offset of zero. The main reasoning here is that it is objectively reasonable to only use space optimization in order to significantly reduce errors in the estimation of location offset. In the following steps of validation, we will only consider the space optimization.

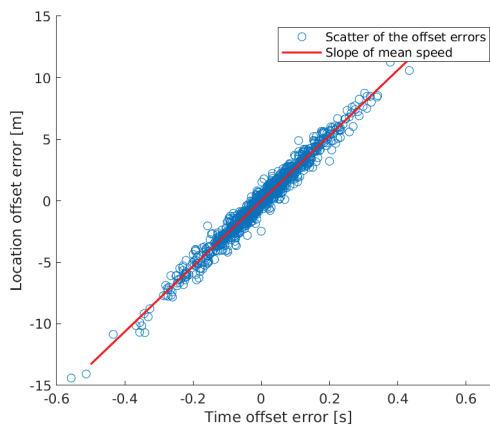


Figure 9. Scatter of the errors made in time and space offset estimation when using spatio-temporal optimization. The errors show a bivariate Gaussian distribution with the main axis representing the slope of the mean speed of the vehicles (red line).

Because the performance showed very limited dependence on the sensor distance used, we will limit our further analysis to a distance of 100 m.

3.2.3. Error in Speed Measurement

One other important source of error when recording traffic data consists of measuring the wrong speed of the vehicles. Although many of the vehicle detectors used in enforcement applications have a very high-quality standard and measurement accuracy, this error can never be completely neglected. Moreover, detectors developed for simpler applications like gathering traffic statistics most probably have lower accuracy standards. Thus, it is important to examine the influence of this error. In order to reduce the complexity of the results, we will neglect variations in the detection location. When not explicitly stated, the number of vehicles used for the optimization will be 200.

In the left plot of Figure 10, we show the results of the offset estimation after altering the measured vehicle speeds of the second sensor with a Gaussian noise, given its standard deviation. From this plot, we can see that the mean error of the offset estimation will only change with the mean error of the speed measurement. Additionally, the relationship between the values can be derived from the mean speed values of the vehicles, which was approximately 25 m/s. When changing the mean speed values of the second sensor, the slope of the linear trajectory used in the EM-algorithm changed to half that value. A distance between the sensors of 100 m led to 4 s of passing time, so the increase of the mean offset error would in this specific case be approximately double the value of the mean speed error. Similarly, the standard deviation of the resulting offset error only depended on the standard deviation of the speed measurement error, but not on the mean value. The right plot of the figure shows that the mean trajectory error changed linearly with $\sigma_{\epsilon v}$ when the mean speed error was zero. This is similar to what we see in Figure 8, although the mean RMS error was lower, while its standard deviation was higher. This means that a zero mean error in speed measurement will have a smaller influence on the result, but the error will be less predictable. In the case where the speed error had a mean value different from zero, the mean RMS error will be dominated by the mean measurement error rather than by its variance. Just as we have seen in the discussion of the detection location error, we can reduce the offset estimation error by increasing the number of vehicles. Naturally, this does not apply to the trajectory errors and will increase the number of iterations and the computation load.

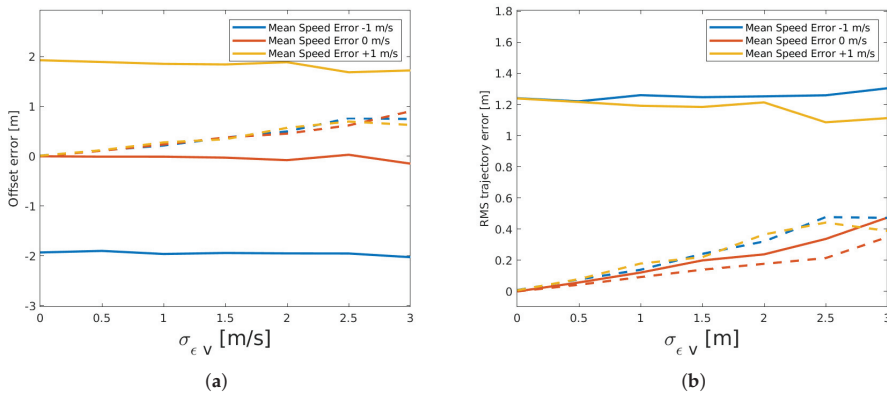


Figure 10. Influence of speed measurement error on the offset estimation (a) and on the trajectory RMS error (b). Continuous lines show the mean values, while dashed lines show the standard deviation of the error.

3.2.4. Quantization

Another limitation of the sensors we used was the level of quantization of the recorded values. Especially, the coarse resolution of time was a limiting factor when using our radar detectors, as they were only able to record timestamps up to a resolution of one second. This led to the necessity of examining the influence of the quantization on the results using our synthetic dataset. We conducted this experiment by again using 200 vehicles and three sensor distance values of 50 m, 100 m, and 150 m, as we expected that the distance could change the outcome. We neglected all previously-discussed errors by setting them to zero mean and variation. We applied a quantization to the time values of both sensors. We implemented the quantization by means of a factor value iterating from 1–10, which defined the partition of a second to which the recorded values were rounded. With a factor of 10, the values were rounded to a tenth of a second, while with a factor of one, the values were discretized to seconds just as the case in our real-world measurement.

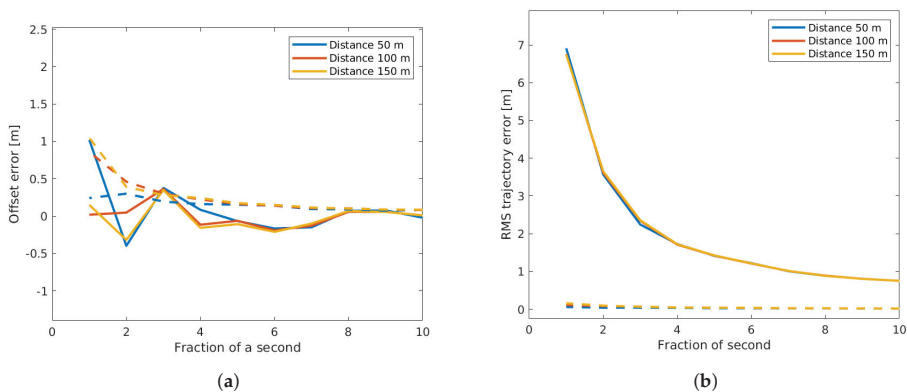


Figure 11. Influence of quantization on the offset estimation (a) and on the trajectory RMS error (b). Continuous lines show the mean values, while dashed lines show the standard deviation of the error.

As we can see from Figure 11, the quantization of the values did have a significant effect on both the sensor offset estimation and the trajectory reconstruction. Note that the lower values of the fraction of a second corresponded to a coarser quantization. As one would expect, the lower values not only led to a higher mean offset estimation error, but also to a higher standard deviation of this value. There was also some change due to the used distance, as a very coarse quantization led to higher errors when low distances were used. This can be explained by the fraction of the quantization time out of the complete passing time of the vehicles. If the quantization of the time values was in the magnitude of a second, then the slopes computed in the EM-algorithm were altered much more when short time ranges were used, thus leading to higher resulting errors. When we examine the outcome of the RMS trajectory error, we can clearly see that the errors in the trajectory reconstruction were a lot higher than in the offset estimation. This again can be explained by the compensation effect of using many different vehicles to estimate the sensor offset, while the errors of individual vehicles remained quite high. We can also see that the sensor distance did not affect the trajectory error. The other metrics, like the number of iterations and the matching sensitivity, were largely unaffected by quantization.

3.2.5. Outliers

We have already discussed many possible measurement errors given a vehicle has correctly been detected. Nevertheless, depending on the circumstances of the recording and on the traffic density, the detectors will also record to some extent false negative and positive detections. For the presented algorithm, the false negatives of one sensor led to outliers in the dataset of the other sensor. Thus, in this validation step, we first examined how these outliers reduced the resulting accuracies. For the

analysis, we varied a percentage of false negative detections for both sensors. We simply removed detections from both sensors at random accordingly to the false negative rate chosen. For each value, we again applied many runs with our algorithm, in each run generating new vehicle trajectories and deleting some of them as described. In Figure 12, we can see that the offset estimation error was zero up to a False-Negative-Rate (FNR) value of 0.25, after which the results became very unreliable. We must note here that in order to be able to examine the effect of outliers specifically, we neglected all other sources of error, which led to the estimation error value of zero. The number of required iterations also remained at a fairly low level up to the FNR value of 0.25.

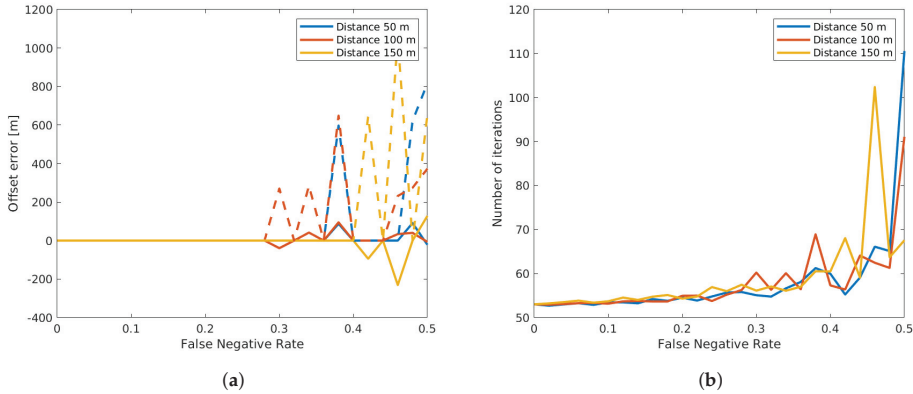


Figure 12. Influence of false negatives in the offset estimation (a) and on the number of required iterations (b). Continuous lines show the mean values, while dashed lines show the standard deviation of the result.

A very interesting effect can be seen in Figure 13, where the left plot shows that the matching sensitivity decreased quite fast. This means that out of all available matches still available in the dataset after deleting random vehicles, the algorithm can only find a portion of vehicle matches due to the very narrow Gaussian distribution after convergence. On the other hand, the right plot clearly shows that a very large portion of matches resulting from the algorithm were true matches, which explains the ability of precisely estimating the sensor offsets. This of course is a trade-off that can further be adapted. If we limit the convergence thresholds in the EM algorithm, one could achieve a vehicle registration sensitivity at the cost of more pairs of vehicles being falsely matched and consequently a higher offset estimation error.

Similar to the false negative detections, we also simulated false positives. From the recorded radar data, it can be observed that false positive detections occurred mostly in the presence of vehicles. It did for example occasionally happen that a large vehicle like a bus or a truck was wrongfully detected as two small vehicles. In the absence of traffic, false positive detections were extremely rare. To simulate this fact, we specifically generated false positive detections at both sensors around the already existing detections. Thus, we extracted a number of vehicles according to the chosen false positive rate and copied the data of those vehicles also adding a random time to the original timestamp with a minimum and maximum absolute difference of 2 s and 3 s, respectively. In order to examine the combination between both false negatives and positives, we additionally set the false negative rate to 0, 0.1 and 0.2. The sensor distance was fixed to 100 m. Figure 14 shows that the matching relevance was significantly higher than the sensitivity, although the false positive rate seemed to have a slightly more negative effect on the resulting relevance than the false negative rate.

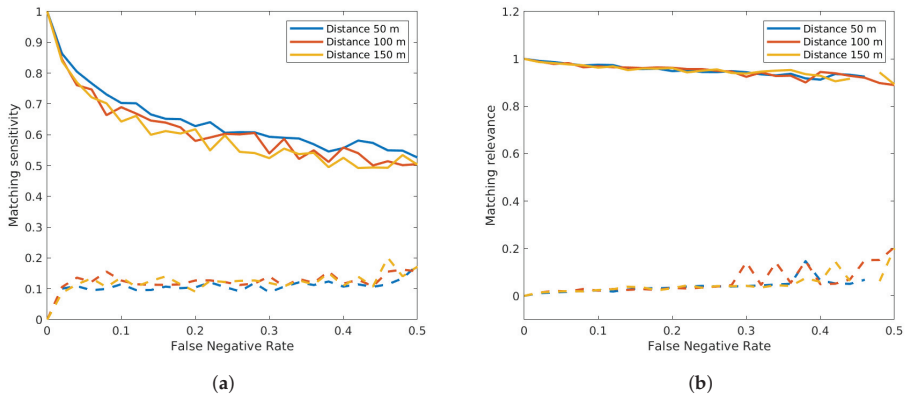


Figure 13. Influence of false negative detections on the matching sensitivity (a) and on the relevance (b). Continuous lines show the mean values, while dashed lines show the standard deviation of the result.

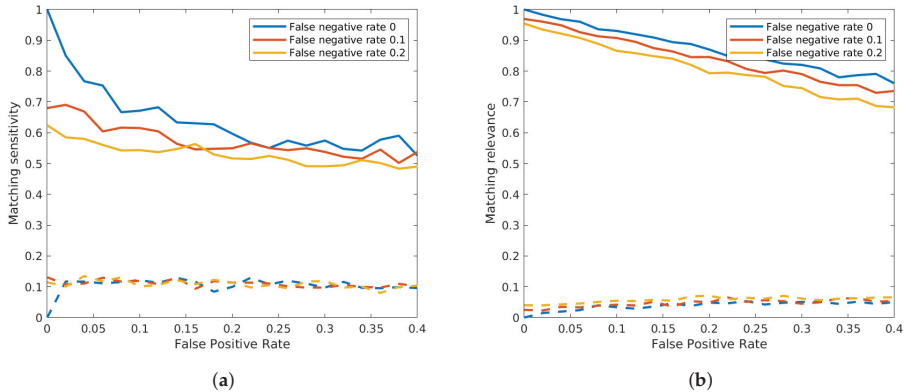


Figure 14. Influence of false positive detections on the matching sensitivity (a) and on the relevance (b). Continuous lines show the mean values, while dashed lines show the standard deviation of the result.

3.3. Real Data Experiment

In addition to validation with synthetic data, in this subsection, we want to demonstrate the capabilities of the proposed methods using real measurements at two cross-sections. In order to be able to validate device offset reconstruction, we used high-precision RTK-GNSS sensors for two purposes: to record the exact location of the radar devices with a precision of a few centimeters and to record the vehicles' paths along the curve. We were able to record the device cross-sections with a high precision with static GNSS data recording at the exact location of the radar devices. The GNSS data recorded when the sensor was mounted on the vehicle showed a high amount of jitter due to driving beneath the bridge seen in Figure 4, where satellite coverage was lost and could only be regained after 5–10 s, which limited our path reconstruction possibilities. Thus, instead of using the GNSS, our reference data of vehicle positions came from an optical distance measurement device mounted at the back of the vehicle (Figure 15). The sensor was based on laser Doppler velocimetry to measure the velocity and length of moving surfaces [28]. With the laser pointed downwards, the measured surface in this case was the road. The technique used had a very high accuracy, being capable of recording the distance and speed with an error less than $\pm 0.1\%$. Additionally, the laser was coupled with a control box, which enabled recording the data with a connected notebook. On the other hand, the control box also

provided the possibility of triggering a measurement reset either by a starter button or by an infrared light barrier. In the latter case, an additional laser device was directed sideways with reflecting markers being installed along the road. When the laser beam was reflected to the device, the measurement of the velocimeter was reset. We installed reflective markers at the location of the radar devices so that passing the relevant cross-sections triggered new distance measurements, and thus, we had the same coordinate system along the road section. At each new section between two radars, the record of the velocimeter started with zero.



Figure 15. Laser Doppler velocimeter used to generate reference data for the validation of the trajectory reconstruction.

For the validation of the reconstruction of microscopic traffic data, we drove with the equipped vehicle 33 times within three hours. For this period of time, we recorded both radar data and vehicle paths from the distance measurement device. As we saw in the previous subsection, our proposed algorithm was unable to converge to a good result when there were more than 25% outliers in the recorded data. When looking at Figure 4, we can see that the first radar device was located at a cross-section with three lanes, while the other devices only monitored the exit lane. This can also be verified in the radar data, as the first devices recorded 2928 vehicles in the same amount of time as the others recorded 444, 286, and 316 vehicles in the order of the devices. It is practically impossible to find feasible reconstruction between Radar 1 on the other devices. Thus, we validate the algorithm based on the data from Radar 2 and Radar 4. The distance between these two devices was 70 m along the curve, while all the devices were synchronized via Bluetooth using a handheld tablet, which connected to each radar device separately. The synchronization with the tablet was possible up to a 1-s deviation from the clock of the tablet. Additionally the radar devices were only capable of detecting vehicles with a resolution of 1 s.

We used the EM algorithm with both spatial and temporal optimization, as we could not guarantee a very precise time synchronization between the radar devices. The calculated offset after convergence was -0.63 s and 69.7 m between Radar 2 and 4, which matched very well with our measured reference. This indeed was in accordance with the results from the synthetic data validation. As the vehicles appeared randomly within the time windows of 1 s, the spatio-temporal offset error balanced out over many matched vehicle pairs. In other words, for a number of vehicle pairs that would correct the temporal offset towards the lower second value, there was a similar number of pairs that corrected the offset towards the higher second value. The results also delivered 274 matches between the two cross-sections. Moreover, from the 33 passes recorded with our equipped vehicle, we could successfully find 25 passes in the resulting matches. Figure 16 shows an outline of the reconstructed

trajectories, where we can see that even with the limited accuracy and resolution specifications of the recording devices used, very good trajectory reconstruction was feasible. To be more precise, the resulting mean RMS trajectory error was 3.48 m with a standard deviation of 2.08 m over these 25 trajectories. This result also fit our expectations as the vehicle passes showed speeds of 10–14 m/s. With a quantization of 1 s, the detection time error for a vehicle could be at most 0.5 s (either rounding up or down), but the mean RMS time error was about 0.3 s, which led to the found RMS error for the mentioned speed values.

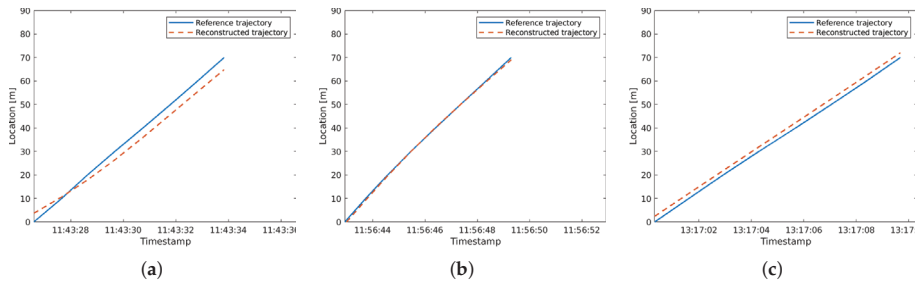


Figure 16. Longitudinal microscopic trajectory of the reference vehicle and the reconstructed data. The subplots (a–c) show three different vehicle passes with blue lines, while the red, dashed lines show the corresponding reconstructed trajectories (data recorded on the date of 30-November-2018).

4. Conclusions

This paper presented a method for spatio-temporal synchronization and microscopic traffic data reconstruction from cross-section-based detection devices. The required input data consisted of the timestamp and speed of individual vehicles. The three main problems that were been overcome were: unknown offset of the recording devices, unknown vehicle matches, and unknown vehicle trajectories. The proposed methods were based on the assumption of limited vehicle dynamics between the recording devices. It has been shown how under these assumptions, the iterative EM-algorithm can be used to register the individual detections of the devices and estimate the previously unknown offset simultaneously. In a further step, the registered datasets were used to reconstruct trajectory data from the devices by means of quintic Beziér curves.

After presenting the basic methodology, a comprehensive validation was presented to demonstrate the capabilities of the method. Using synthetically-generated datasets, different error possibilities, like location variance, speed measurement error, and outliers in the input data, were discussed, and their effect on the outcome of the method was thoroughly investigated. It was shown that errors in the offset estimation can be reduced by increasing the number of vehicles in the datasets, which led to a higher computational load. Thus, one must find a trade-off between these two parameters. Additionally, on the synthetic datasets, a validation based on empirical data was also conducted, where was shown that the algorithm not only precisely estimated the offsets of the used devices, but also accurately reconstructed vehicle trajectories.

Based on this work, further validation of the matching performance could be conducted by precisely recording individual vehicle matches. This could be done by using detectors based on number plate recognition. Furthermore, the applicability of the proposed methods should also be demonstrated by using other types of cross-section detectors like inductive loops and computer vision systems. In order to provide more information of the vehicle passes, the method could be applied in a multidimensional space to also reconstruct the lateral movements of vehicles rather than only longitudinal trajectories. Additionally, an iterative version of the method could be developed to be able to optimize continuously the parameters of the cross-sections with a continuously-growing data volume. Finally, a very interesting and promising extension of this work would be a thorough analysis on the possibility of overcoming the limitations regarding dynamic behavior. If the propagation of

shock waves in congested traffic is of interest, many dynamic maneuvers between cross-sections need to be taken into account. This could be done by incorporating driver modeling and microscopic traffic simulation in combination with data on the vehicle lengths in order to derive complex vehicle interactions.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing, original draft preparation, and writing, review and editing and visualization, A.F.; supervision, project administration, and funding acquisition, M.O.

Funding: This research is a part of the project “Highly automated tunnel surveillance for catastrophe management and regular operation” (Grant Number 13N13874) funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung (BMBF)). The APC was funded by the Post-Grant-Fund of the BMBF. This method was adapted and used within the project “Basic Evaluation for Simulation-Based Crash-Risk-Models: Multi-Scale Modeling Using Dynamic Traffic Flow States” (project number 280497386) funded by the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG)).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|----------|--|
| V2V | Vehicle-to-Vehicle |
| V2I | Vehicle-to-Infrastructure |
| EM | Expectation Maximization |
| RTK-GNSS | Real-Time Kinematic Global Navigation Satellite System |
| FNR | False-Negative-Rate |
| RMS | Root Mean Square |

References

1. Azevedo, C.L.; Cardoso, J.L.; Ben-Akiva, M.E. Probabilistic safety analysis using traffic microscopic simulation. *arXiv Preprint* **2018**, arXiv:1810.04776.
2. Bommers, M.; Fazekas, A.; Volkenhoff, T.; Oeser, M. Video Based Intelligent Transportation Systems—State of the Art and Future Development. *Transp. Res. Procedia* **2016**, *14*, 4495–4504. [[CrossRef](#)]
3. Fazekas, A.; Hennecke, F.; Kalló, E.; Oeser, M. A novel surrogate safety indicator based on constant initial acceleration and reaction time assumption. *J. Adv. Transp.* **2017**, *2017*. doi:10.1155/2017/8376572. [[CrossRef](#)]
4. Abdel-Aty, M.; Shi, Q.; Wang, L.; Wu, Y.; Radwan, E.; Zhang, B. *Integration of Microscopic Big Traffic Data in Simulation-Based Safety Analysis*; TRB: Washington, WA, USA, 2016.
5. Gora, P.; Rüb, I. Traffic models for self-driving connected cars. *Transp. Res. Procedia* **2016**, *14*, 2207–2216. [[CrossRef](#)]
6. Liu, C.; Kochenderfer, M.J. Analytically Modeling Unmanaged Intersections with Microscopic Vehicle Interactions. *arXiv Preprint* **2018**, arXiv:1804.04746.
7. Baek, S.; Liu, C.; Watta, P.; Murphey, Y.L. Accurate vehicle position estimation using a Kalman filter and neural network-based approach. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
8. Milanés, V.; Naranjo, J.E.; González, C.; Alonso, J.; de Pedro, T. Autonomous vehicle based in cooperative GPS and inertial systems. *Robotica* **2008**, *26*, 627–633. [[CrossRef](#)]
9. Pink, O.; Moosmann, F.; Bachmann, A. Visual features for vehicle localization and ego-motion estimation. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi’an, China, 3–5 June 2009; pp. 254–260.
10. Wei, L.; Cappelle, C.; Ruichek, Y. Camera/Laser/GPS Fusion Method for Vehicle Positioning Under Extended NIS-Based Sensor Validation. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 3110–3122. [[CrossRef](#)]
11. Coifman, B.; Kim, S. Speed estimation and length based vehicle classification from freeway single-loop detectors. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 349–364. [[CrossRef](#)]
12. Qiu, T.Z.; Lu, X.Y.; Chow, A.H.; Shladover, S.E. Estimation of freeway traffic density with loop detector and probe vehicle data. *Transp. Res. Rec.* **2010**, *2178*, 21–29. [[CrossRef](#)]

13. Aoude, G.S.; Desaraju, V.R.; Stephens, L.H.; How, J.P. Behavior classification algorithms at intersections and validation using naturalistic data. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 601–606.
14. Felguera-Martín, D.; González-Partida, J.T.; Almorox-González, P.; Burgos-García, M. Vehicular traffic surveillance and road lane detection using radar interferometry. *IEEE Trans. Veh. Technol.* **2012**, *61*, 959–970. [[CrossRef](#)]
15. Fuerstenberg, K.C.; Hipp, J.; Liebram, A. A Laserscanner for detailed traffic data collection and traffic control. In Proceedings of the 7th World Congress on Intelligent Systems, Turin, Italy, 6–9 November 2000.
16. Coifman, B.A.; Lee, H. *LIDAR Based Vehicle Classification*; Purdue University: West Lafayette, IN, USA, 2011.
17. Chen, S.; Sun, Z.; Bridge, B. Automatic traffic monitoring by intelligent sound detection. In Proceedings of the IEEE Conference on ITSC'97 Intelligent Transportation System, Boston, MA, USA, 12 November 1997; pp. 171–176.
18. Duffner, O.; Marlow, S.; Murphy, N.; O'Connor, N.; Smeanton, A. Road traffic monitoring using a two-microphone array. In *Audio Engineering Society Convention 118*; Audio Engineering Society: New York, NY, USA, 2005.
19. Antoniou, C.; Balakrishna, R.; Koutsopoulos, H.N. A synthesis of emerging data collection technologies and their impact on traffic management applications. *Eur. Transp. Res. Rev.* **2011**, *3*, 139–148. [[CrossRef](#)]
20. Behrendt, R. Traffic monitoring radar for road map calculation. In Proceedings of the 2016 17th International Radar Symposium (IRS), Krakow, Poland, 10–12 May 2016; pp. 1–4.
21. Zhao, H.; Sha, J.; Zhao, Y.; Xi, J.; Cui, J.; Zha, H.; Shibasaki, R. Detection and tracking of moving objects at intersections using a network of laser scanners. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 655–670. [[CrossRef](#)]
22. Buch, N.; Velastin, S.A.; Orwell, J. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 920–939. [[CrossRef](#)]
23. Hanif, A.; Mansoor, A.B.; Imran, A.S. Performance Analysis of Vehicle Detection Techniques: A Concise Survey. In *World Conference on Information Systems and Technologies*; Springer: London, UK, 2018; pp. 491–500.
24. Abdulrahim, K.; Salam, R.A. Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. *Int. J. Appl. Eng. Res.* **2016**, *11*, 713–726.
25. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–38. [[CrossRef](#)]
26. Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [[CrossRef](#)]
27. Marsh, D. *Applied Geometry for Computer Graphics and CAD*; Springer: London, UK, 2006.
28. Truax, B.E.; Demarest, F.C.; Sommargren, G.E. Laser Doppler velocimeter for velocity and length measurements of moving surfaces. *Appl. Opt.* **1984**, *23*, 67–73. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

A Virtual In-Cylinder Pressure Sensor Based on EKF and Frequency-Amplitude-Modulation Fourier-Series Method

Qiming Wang ^{1,*}, Tao Sun ¹, Zhichao Lyu ² and Dawei Gao ¹

¹ School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

² School of Automotive Studies, Tongji University, Shanghai 201804, China

* Correspondence: wangqm@usst.edu.cn; Tel.: +86-138-1717-0434

Received: 29 May 2019; Accepted: 10 July 2019; Published: 15 July 2019

Abstract: As a crucial and critical factor in monitoring the internal state of an engine, cylinder pressure is mainly used to monitor the burning efficiency, to detect engine faults, and to compute engine dynamics. Although the intrusive type cylinder pressure sensor has been greatly improved, it has been criticized by researchers for high cost, low reliability and short life due to severe working environments. Therefore, aimed at low-cost, real-time, non-invasive, and high-accuracy, this paper presents the cylinder pressure identification method also called a virtual cylinder pressure sensor, involving Frequency-Amplitude Modulated Fourier Series (FAMFS) and Extended-Kalman-Filter-optimized (EKF) engine model. This paper establishes an iterative speed model based on burning theory and Law of energy Conservation. Efficiency coefficient is used to represent operating state of engine from fuel to motion. The iterative speed model associated with the throttle opening value and the crankshaft load. The EKF is used to estimate the optimal output of this iteration model. The optimal output of the speed iteration model is utilized to separately compute the frequency and amplitude of the cylinder pressure cycle-to-cycle. A standard engine's working cycle, identified by the 24th order Fourier series, is determined. Using frequency and amplitude obtained from the iteration model to modulate the Fourier series yields a complete pressure model. A commercial engine (EA211) provided by the China FAW Group corporate R&D center is used to verify the method. Test results show that this novel method possesses high accuracy and real-time capability, with an error percentage for speed below 9.6% and the cumulative error percentage of cylinder pressure less than 1.8% when A/F Ratio coefficient is setup at 0.85. Error percentage for speed below 1.7% and the cumulative error percentage of cylinder pressure no more than 1.4% when A/F Ratio coefficient is setup at 0.95. Thus, the novel method's accuracy and feasibility are verified.

Keywords: in-cylinder pressure identification; speed iteration model; EKF; frequency modulation; amplitude modulation

1. Introduction

The development of cleaner and more efficient engines requires the continuous measurement of in-cylinder pressure which is of fundamental importance for combustion optimization, air-fuel ratio control, noise and pollutant reduction. More specifically, cylinder pressure is a fundamental intermediate variable that indicates engine state and drives models for engine combustion control. High-bandwidth control of ignition and the air-fuel ratio may permit these engines to operate reliably near the lean-burn limit, with significant improvements in efficiency and decreased emission of CO and NO_x. Fully utilizing this technology requires a key factor: real-time cylinder pressure [1]. The goal of Rizzoni's [2] method is to build a stochastic model for the combustion pressure process in the

spark-ignition engine. Control of torque balance can improve drivability and can suppress noise from light-duty diesel engines, but as a crucial part of the control algorithm, obtaining pressure data necessitates expensive pressure sensors and demands considerable computational time [3]. The advent of cylinder-pressure transducers seems promising, boasting a better approach to detect the in-cylinder pressure; however, its reliability, complex installation, high cost, and short working lifespan are full of controversy. As new technologies improve, such as computational methods, reconstructing cylinder pressure in a multi-cylinder internal combustion (IC) engines becomes more feasible. Bennett [4] proposed a robust adaptive-gradient descent-trained NARX neural network using both crank velocity and crank acceleration as inputs to reconstruct cylinder pressure in multi-cylinder IC engines. Another work has shown comprehensive results from an experimental assessment of a common rail diesel engine operated with neat diesel fuel and with ethanol substitutions [5]. With different substitutions and different substitution rates, pressure varies rapidly. Researchers have pursued other various approaches to develop a comparatively optimal way to obtain the desired cylinder pressure, directly or indirectly. Rizvi [6] proposed a novel method to detect engine misfire faults based on a hybrid model of the gasoline engine. As this method mainly employs cylinder pressure, closed-loop [7] combustion control [8] becomes feasible and misfire can be detected in multiple ways. However, these approaches vary in cost, reliability, robustness, accuracy, and convenience. Therefore, a low-cost [9] noninvasive soft-pressure sensor with a high level of reliability and accuracy is necessary. Payri [10] presented a step-by-step approach to optimize the signal processing both for offline and online applications based on the characteristics of the signal. Maurya [11] proposed that a method based on standard deviation of pressure, and pressure rise rate is used to find the minimum number of engine cycles to be recorded for averaging to get reasonably accurate pressure data independent of cyclic variability. Bhatti [12] proposed that two robust second-order sliding mode observers are employed that require two state mean value engine models based on inlet manifold pressure and rotational speed dynamics. The design of the second-order sliding mode observer depends largely on the model accuracy. And, more importantly, this method had the problems of chattering and large initial error. As a result, the tracking error of the steady-state speed was only about 4% and required a certain convergence time. Ferrari [13] proposed that pressure of nozzle opening and closing was identified by means of pressure data fitted by bench test. Then the time-frequency analysis was used to obtain the mean instantaneous frequency and adjust the control strategy of the injector. However, this method did not process data effectively, such as filter, before fast Fourier time-frequency transform, so this may affect the accuracy. Eriksson [14] calculated the corresponding relationship between pressure value and crankshaft rotation angle. The error of peak identification based on ion current in (Eriksson, 2003, Figure 4) was still large. Comparing with the corresponding relationship proposed by Eriksson [14], the cylinder pressure time curve is presented in this paper. It is based on burning theory and Law of energy conservation, and then modified by Extended-Kalman-Filter-optimized (EKF) engine model, which has a higher stability.

Observer-based method [12–15] is proven to be a good way to predict the output of SISO system. Engine cylinder pressure signal can be used to characterize the engine combustion state [16,17]. Due to the influence of multiple variables, it is difficult to establish a precise physical model of cylinder pressure. However, by observing and analyzing the cylinder pressure signal, this signal is cyclical with variable frequency (associated with speed) and variable amplitude (associated with the peak combustion pressure). Meanwhile, the engine cylinder pressure signal has a delay characteristic, so it is necessary to predict the current cylinder pressure value by the input of the current moment.

In this paper, a novel method which can identify engine pressure data accurately will provide a spacious viewpoint and means to process signals, thereby achieving important information for practical engineering. This novel method involving Frequency-Amplitude Modulated Fourier Series (FAMFS) and Extended-Kalman-Filter-optimized (EKF) engine model for identifying the periodic signal with its variable frequency and amplitude is proposed. This cylinder pressure identification method is also called a virtual cylinder pressure sensor.

Figure 1 shows a flow chart of method. Crankshaft speed [18–20] plays an important role in identifying the pressure as well as other applications, especially in misfire detection [21]. Instantaneous speed fluctuations [22], crankshaft segment acceleration, and transient rotational speed are the most widely utilized variables to locate misfire. Crank speed can also reveal and identify real-time engine combustion parameters [20], which is adopted in this paper. Speed data is affected by measurement noise from the crankshaft and process noise from the engine. These noise sources result in a huge accumulation error during pressure measurements, and EKF is chosen to optimize prediction and filtering [23,24].

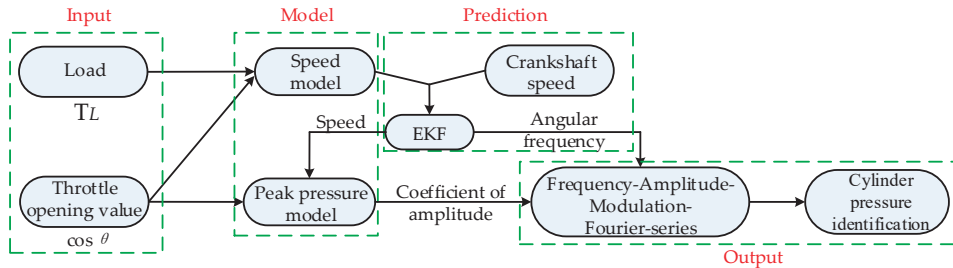


Figure 1. Flow chart of method.

The crankshaft speed in one solution step is related to the speed of the last step, so the speed model is an iterative process. Firstly, a novel speed iteration model based on burning theory and Law of energy Conservation is proposed. This iteration model is associated with the throttle opening value and the load. The EKF is then used to estimate the optimal output of the speed iteration model, and this optimal output can then be used for computing the frequency and amplitude of the cylinder pressure cycle-to-cycle. Secondly, taking Otto cycle of standard four-stroke engine as an example, a 24th order Fourier series is chosen to fit the standard working cycle. Thirdly, associated with the variable frequency and amplitude, a frequency-amplitude modulation method is adopted to modulate the standard pressure model identified in step two. Finally, system performance is evaluated by an actual working engine. Therefore, the main contributions of this paper can be summarized as follows:

- (1) Aimed to identify in-cylinder pressure and its periodic signals with variable frequency and amplitude, a virtual pressure sensor for engine healthy monitoring with low-cost, real-time, non-intrusive, a long-lifespan, and high-reliability in severe working conditions is presented.
- (2) A novel speed model is established according to burning theory, Law of energy Conservation and EKF. The FAMFS method is next developed to fit the periodic signals with variable frequency and variable amplitude.
- (3) The proposed method can be applied to multi-cylinder internal combustion (IC) engines including four-cylinder engines.

In this paper, on the one hand, as the key parameter of cylinder pressure identification, the speed iterative model using the EKF has been predicted with high accuracy. However, the overall deviation came from the calculation of pressure due to the unpredictable process of power stroke. On the other hand, due to only related to the crankshaft speed and the amount of air and fuel injection in engine, this method can be applied to multi-cylinder internal combustion (IC) engines including four-cylinder engines. More specifically, many key parameters of ignition engine such as spark advance angle (SAA) need to be considered in the process of building virtual in-cylinder pressure sensor model. Accordingly, at present, the application of the virtual sensor is only limited to spark ignition engines.

2. Crankshaft Speed Model

2.1. Design of Speed Iteration Model

As generally known, the angular acceleration of crankshaft in one cycle is produced by the output power of the engine, and the output energy per cycle is determined by the amount of air and fuel injected into engine. In internal combustion engines, the amount of air and fuel trapped in the cylinder is relatively influenced by engine speed through the volumetric efficiency, and mainly depends on the engine load. The change of engine load will affect crankshaft speed; hence the amount of air and fuel is deemed relating to the speed. Therefore, it forms a closed-loop iteration process. The crankshaft speed model (CSM) is established in two parts: constant load (Part A) and variable load (Part B). In Part A, the amount of air and fuel is related to the speed of the previous cycle. The power produced through fuel burning compels the crankshaft to rotate; and in Part B, the engine overcomes the load with the price of crankshaft speed reduction. The complete model is then forged by the combination of these two parts.

2.1.1. Constant Load Condition

In the following process, the load is constant. Cylinder pressure is directly related to the amount of fuel and air injection in engine. Thus, the air volume is equal to the engine displacement. The air mass flow of the intake manifold can be calculated as

$$\dot{m}_{air} = \frac{S_{eff}C_qC_mP_{im}}{\sqrt{T_{im}}} \quad (1)$$

where, S_{eff} is the active area of throttle valve; C_q is the flow coefficient; C_m represents the quality factor, associating with the engine. Once the engine model is specified, these parameters such as S_{eff} , C_q and C_m are determined. T_{im} and P_{im} are the temperature and pressure of intake manifold, respectively.

According to the actual working conditions, air mass is then obtained by Refs. [25–28]:

$$m_{air} = A(\varepsilon) \int_0^t \dot{m}_{air} dt \approx \frac{\pi A(\varepsilon) \dot{m}_{air}}{N} = \frac{\pi A(\varepsilon) S C_q C_m P_{im}}{N \sqrt{T_{im}}} (1 - \cos \theta) \quad (2)$$

where, the value for ε is the engine volumetric coefficient and considered constant when the crankshaft speed does not dramatically change. The integration of air mass flow will inevitably produce a constant term. The sum of constant term and non-constant term is air mass. Since there must be a proportional relationship between the constant term and the non-constant term, in order to facilitate the subsequent calculation, the proportional relationship is defined as $A(\varepsilon)$, so it is considered to be the engine parameter coefficient related to ε . Once the engine model is specified, these two parameters are determined. In addition, $t = \frac{\pi}{N}$, and t is opening time of intake valve, and N is the average value of speed in one working cycle.

Different fuels and different fuel ratios will strongly affect the combustion state of the engine [5]. The main elements in the fuel are oxygen, hydrogen, and carbon, with other elements being negligible. Thus, the relationship between mass fractions for the oxygen element w_o , hydrogen element w_h , and carbon element w_c in 1 kg of fuel are shown below:

$$w_c + w_h + w_o = 1 \quad (3)$$

The theoretical air volume for 1 kg of fuel that is entirely burned is shown in Equation (4):

$$L_O = \frac{22.4}{0.21} \left(\frac{w_C}{12} + \frac{w_H}{4} - \frac{w_O}{32} \right) \quad (4)$$

The engine output power is determined by the calorific value of the gas mixture, which can be defined as (considering 1 kg of fuel)

$$Q_{mix} = \frac{h_u}{22.4 \times (\lambda \frac{L_O}{22.4} + \frac{1}{M_\tau})} \tag{5}$$

where, λ is the excess-air factor; h_u is the low calorific value of the fuel; and M_τ is the relative molecular mass of fuel. Therefore, the power output related to the throttle valve is defined as

$$Q = \left[(L_O \rho_{air} + 1) \frac{m_{air}}{4 \rho_{fuel} L_O \rho_{air}} \right] \frac{h_u}{22.4 \times (\lambda \frac{L_O}{22.4} + \frac{1}{M_\tau})} \tag{6}$$

where, ρ_{air} and ρ_{fuel} is the density of air and fuel, respectively. By substituting Equation (2) into Equation (6), Equation (6) then becomes

$$Q = \left[(L_O \rho_{air} + 1) \frac{1}{4 \rho_{fuel} L_O \rho_{air}} \frac{\pi A(\varepsilon) h_u}{(\lambda L_O + \frac{22.4}{M_\tau})} \frac{S C_q C_m P_{im}}{\sqrt{T_{im}}} \right] \frac{(1 - \cos \theta)}{N} = B \frac{(1 - \cos \theta)}{N} \tag{7}$$

where,

$$B = (L_O \rho_{air} + 1) \frac{1}{4 \rho_{fuel} L_O \rho_{air}} \frac{\pi A(\varepsilon) h_u}{(\lambda L_O + \frac{22.4}{M_\tau})} \frac{S C_q C_m P_{im}}{\sqrt{T_{im}}} \tag{8}$$

As widely known, the total energy produced through burning cannot be completely converted to useful work. The process of energy transfer is shown in Figure 2.

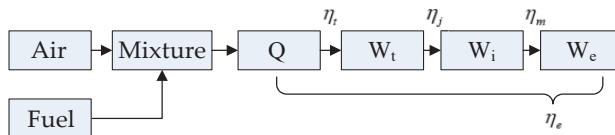


Figure 2. Flowchart of Engine energy transfer.

η_t is the theoretical thermal efficiency; η_j is the loss coefficient; η_m is the mechanical efficiency; and η_e is the engine efficiency.

In a four-cylinder four-stroke engine, each cylinder burns once per cycle, and the crankshaft is rotated 720 degrees in one working cycle.

$$W_e = U_{kin}(\varphi) + U_{pot}(\varphi) = \frac{J}{2} N^2 \tag{9}$$

where, $U_{kin}(\varphi)$ is the kinetic energy of the system; $U_{pot}(\varphi)$ is the potential energy of the system; J is the rotational inertia parameter (in this paper, it is equal to a flywheel’s rotational inertia parameter [24]); and φ is the crankshaft angle. By combining Equations (7)–(9), the CSM for Part A is shown in the following equation, and the relationship between $N_{k|k}$ and $N_{k|k+1}$ is shown in Figure 3.

$$\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}} = \frac{J}{2} N_{k|k+1}^2 \tag{10}$$

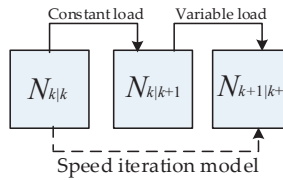


Figure 3. Relationship of speed in the iteration model.

2.1.2. Variable Load Condition

The change of engine load will affect crankshaft speed. This signal is discretized by computing the mean value of crank speed in power stroke. Since the time step of power stroke is enough small, Equations (11) and (12) are approximately satisfied:

$$T - T_L = J \frac{dN}{dt} = J \frac{N_{k+1|k+1} - N_{k|k+1}}{t_{cyc}} \Rightarrow N_{k+1|k+1} = \frac{T - T_L}{J} t_{cyc} + N_{k|k+1} \tag{11}$$

$$t_{cyc} = \frac{1}{\frac{2N_{k|k+1}}{60}} = \frac{30}{N_{k|k+1}} \tag{12}$$

where, J is the rotational inertia; T_L is the load torque; and t_{cyc} is the working cycle time. T is the engine output torque and can be obtained by the empirical model [25], and its coefficients are obtained by calibrating the actual engine.

$$T = -181.3 + 379.36m_{air} + 21.91R_{A/F} - 0.85R_{A/F}^2 + 0.26\sigma + 0.0028\sigma^2 + 0.027N_{k|k+1} - 0.000107N_{k|k+1}^2 + \dots + 0.00048N_{k|k+1}\sigma + 2.55\sigma m_{air} - 0.05\sigma^2 m_{air} \tag{13}$$

where, m_{air} is the mass of air in cylinder for combustion (g); σ is the spark advance angle (SAA).

In this case, only the load T_L is variable, thereby satisfying Equation (14).

$$T = C + DB \frac{1 - \cos \theta}{N_{k|k}} + 0.027N_{k|k+1} - 0.000107N_{k|k+1}^2 + 0.00048N_{k|k+1}\sigma \tag{14}$$

where $C = -181.3 + 21.91R_{A/F} - 0.85R_{A/F}^2 + 0.26\sigma + 0.0028\sigma^2$, $D = 379.36 + 2.55\sigma - 0.05\sigma^2$.

From this analysis, the model for Part B can be described by:

$$N_{k+1|k+1} = \frac{C + DB \frac{1 - \cos \theta}{N_{k|k}} + 0.027N_{k|k+1} - 0.000107N_{k|k+1}^2 + 0.00048N_{k|k+1}\sigma - T_L}{J} \cdot \frac{30}{N_{k|k+1}} + N_{k|k+1} \tag{15}$$

By combining Equations (12) and (15), the speed iteration model can then be established:

$$N_{k+1|k+1} = \frac{C + DB \frac{1 - \cos \theta}{N_{k|k}} + 0.027 \sqrt{\frac{2\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}}}{J}} - 0.000107 \frac{2\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}}}{J} + 0.00048 \frac{2\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}}}{J} - \sigma - T_L}{J} \cdot \frac{30}{\frac{2\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}}}{J}} + \frac{2\eta_e B \frac{(1 - \cos \theta)}{N_{k|k}}}{J} \tag{16}$$

Equation (16) can be expressed as

$$N_{k+1|k+1} = g(N_{k|k}, \theta) \Rightarrow N_{k+1} = g(N_k, \theta) \tag{17}$$

where, $g(\cdot)$ is a nonlinear function.

The relationship among $N_{k|k}$, $N_{k|k+1}$, and $N_{k+1|k+1}$ is depicted in Figure 3.

2.2. Design of Extended Kalman Filter

Under actual working conditions, the throttle opening value can be decomposed into two parts: an ideal opening value and throttle position signal noise mainly due to EMI. In measuring the crankshaft speed process, speed data is affected by measurement noise from the crankshaft and process noise from the engine. Engine vibration caused by engine knock, etc., is transmitted to the crankshaft. In addition, the disturbance caused by road surface irregularity is also gradually transmitted to crankshaft. According to Kalman filter theory, these fluctuations are considered as process noise and need to be processed. These noise sources result in a huge accumulation error during pressure measurements, and the Kalman filter (KF) [29,30] is the most preferable filtering method for measurement and process noise. The KF is used to predict an optimal output for the current step based on the optimal output of the previous step and the observed value of the current step. However, for nonlinear systems, the KF cannot achieve optimal prediction. Therefore, the Extended Kalman filter (EKF) [31–33] is proposed by using the local linear property of nonlinear systems. State prediction is accomplished by using EKF gain to update state and error covariance. After linearization by computing the nonlinear function's first order Taylor series and Jacobian matrix at the operating point, the EKF algorithm matches the KF and it comprises two parts: time update and state update. A discrete-time state-space model with a zero mean and a Q_k variance in the white process noise (w_{k-1}) is shown as

$$X_k = g(X_{k-1}) + w_{k-1} \quad (18)$$

where, X_k is the state vector of the k th step, and $g(\cdot)$ is a nonlinear function.

The state observation can be written as

$$Z_k = h(x_k) + v_k \quad (19)$$

where v_k is the white measurement noise with a zero mean; R_k is the variance; and $h(\cdot)$ is a nonlinear function.

The prediction is shown below:

$$\hat{X}_{k|k-1} = g(X_{k-1}) \quad (20)$$

The first-order Taylor series for $g(\cdot)$ at $\hat{X}_{k-1|k-1}$ and $h(\cdot)$ at $\hat{X}_{k|k-1}$ are obtained as shown in Equations (21) and (22), respectively.

$$G_k = \left. \frac{\partial g}{\partial X} \right|_{\hat{X}_{k-1|k-1}} = g(\hat{X}_{k-1|k-1}) + \left. \frac{\delta g(X_{k-1})}{\delta X_{k-1}} \right|_{X_{k-1}=\hat{X}_{k-1|k-1}} (X_{k-1} - \hat{X}_{k-1|k-1}) \quad (21)$$

$$H_k = \left. \frac{\partial h}{\partial X} \right|_{\hat{X}_{k|k-1}} = h(\hat{X}_{k|k-1}) + \left. \frac{\delta h(X_k)}{\delta X_k} \right|_{X_k=\hat{X}_{k|k-1}} (X_k - \hat{X}_{k|k-1}) + v_k \quad (22)$$

Let:

$$\Delta_k = X_k - \hat{X}_k \quad (23)$$

Defining P_{k+1} as the covariance of Δ_{k+1} , the transcendental error covariance can be calculated as

$$P_{k+1|k} = G_k P_k G_k^T + Q_k \quad (24)$$

The major purpose of the EKF is to obtain a minimum-variance state estimate. The gain matrix of the EKF is computed subject to minimization of the estimation error covariance.

$$Tr \left[P_{k+1}^x \right]_{K_{k+1}} = \min! \Rightarrow K_{k+1} = P_{k+1|k} H_k^T (H_k^T P_{k+1|k} H_k^T + R)^{-1} \quad (25)$$

State prediction is accomplished by using EKF gain to update state and error covariance.

$$\hat{X}_{k+1} = \hat{X}_{k+1|k} + K_{k+1} (Z_k - H_k \hat{X}_{k+1|k}) \quad (26)$$

$$P_{k+1} = (E - K_{k+1}H)P_{k+1|k} \tag{27}$$

The observed vector is written as follows, where the observed matrix $H = 1$:

$$Z_k = H_k x_k + v_k \tag{28}$$

3. Calculation of Cylinder Pressure

In this paper, due to only related to the crankshaft speed and the amount of air and fuel injection in engine, this method can be applied to multi-cylinder internal combustion (IC) engines including four-cylinder engines. More specifically, many key parameters of ignition engine such as spark advance angle (SAA) need to be considered in the process of building virtual in-cylinder pressure sensor model. Accordingly, at present the application of the virtual sensor is only limited to spark ignition engines. In addition, now there are many ways to describe the status of engine working cycle such as Otto-cycle, Diesel-cycle, Sabtache-cycle, and Atkinson-cycle. Among the many cycles, the Otto-cycle optimally depicts the four-stroke ignition combustion engine. Taking Otto cycle of standard four-stroke engine as an example, but the virtual sensor method is not limited to the engine described by Otto cycle.

In this cycle, the crankshaft rotates 720° as one working period, where the 720° can be divided into four sub-cycles each with a crankshaft rotation of 180° [34]. A diagram of the standard working cycle is shown in Figure 4. These four smaller cycles are known as: intake stroke, compression stroke, power stroke, and exhaust stroke. According to the Otto-cycle, the cylinder pressure during the intake stroke and the exhaust stroke is equal to atmosphere. The compression stroke and the expanding process are considered as isentropic, and the burning process is regarded as constant volume combustion [35].

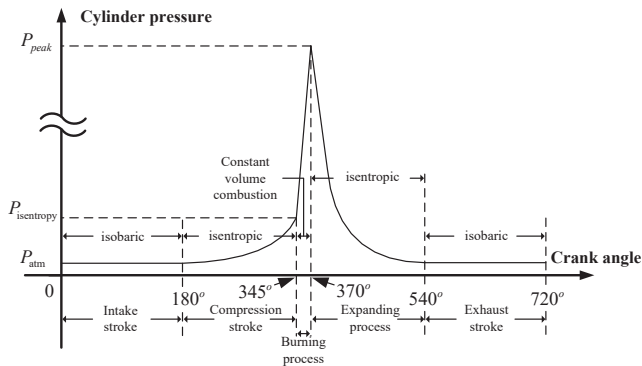


Figure 4. Diagram of the standard working cycle.

The pressure at the end of compression stroke is also the initial pressure of the burning process. The peak occurs at the end of the burning process and is known as the peak pressure of the cylinder. This pressure is an important variate to understand the combustion efficiency of the burning process and to monitor the state of the cylinder.

3.1. Cylinder Pressure Peak Time Confirmation

It's generally accepted that when the spark advance angle (SAA) is $10\text{--}15^\circ$ after top-dead-center (TDC), combustion efficiency becomes maximum, while engine vibration reaches a minimum [35]. In order to obtain the maximum power of engine output at Optimal-Spark-Advance-Angle (OSAA), the OSAA need to be adjusted continuously.

In the case of high speed and low load of engine, the OSAA should be increased. This is due to the delay period of fuel combustion in the cylinder. The faster the speed is, the bigger the OSSA.

On the contrary, with the speed increased, the stronger the turbulence formed by combustion gas in the cylinder is, the faster the combustion speed and the lower the advance angle is. combustion turbulence, which positively affects the burning speed, will increase with faster crank speeds. In this case, the corresponding OSAA delay period will be small and because of the delay caused by fuel burning in the power stroke, the OSAA should be slightly smaller. Additionally, in this paper, the pressure identification based on the power flow is adopted. The comparison between identified pressure signal and actual pressure signal can be realized by computing each power. Hence the SAA is deemed as constant. Due to the interaction and method selected, it's deemed that the OSAA is constant at 15°, and the maximum power output will occur when the crankshaft angle is 10° after the TDC where pressure in the cylinder reaches its peak value.

3.2. Computation of Cylinder Pressure Peak Value

After the burning process, the exhaust gases in cylinders is primarily composed of CO_2 , H_2O and N_2 . According to Equations (7) and (8), the specific heat capacity at constant volume for the burned gas is

$$C_{mix} \approx \left(\frac{\frac{w_C}{12}}{\frac{w_C}{12} + \frac{w_H}{8}} C_{CO_2} + \frac{1}{2} \cdot \frac{\frac{w_H}{8}}{\frac{w_C}{12} + \frac{w_H}{8}} C_{H_2O} \right) \cdot \frac{2}{9} + C_{N_2} \cdot \frac{7}{9} \quad (29)$$

where, C_{CO_2} is the specific heat capacity at a constant volume for CO_2 , and C_{H_2O} is that for H_2O .

As computed in Equation (30), Cylinder temperature varies with output power.

$$\Delta T = \frac{Q}{C_{mix}m} \quad (30)$$

where, m is the mass of the burned gas, and according to conservation of mass theory:

$$m = m_{air} + m_{fuel} \quad (31)$$

According to the ideal gas state equation,

$$pV = nRT \quad (32)$$

Since the burning process is deemed as isovolumetric, the peak pressure in cylinder is written as:

$$p_{peak} = \frac{nR \left(\frac{Q}{C_{mix}(m_{air} + m_{fuel})} + T_{comp} \right)}{V} \quad (33)$$

where, n is the amount of substance which can be gained from the coefficient of the fuel burning equation; V is the volume of cylinder at top dead center; T_{comp} is the temperature at the end of the compression stroke; and R is the ideal gas constant.

4. Modeling of in-Cylinder Pressure

As previously mentioned, the cylinder pressure signal is periodic with varying frequency and amplitude, FAMFS is proposed to modulate the cylinder pressure [36]. A high-factorial Fourier series with varying frequency and amplitude is adopted to modulate the cylinder pressure.

4.1. Frequency-Modulated Fourier-Series

If there is a periodic signal $f(x)$, then for every x there exists a positive value L which makes $f(x) = f(x + L)$ correct, where L is known as the fundamental period and $\omega_0 = \frac{2\pi}{L}$ is termed the fundamental frequency. The Fourier-series is written as [37]:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos nx \frac{2\pi}{L} + b_n \sin nx \frac{2\pi}{L} \right) \tag{34}$$

where, a_0 is the intercept and a_n, b_n are coefficients of the Fourier-series.

The frequency-modulated method [38] allows the instantaneous frequency of the carrier signal to vary with the change law of the delivery signal. This modulation can be classified as primary or secondary according to the category of the modulation effects.

With a defined carrier signal $x_c(t) = A \cos(2\pi f_c t)$ and a transmitted signal $y(t)$, the modulating signal can be written as:

$$x_c(t) = A \cos(2\pi \int_0^t [f_c + f_{\Delta} y(\tau)] d\tau) \tag{35}$$

where, f_c is the center frequency of the carrier signal; A is the amplitude; $f_c + f_{\Delta} y(\tau)$ is the instantaneous frequency; and f_{Δ} is the frequency deviation gain.

Based on the engineering practice, the 24-order FAMFS is selected. It should be noted that the identification process is completed offline, and the only online requirement is adjustment of the frequency and amplitude of the FAMFS. Thus, the instantaneity of method is well guaranteed. The 24th order FAMFS is shown below:

$$f(x) = Aa_0 + A \sum_{n=1}^{24} \left(a_n \cos n(2\pi \sum_{t_i=0}^t \frac{t}{N_n} y(t_i)) + b_n \sin n(2\pi \sum_{t_i=0}^t \frac{t}{N_n} y(t_i)) \right) \tag{36}$$

where, t is the total running time; N_n is the number of working cycles; and t_i is time sequence.

4.2. Pressure Model

The optimal output of the Kalman observer is the speed of each power stroke. For a four-cylinder engine, the cylinders work in a logical sequence, known as firing order. Generally, the firing order is 1-3-4-2. By sampling the optimal output of the EKF, according to the working sequence, the optimal power-stroke speed for each cylinder is obtained:

$$\begin{aligned} N_{num1} &= N_{1+kn} \\ N_{num2} &= N_{2+kn} \\ &\vdots \\ N_{numi} &= N_{i+kn} \end{aligned} \tag{37}$$

where, N_{numi} is the speed of the i th cylinder; N_{i+kn} is the optimal speed of $(i + kn)$ th cycle; and k is a natural number.

According to actual working conditions, the cylinder pressure signal can be deemed as a FAMFS [39,40] signal with a center frequency at zero and a unit gain of frequency deviation. Thus, the pressure identification is ultimately written as:

$$F(x) = A_k a_0 + P_{atm} + A_k \sum_{n=1}^{24} \left(a_n \cos n(2\pi \sum_{t_i=0}^t \frac{t}{N_n} y(t_i)) + b_n \sin n(2\pi \sum_{t_i=0}^t \frac{t}{N_n} y(t_i)) \right) \tag{38}$$

$$y(t_i) = \frac{\pi N_k}{15} \sum_{n=1}^k \frac{2\pi}{\omega_n} \leq t_i \leq \sum_{n=1}^{k+1} \frac{2\pi}{\omega_n} \tag{39}$$

$$A_k = \frac{nR(\frac{Q_k}{C_{mixk}(m_{airk} + m_{fuelk})} + T_{comp})}{V} \sum_{n=1}^k \frac{2\pi}{\omega_n} \leq t_i \leq \sum_{n=1}^{k+1} \frac{2\pi}{\omega_n} \tag{40}$$

where, t is time, and A_k is the theoretical pressure peak at the k th working cycle.

5. Validation and Results

When the engine is in normal combustion state, its air-fuel ratio usually exceeds 0.85. A ratio of 0.7 or 0.8 will make the burning process of inside cylinder unstable and unpredictable, and this condition only happens in a very special condition like startup in an extremely cold environment or engine malfunctioned. No matter in which condition, it’s not a normal working status so the prediction is not that useful. The validity of the proposed method is verified by comparing the identification value of the virtual cylinder pressure sensor with the measured value through the cylinder pressure sensor.

In the following section, data collected from a genuine 2.0-L four-stroke four-cylinder engine was provided by the FAW Group Corporation R&D Center (Changchun, China). The engine model is EA211 and produced by VW (Volkswagen) (Changchun, China). Data was collected over the course of 100 working cycles adopting Kistler Model 6052A Pressure Sensor to test the performance of the proposed method.

5.1. Set Air-Fuel Ratio Coefficient at 0.85

The parameters and setup of the engine are shown in Figure 5 and Table 1.

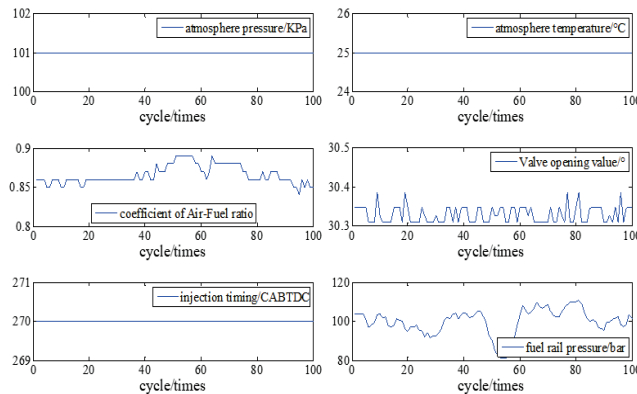


Figure 5. Parameters and setup of Engine at A/F ratio 0.85.

Table 1. The parameters and setup of the engine.

| A/F Ratio | Atmosphere Pressure/KPa | Atmosphere Temperature/°C | Valve Opening Value/° | CA BTDC | Fuel Rail Pressure /bar |
|-----------|-------------------------|---------------------------|-----------------------|---------|-------------------------|
| 0.85 | 101 | 25 | 30 | 270 | 100 |
| 0.95 | 101 | 25 | 30 | 270 | 90 |

Figure 6 compares the crankshaft speed outputs of the EKF with the measured results acquired from the genuine engine. As seen in Figures 6 and 7, from 30 cycles to 70 cycles the engine speed measured showed signs of rising and began to decline after reaching the maximum at 70 cycles. The EKF method accurately tracks this trend and its maximum percentage of error is only 9.6%. This shows acceptable agreement between identified and actual values.

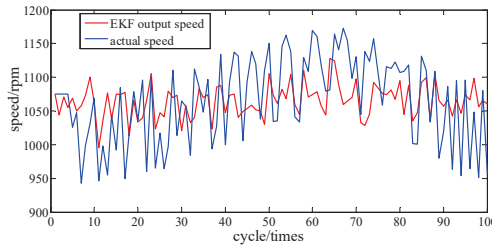


Figure 6. Speed tracking at A/F ratio 0.85.

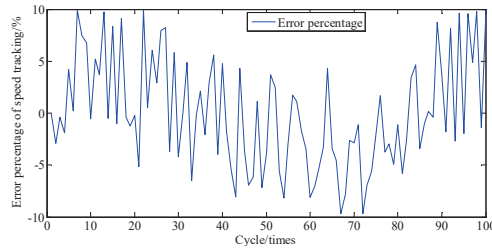


Figure 7. Error percentage of speed tracking at A/F ratio 0.85.

Once the optimal speed from the EKF engine model is predicted, the peak pressure of cylinder can be calculated. Furthermore, the 24th order FAMFS is adopted for identification purposes, and its parameters are shown in Table 2:

Table 2. Parameters of the 24th order Fourier series.

| a0~a6 | a7~a13 | a14~a20 | a21~a24, b1~a3 | b4~b10 | b11~b17 | b18~b24 |
|--------|--------|----------|----------------|--------|---------|---------|
| 12.06 | -2.639 | 0.324 | 0.3783 | 5.68 | -1.775 | 0.4077 |
| -11.92 | 0.1302 | 0.7365 | -0.3231 | -5.136 | 0.9967 | 0.2995 |
| -3.768 | 1.882 | -0.9149 | -0.0095 | 1.267 | 0.3929 | -0.5088 |
| 9.395 | -1.854 | 0.1323 | 0.257 | 1.927 | -1.146 | 0.2242 |
| -4.813 | 0.2843 | 0.5965 | 13.11 | -2.787 | 0.7361 | 0.202 |
| -1.215 | 1.142 | -0.5427 | -11.71 | 1.365 | 0.291 | -0.3184 |
| 3.652 | -1.264 | -0.00471 | 1.303 | 0.8418 | -0.8235 | 0.1462 |

A comparison of the actual cylinder pressure with the model-identified values is shown in Figure 8. It shows that in early stage of identification process, the identified values are basically in agreement with the measured values, whether in phase or amplitude. However, the phase deviation between measured and identified values begins to emerge and become larger with time. This is due to the cumulative error caused by the fitting error of the 24-order Fourier series and the speed tracking error.

For Figure 8, the area under the curve is multiplied by the piston area to represent the piston impulse. Hence, the cumulative error percentage of piston impulse is the cumulative error percentage of cylinder pressure. Figure 9 shows the curves of cumulative piston impulse including the actual and identification situation within 100 cycles. The two coincide basically. And Figure 10 shows that its cumulative error percentage is less than 1.8%. This agreement as well as the permitted errors, represents good performance from the proposed method.

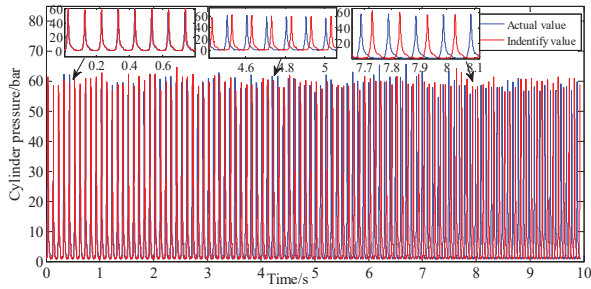


Figure 8. Pressure identification at A/F ratio 0.85.

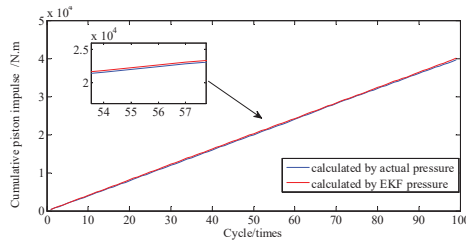


Figure 9. Piston impulse accumulated at A/F ratio 0.85.

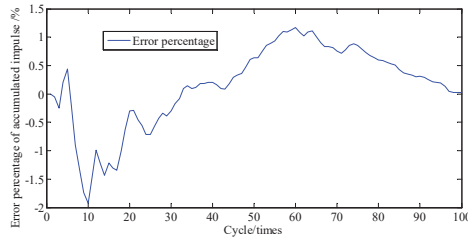


Figure 10. Error percentage of piston impulse accumulated.

In general, the engine is a complex and sophisticated system, and small changes in valve or atmospheric temperature, fuel rail pressure, vibration from irregularity, etc., can severely impact the state of the engine; thus, besides a direct comparison of pressure and the cumulative piston impulse, comparison of piston impulse per cycle is also needed. As depicted in Figure 11, the actual and identified IMEP of each cycle one by one are contrasted. Figure 12 shows that these two error percentages are below 5.4%. And then the proposed method is proven to possess a great ability to track the IMEP calculated by actual pressure.

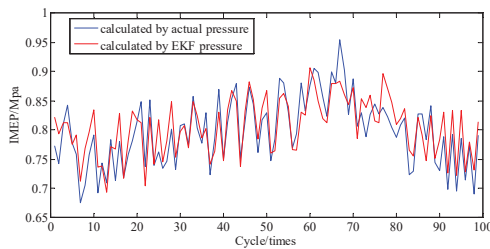


Figure 11. IMEP at A/F ratio 0.85 per cycle.

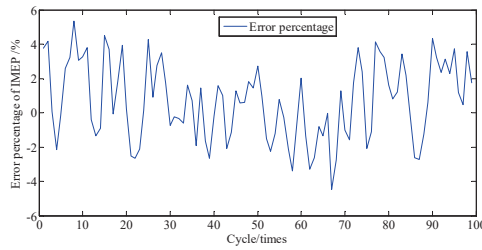


Figure 12. Error percentage of IMEP at A/F ratio 0.85 per cycle.

5.2. Set Air-Fuel Ratio Coefficient at 0.95

The parameters and setup of the engine are shown in Figure 13.

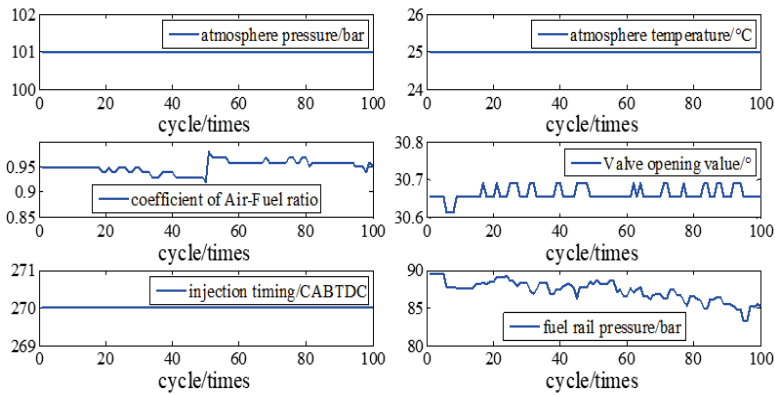


Figure 13. Parameters and setup of Engine at A/F ratio 0.95.

Figure 14 shows the comparing result of speed from EKF and a genuine engine at A/F ratio 0.95. Error percentages of the speed in Figure 15 is below 1.7%. Calculated and actual values have a great uniformity. Results show that proposed method possessed a great performance on tracking speed.

The peak pressure of cylinder can be calculated by using the predicted speed. Furthermore, the 24th order FAMFS is adopted for identification purposes, and its parameters are the same as Table 2. A comparison of the actual cylinder pressure with the model-identified values is shown in Figure 16.

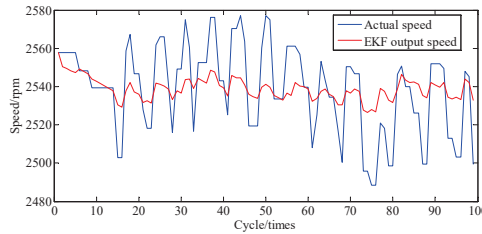


Figure 14. Speed tracking at A/F ratio 0.95.

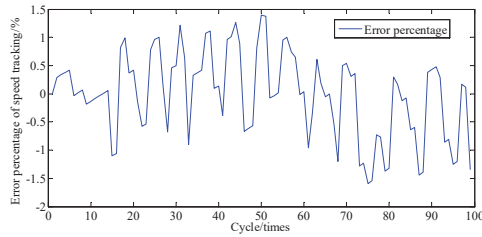


Figure 15. Error percentage of speed tracking at A/F ratio 0.95.

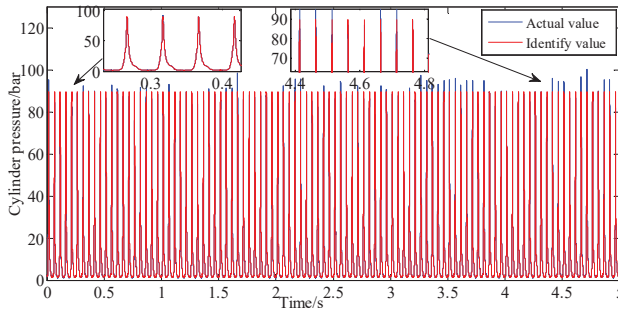


Figure 16. Pressure identification at A/F ratio 0.95.

Figure 16 shows that the comparing result of actual and identified cylinder pressure. Figure 17 also shows the curves of cumulative piston impulse including the actual and identification situation but at A/F ratio 0.95 within 100 cycles. The two coincide also basically. As depicted in Figure 18, the actual and identified IMEP of each cycle one by one are contrasted. Compared with Figures 8 and 16, the proposed method possesses a better ability to track the cylinder pressure at A/F ratio 0.95 than at A/F ratio 0.85. The closer the air-fuel ratio coefficient is to the optimal A/F ratio, the more predictable the combustion state of the engine is. This is also proved by the cumulative error percentage of piston impulse is less than 1.4% in Figure 19, and the error percentage of IMEP in each cycle is less than 4.9% in Figure 20. On the contrary, the phase difference of identified cylinder pressure began to emerge and become larger as the air-fuel ratio coefficient getting far away from optimal Air-Fuel ratio coefficient. But no matter how the A/F ratio coefficient changed, if it is within reasonable range, error percentage of cylinder pressure is within tolerance.

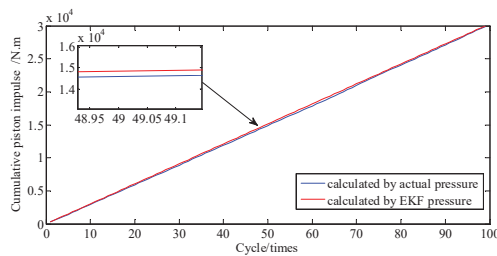


Figure 17. Piston impulse accumulated at A/F ratio 0.95.

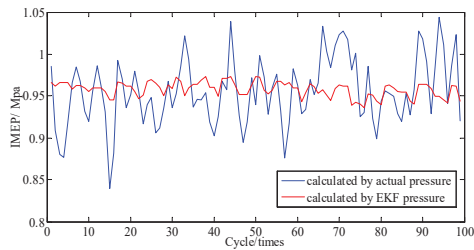


Figure 18. IMEP at A/F ratio 0.95 per cycle.

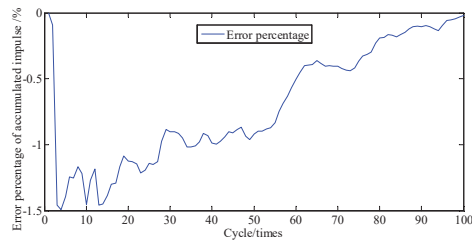


Figure 19. Error percentage of piston impulse accumulated.

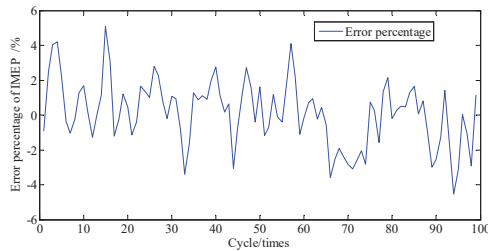


Figure 20. Error percentage of IMEP at A/F ratio 0.95 per cycle.

6. Conclusions

As a crucial and critical factor in monitoring the internal state of an engine, pressure identification is significantly essential. In this paper, aimed at solving problems associated with reliability and cost of invasive pressure sensors, virtual cylinder pressure identification sensor based on EKF and FAMFS is proposed. This new method employs three key steps:

- (1) An iterative speed model based on burning theory and Law of energy Conservation. Efficiency coefficient is used to represent operating state of engine from fuel to motion. The iterative speed model associated with the throttle opening value and the crankshaft load.
- (2) The EKF is used to estimate the optimal output of this iteration model. The optimal output of the speed iteration model is utilized to separately compute the frequency and amplitude of the cylinder pressure cycle-to-cycle.
- (3) A pressure fitting algorithm is established by a 24th-order FAMFS. With this process an approximate identification method for engine cylinder pressure is developed in a standard engine's working cycle.

To further verify the validity of the proposed method, data collected from a genuine engine (EA211) provided by the China FAW Group Corporate R&D Center was used. Test results demonstrate that the proposed method exhibits great performance for tracking crank speed and the engine's real-time cylinder pressure. By comparing the identified pressure outputs with measurement results,

the cumulative error percentage for cylinder pressure was below 1.8%, the error percentage of IMEP each cycle was no more than 5.4%, and the error associated with speed was less than 9.6%, when A/F Ratio coefficient was set to 0.85. However, the cumulative error percentage for cylinder pressure was below 1.4%, the error percentage of IMEP each cycle was no more than 4.9%, and the error associated with speed was less than 1.7%, when A/F Ratio coefficient was setup at 0.95. The effectiveness and instantaneity are thereby proven and shown to be capable of achieving the desired accuracy despite the uncertainty of the engine's power stroke. The closer the A/F ratio set to optimal A/F ratio, the more exact the identified pressure will be. Furthermore, it is important to note that the proposed method is valid, in general, and may be applied to more complex situations.

Author Contributions: Study design and investigation, Q.W. and T.S.; Methodology, Q.W.; Resources, Q.W.; Supervision, D.G.; Writing—original draft, Q.W.; Writing—review & editing, Q.W., T.S., Z.L., D.G.

Funding: The author(s) disclose receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (grant numbers 51806143) and sponsored by Shanghai Sailing Program (No. 19YF1434600).

Conflicts of Interest: The author(s) declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- Guardiola, C.; Pla, B.; Bares, P.; Stefanopoulou, A. Cylinder charge composition observation based on in-cylinder pressure measurement. *Measurement* **2019**, *131*, 559–568. [[CrossRef](#)]
- Rizzoni, G.A. A stochastic model for the indicated pressure process and the dynamics of the internal combustion engine. *IEEE Trans. Veh. Technol.* **1989**, *38*, 180–192. [[CrossRef](#)]
- Min, K.; Chung, J.; Sunwoo, M. Torque balance control for light-duty diesel engines using an individual cylinder IMEP estimation model with a single cylinder pressure sensor. *Appl. Therm. Eng.* **2016**, *109*, 440–448. [[CrossRef](#)]
- Bennett, C.; Dunne, J.F.; Trimby, S.; Richardson, D. Engine cylinder pressure reconstruction using crank kinematics and recurrently-trained neural networks. *Mech. Syst. Signal Process.* **2017**, *85*, 126–145. [[CrossRef](#)]
- Bodisco, T.; Brown, R.J. Inter-cycle variability of in-cylinder pressure parameters in an ethanol fumigated common rail diesel engine. *Energy* **2013**, *52*, 55–65. [[CrossRef](#)]
- Rizvi, M.A.; Bhatti, A.I.; Butt, Q.R. Hybrid model of the gasoline engine for misfire detection. *IEEE Trans. Ind. Electron.* **2011**, *58*, 3680–3692. [[CrossRef](#)]
- Al-Durra, A.; Canova, M.; Yurkovich, S. A real-time pressure estimation algorithm for closed-loop combustion control. *Mech. Syst. Signal Process.* **2013**, *38*, 411–427. [[CrossRef](#)]
- Cook, J.A.; Powell, B.K. Modeling of an internal combustion engine for control analysis. *IEEE Control Syst. Mag.* **1988**, *8*, 20–26. [[CrossRef](#)]
- Sellnau, M.C.; Matekunas, F.A.; Battiston, P.A.; Chang, C.-F.; Lancaster, D.R. Cylinder-Pressure-Based Engine Control Using Pressure-Ratio-Management and Low-Cost Non-Intrusive Cylinder Pressure Sensors. *SAE Technical Paper Series* **2000**, 899–918.
- Payri, F.; Luján, J.M.; Martín, J.; Abbad, A. Digital signal processing of in-cylinder pressure for combustion diagnosis of internal combustion engines. *Mech. Syst. Signal Process.* **2010**, *24*, 1767–1784. [[CrossRef](#)]
- Maurya, R.K. Estimation of optimum number of cycles for combustion analysis using measured in-cylinder pressure signal in conventional CI engine. *Measurement* **2016**, *94*, 19–25. [[CrossRef](#)]
- Ahmed, Q.; Bhatti, A.I.; Iqbal, M. Virtual sensors for automotive engine sensors fault diagnosis in second-order sliding modes. *IEEE Sens. J.* **2011**, *11*, 1832–1840. [[CrossRef](#)]
- Ferrari, A.; Paolicelli, F. A virtual injection sensor by means of time frequency analysis. *Mech. Syst. Signal Process.* **2019**, *116*, 832–842. [[CrossRef](#)]
- Eriksson, L.; Nielsen, L. Towards on-board engine calibration with feedback control incorporating combustion models and ion-sense. *Automatisierungstechnik* **2003**, *51*, 204–212. [[CrossRef](#)]
- Huo, X.; Ma, L.; Zhao, X.; Zong, G. Observer-based fuzzy adaptive stabilization of uncertain switched stochastic nonlinear systems with input quantization. *J. Frankl. Inst.* **2019**, *356*, 1789–1809. [[CrossRef](#)]

16. Zhao, X.; Wang, X.; Zhang, S.; Zong, G. Adaptive neural backstepping control design for a class of non-smooth nonlinear systems. *IEEE Trans. Syst. Man, Cybern. Syst.* **2018**, *1–12*. [[CrossRef](#)]
17. Balyts'kyi, O.I.; Abramek, K.F.; Shtoeck, T.; Osipowicz, T. Diagnostics of degradation of the lock of a sealing ring according to the loss of working gases of an internal combustion engine. *Mater. Sci.* **2014**, *50*, 156–159. [[CrossRef](#)]
18. Balyts'kyi, O.I.; Abramek, K.F.; Mruzik, M.; Stoeck, T.; Osipowicz, T. Evaluation of the losses of hydrogen-containing gases in the process of wear of pistons of an internal-combustion engine. *Mater. Sci.* **2017**, *53*, 289–294. [[CrossRef](#)]
19. Desbazeille, M.; Randall, R.B.; Guillet, F.; Badaoui, M.; Hoisnard, C. Model-based diagnosis of large diesel engines based on angular speed variations of the crankshaft. *Mech. Syst. Signal Process.* **2010**, *24*, 1529–1541. [[CrossRef](#)]
20. Tagliatalata, F.; Lavorgna, M.; Mancaruso, E.; Vaglieco, B.M. Determination of combustion parameters using engine crankshaft speed. *Mech. Syst. Signal Process.* **2013**, *38*, 628–633. [[CrossRef](#)]
21. Naik, S. Advanced misfire detection using adaptive signal processing. *Int. J. Adapt. Control Signal Process.* **2004**, *18*, 181–198. [[CrossRef](#)]
22. Zhihua, P.L.Y.J.W.; Xinping, Z.Y.Y. Diagnosing Leakage of Valves in Engines by Analyzing Instantaneous Speed Fluctuations. *J. Wuhan Transp. Univ.* **2000**, *1*, 017.
23. Seuling, S.; Hamedovic, H.; Fischer, W.; Schuerg, F. Model based engine speed evaluation for single-cylinder engine control. *SAE Tech. Pap.* **2012**.
24. Chang, X.H.; Wang, Y.M. Peak-to-peak filtering for networked nonlinear DC motor systems with quantization. *IEEE Trans. Ind. Inform.* **2018**, *14*, 5378–5388. [[CrossRef](#)]
25. Gupta, H.N. *Fundamentals of Internal Combustion Engines*; PHI Learning Pvt. Ltd.: New Delhi, India, 2012.
26. Metghalchi, M.; Keck, J.C. Burning velocities of mixtures of air with methanol, isooctane, and indolene at high pressure and temperature. *Combust. Flame* **1982**, *48*, 191–210. [[CrossRef](#)]
27. Colin, O.; Benkenida, A.; Angelberger, C. 3D modeling of mixing, ignition and combustion phenomena in highly stratified gasoline engines. *Oil Gas Sci. Technol.* **2003**, *58*, 47–62. [[CrossRef](#)]
28. Crowl, D.A.; Louvar, J.F. *Chemical Process Safety: Fundamentals with Applications*; Pearson Education: London, UK, 2001.
29. Zhang, T.; Liao, Y. Attitude measure system based on extended Kalman filter for multi-rotors. *Comput. Electron. Agric.* **2017**, *134*, 19–26. [[CrossRef](#)]
30. Helm, S.; Kozek, M.; Jakubek, S. Combustion torque estimation and misfire detection for calibration of combustion engines by parametric Kalman filtering. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4326–4337. [[CrossRef](#)]
31. Zhao, J.B.; Netto, M.; Mili, L. A robust iterated extended Kalman filter for power system dynamic state estimation. *IEEE Trans. Power Syst.* **2017**, *32*, 3205–3216. [[CrossRef](#)]
32. Al-Durra, A.; Canova, M.; Yurkovich, S. Application of extended Kalman filter to on-line diesel engine cylinder pressure estimation. In Proceedings of the ASME 2009 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, Hollywood, CA, USA, 12–14 October 2009; pp. 541–548.
33. Cordeiro, T.F.K.; Da Costa, J.P.C.; De Sousa Júnior, R.T.; So, H.C.; Borges, G.A. Improved Kalman-based attitude estimation framework for UAVs via an antenna array. *Digit. Signal Process.* **2016**, *59*, 49–65. [[CrossRef](#)]
34. Department of Theoretical Mechanics of Harbin Institute of Technology. *Theoretical Mechanics*, 7th ed.; Higher Education Press: Beijing, China, 2009; pp. 288–322.
35. Yu, Z. *Automobile Theory*; Machinery Industry Press: Beijing, China, 2009.
36. Yang, Y.; Peng, Z.K.; Dong, X.J.; Zhang, W.M.; Meng, G. General parameterized time-frequency transform. *IEEE Trans. Signal Process.* **2014**, *62*, 2751–2764. [[CrossRef](#)]
37. Phillips, C.L.; Parr, J.M.; Riskin, E.A. *Signals, Systems, and Transforms*; Prentice Hall: Upper Saddle River, UK, 2003.
38. Waadeland, C.H. Synthesis of asymmetric movement trajectories in timed rhythmic behavior by means of frequency modulation. *Hum. Mov. Sci.* **2017**, *51*, 112–124. [[CrossRef](#)]

39. Leclere, Q.; Pezerat, C.; Laulagnet, B.; Polac, L. Application of multi-channel spectral analysis to identify the source of a noise amplitude modulation in a diesel engine operating at idle. *Appl. Acoust.* **2005**, *66*, 779–798. [\[CrossRef\]](#)
40. Payri, F.; Olmeda, P.; Guardiola, C.; Martín, J. Adaptive determination of cut-off frequencies for filtering the in-cylinder pressure in diesel engines combustion analysis. *Appl. Therm. Eng.* **2011**, *31*, 2869–2876. [\[CrossRef\]](#)



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

An Adaptive Track Segmentation Algorithm for a Railway Intrusion Detection System

Yang Wang ^{1,2}, Liqiang Zhu ^{1,2,*}, Zujun Yu ^{1,2} and Baoqing Guo ^{1,2}

¹ School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China; 12116331@bjtu.edu.cn (Y.W.); zjyu@bjtu.edu.cn (Z.Y.); bqguo@bjtu.edu.cn (B.G.)

² Key Laboratory of Vehicle Advanced Manufacturing, Measuring and Control Technology (Beijing Jiaotong University), Ministry of Education, Beijing 100044, China

* Correspondence: lqzhu@bjtu.edu.cn; Tel.: +86-10-51684151

Received: 6 May 2019; Accepted: 3 June 2019; Published: 6 June 2019

Abstract: Video surveillance-based intrusion detection has been widely used in modern railway systems. Objects inside the alarm region, or the track area, can be detected by image processing algorithms. With the increasing number of surveillance cameras, manual labeling of alarm regions for each camera has become time-consuming and is sometimes not feasible at all, especially for pan-tilt-zoom (PTZ) cameras which may change their monitoring area at any time. To automatically label the track area for all cameras, video surveillance system requires an accurate track segmentation algorithm with small memory footprint and short inference delay. In this paper, we propose an adaptive segmentation algorithm to delineate the boundary of the track area with very light computation burden. The proposed algorithm includes three steps. Firstly, the image is segmented into fragmented regions. To reduce the redundant calculation in the evaluation of the boundary weight for generating the fragmented regions, an optimal set of Gaussian kernels with adaptive directions for each specific scene is calculated using Hough transformation. Secondly, the fragmented regions are combined into local areas by using a new clustering rule, based on the region's boundary weight and size. Finally, a classification network is used to recognize the track area among all local areas. To achieve a fast and accurate classification, a simplified CNN network is designed by using pre-trained convolution kernels and a loss function that can enhance the diversity of the feature maps. Experimental results show that the proposed method finds an effective balance between the segmentation precision, calculation time, and hardware cost of the system.

Keywords: railway intrusion detection; scene segmentation; scene recognition; adaptive feature extractor; convolutional neural networks

1. Introduction

With a continuous increase in the public's expectation for railway safety, railway intrusion detection systems require more effective technology to detect objects intruding into the track area and to provide real-time alarm information for the command center [1]. Railway intrusion behavior is defined as an object intruding into the track area and endangering the safe operation of trains. Typical intruding objects include rocks falling from a hill beside railway line or a tunnel entrance, pedestrians, vehicles and their cargo staying in the railroad crossing area or falling from the bridge over the railway.

Depending on the detecting principle, railway intrusion detection systems can be divided into two categories: the contact type and the non-contact type. A representative of the contact type is the protective metal net installed along the line to block an object from intruding into the clearance, and the system will send the alarm information when the physical deformation of the net is measured by a dual-power sensor [2] or fiber grating sensor [3,4]. The systems based on the non-contact measurement technology use infrared sensor [5] or laser scanner [6,7] to get the size and location

of the intruding object [8]. Video surveillance is also widely used as another kind of non-contact intrusion detection systems because of the large monitoring area, convenient installation, maintenance, and good observable results [9]. As shown in Figure 1, we established an intrusion detection system for the Shanghai–Hangzhou high-speed railway in China. The system contains data process servers, communication networks, and 1550 cameras, including both of fixed and PTZ cameras.

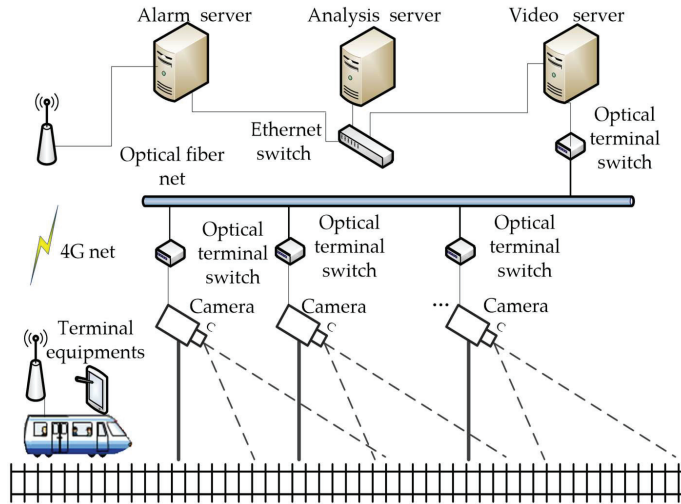


Figure 1. Structure of the railway intrusion detection system.

The threat level of an intrusion behavior will be evaluated by the category, location, and moving trajectory of the object with respect to the track area. The information of the intruding object can be extracted by image processing algorithms, e.g., density-based spatial clustering of applications with noise (DBSCAN) [10], fast background subtraction (FBS) [11], Kalman filtering [12], principal components analysis (PCA) [13]. DBSCAN uses extremum points of scan sequence as core objects of clustering, and the movement and distribution characters are used to judge whether the cluster is a train or other foreground object. FBS projects the scene image into one dimension (x or y dimension) to locate position of the foreground object by the change of the peak value. KF classifies the objects acquired via image background subtraction by support vector machine (SVM), and then using the Kalman-filter tracking algorithm to analyze the behavior and moving trend of the objects. PCA projects the statistic of the scene images and the successive images in a transformation space and calculates the Euclidean distance, which is greater than a threshold, is considered like belonging to motion objects. Most of the above-mentioned algorithms only focus on the foreground object, rather than the track area in the background. Therefore, the position and boundary of track area are still delineated manually in advance, as shown in Figure 2.

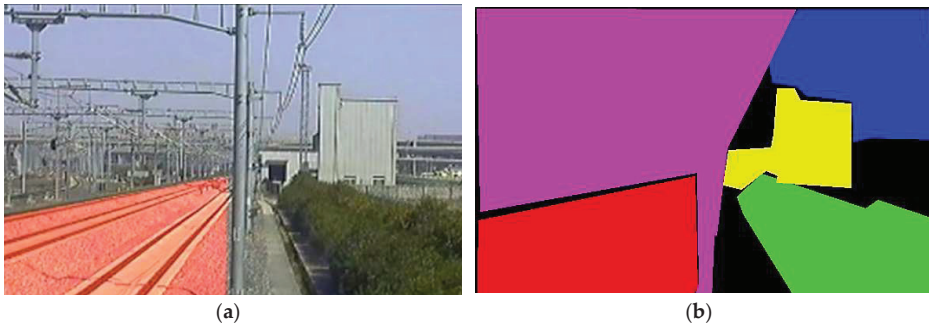


Figure 2. Railway scene and the local areas, labeled manually. The image quality is susceptible to external influences, such as the illumination, weather, and even the dust on the lens. **(a)** The red area is the track area to be surveilled. The track area includes the rails, sleepers, subgrades, or high-speed railway slabs. **(b)** Labeling the different area of the railway scene with different colors by manual, including track area (**red**), sky (**blue**), catenary system (**purple**), green belt (**green**), and ancillary buildings (**yellow**). The precision depends on the patience of the manual operator.

The precision of the track area boundary directly affects the reliability of intrusion detection. With an increasing number of surveillance cameras along the railway line, especially as some PTZ cameras will change their focal lengths and angles temporarily for different applications, manual labeling has become time-consuming and laborious. Thus, for the efficiency of the railway intrusion detection system, a scene segmentation algorithm is needed to recognize the track area and delineate the boundary automatically. The algorithm will be applied to initialize surveillance areas after the installation of all cameras, and to relearn them when the operator adjusts PTZ cameras. Meanwhile, the practical engineering application has many requirements: the relevant image parsing algorithm should not only have good segmentation precision and classification accuracy, but also be able to process temporarily changing scenes quickly. In addition, the algorithm should have small number of parameters and can be easily applied into the data processing servers with different hardware configurations and even into the embedded surveillance equipment in the field.

Currently, there are two ways to parse a scene. The traditional way will segment the scene image into superpixels, ultrametric contour maps (UCM), or other fragmented segment regions [14,15], and then combine them into candidates of objects or local areas based on Markov random fields (MRFs), conditional random fields (CRFs), multiscale combinatorial grouping (MCG), or other rules [16–18]. These traditional methods will generate fragmented regions with precise boundaries and require time-consuming iterative calculations to form a best candidate of an object or a local area. In addition, category information of objects cannot be produced. The second way relies on deep neural networks, e.g., fully convolutional networks (FCN) [19,20], to process the feature extraction, combination, segmentation, and recognition at the same time. A FCN can achieve the segmentation and recognition in a single process. One drawback of FCNs is that the boundary line generated is usually a smooth curve, which will miss the corner of the track area. In addition, FCN has big memory footprint and needs a GPU to accelerate its large amount of computation.

In this paper, we propose an adaptive segmentation algorithm that can take advantage of both methods while avoiding their shortcomings. Like the existing traditional methods, we extract the texture distribution of the image to generate the boundary point with different weight for segmenting the image into small fragmented regions, and then the regions are combined into local areas with precise boundaries; finally, we apply a specially designed convolutional neural network (CNN) for the area's classification without the need of GPU. Our main contributions include:

- To accelerate the generation of small fragmented regions, we propose a method to find the optimal set of Gaussian kernels with adaptive directions for each specific scene. By making full use of the straight-line characters of the railway scene, a smaller number of adaptive directions are calculated according to the maximum points in Hough transformation rather than being chosen from a set of fixed angles in the traditional way. As a result, the calculation time for the boundary extraction and fragmented region generation is cut in half;
- A new clustering rule based on the boundary weight and the size of the region is set up to accelerate the combination of the regions into local areas. The number of regions is reduced in the process of weak boundary point removal by filtration, and the smallest remaining region is combined with its neighbor region, which shares the weakest boundary;
- We propose a specially designed CNN model to achieve the fast classification of local areas without the need of GPU. The local areas are divided into two categories: the track area which is used to judge the intrusion behavior and the rest area which is unrelated to the intrusion. The convolution kernels are pre-trained, and a sparsity penalty term is added into the loss function to enhance the diversity of the convolutional feature maps.

The rest of this paper is organized as follows. In Section 2, we review the related works on image parsing algorithms. Section 3 explains the proposed fast image segmentation process. Section 4 explains the proposed simplified CNN network structure and the optimization process. Section 5 presents the experimental results and discusses them. The last section summarizes our conclusions.

2. Related Work

2.1. Image Parsing by Traditional Methods

To segment an image using the traditional methods, the first step is to calculate the correlation between the adjacent pixels in the scene image, and then segment the image into fragmented regions by a certain convergence criterion [21,22]. The superpixel algorithm, for example, converts the image from the RGB color space to CIE-Lab color space to form a five-dimensional vector (brightness, color A, color B, and position x, position y), and the vector distance between two pixels representing their similarity, is used to generate the small segments patches [14,23]. A spatial pyramid descriptor fuses the gray, colored and edge gradient into one feature vector for the SVM classifier to recognize a traffic sign [24]. The image can also be converted into the YCbCr color space, and the local texture features in different channels are matched with the artificially designed template to locate the position of the traffic sign [25]. Therefore, converting the image from the RGB color space into another feature space can obtain more dimensional information channels: brightness, texture, and other feature maps besides RGB color.

To achieve the final segmentation, the fragmented regions need to be combined. The internal correlations among the adjacent regions are calculated according to different rules, and the regions are combined into local areas according to their correlation values. For example, the K-means clustering rules are used in different practical engineering applications, such as object detection for the synthetic aperture radar (SAR) image and the sea scene [26–28]. The MCG algorithm is another grouping strategy using random forests to combine the multiscale regions into highly accurate object candidates.

MCG can process one image (pixel size 90×150) in 7 s, and the mean Intersection over Union (IU) is about 80% [18,29,30]. The clustering rules influence the combination precision, which is also directly proportional to the calculation time; as a result, MCG is suitable for the initial or post processing of a fixed scene, not for real-time processing of temporarily changing scenes. Therefore, to accelerate the whole scene segmentation process, we choose to improve the traditional methods in both generation and combination of the fragmented regions while maintaining the segmentation precision.

2.2. Image Parsing by Deep Learning Methods

Deep learning methods have also been widely used in image parsing recently, e.g., various convolutional networks, which have better robustness to image translation, rotation, scaling, and distortion. Deep learning methods can be divided into three types: image classification [19], object detection [31], and pixelwise prediction [20]; and the complexity of their network structures increases from image-wise to pixel-wise.

For the pixelwise segmentation of a scene, the convolutional networks can be combined with the superpixels, the random effect model and the texture segmentation to generate the pixelwise labels [32], and also can be used as a classifier to classify the feature maps containing RGB and depth information [33–35]. FCN can even process feature extraction, combination, segmentation, and recognition at the same time, also achieving a pixelwise prediction [20].

Depending on the details of different FCN structures, the mean IU of FCN is about 80%, the accuracy is about 90%, and the quantity of the parameter is about 57 M to 134 M. The massive number of parameters and computation need a GPU with big memory to handle the operation, leading to a high cost for practical applications. Therefore, we choose to use the traditional methods to get the precise boundary of the local area first, and then use a simplified CNN only to classify the local areas without the need of GPU. However, the reduction of the network size causes low accuracy in the classification, so extra care has to be taken in optimizing the network structure and the training process.

3. Railway Scene Segmentation

As shown in Figure 2b, typical railway scene consists of different areas, including track area, sky, catenary system, green belt, and ancillary buildings. The precision of the track area boundary directly affects the reliability of the judgement about whether the intrusion occurs or not. The track area is defined as the clearance area including rails, sleepers, subgrades or high-speed railway slabs, as shown in Figure 2a. To avoid manual labeling, a fast and precise railway scene segmentation algorithm is proposed.

Figure 3 illustrates the outline of the proposed algorithm. We first calculate the feature distribution in a small image patch (pixel size 15×15) representing the central pixel of the patch, then evaluate the central pixel's probability of being a boundary point, and finally use the boundary weights to segment the image according to a fast combination rule. Unlike the traditional method, we use a smaller set of adaptive Gaussian kernels to extract the pixel color (*PC*) distribution and pixel similarity (*PS*) distribution of the image in different channels *C* and by different scales *S*. The Gaussian kernels are rotated by a set of adaptive θ s, calculated from Hough transformation. The detailed procedure of boundary weight generation is described in the remainder of this section.

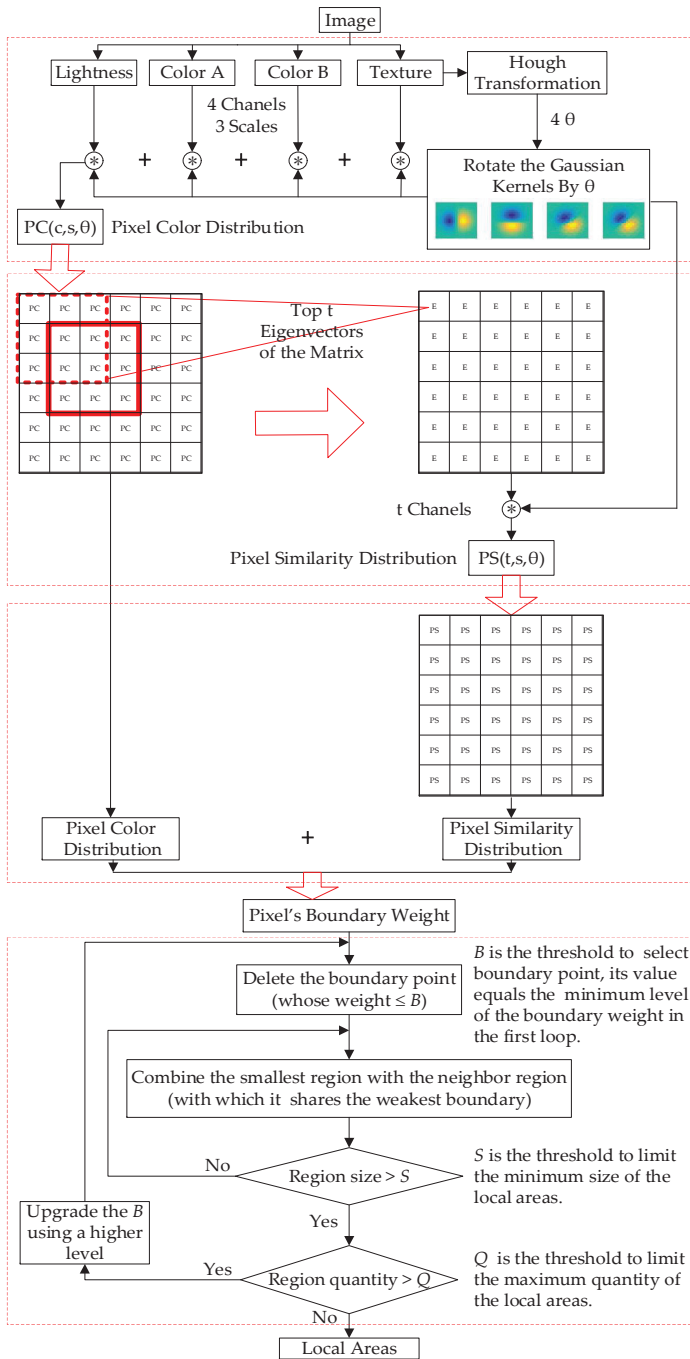


Figure 3. The procedure to segment an image into fragmented regions and combine them into local areas.

3.1. Generation of Fragmented Regions

Firstly, we convert the image into the CIE lab color space, getting 3 channels: brightness, color A, and color B. Images in different channels will be scaled by $s = (0.5, 1, 2)$. In each channel, the image is convoluted with Gaussian kernels to get the color value distribution; each kernel has a special orientation angle θ . Define $G(x, y, \theta, c, s)$ as the convolution result at pixel $P(x, y)$, with angle θ , in channel c , by scale s . Then PC , the pixel's color distribution, can be obtained by

$$PC(x, y, \theta) = \sum_s \sum_c \alpha_{c,s} G(x, y, \theta, c, s) \quad (1)$$

where $\alpha_{c,s}$ is a weighting coefficient.

Secondly, define $S_{similarity}(i, j)$ as the maximum PC value of all pixels on the line $l_{i,j}$ connecting two pixels i and j in a small image patch by Equation (2), representing the similarity between pixel i and j .

$$S_{similarity}(i, j) = \exp(-\text{Max}\{PC(x, y) | (x, y) \in l_{i,j}\}) \quad (2)$$

Calculate the similarity of each pixel $i_{x,y}$ in the patch and the central pixel j_{center} , assign $S_{similarity}(i_{x,y}, j_{center})$ to each element $MS(x, y)$ of the Matrix of Similarity MS , and assemble MS representing the similarity matrix between each pixel in the image patch and the central pixel.

Calculate the top t eigenvalues and eigenvectors of MS . Assign the eigenvector to the central pixel $P(x, y)$ marked as $e(x, y, t)$, forming a feature map E of the image, representing the similarity of the adjacent points. Again, in each dimension of the feature map, convolute $E(t)$ with Gaussian kernel of orientation θ to get the similarity distribution. Define $g(x, y, \theta, t, S)$ as the convolutional result at location $E(x, y)$, with angle θ , in dimension t , by scale s . Then, the pixel's similarity value distribution can be obtained as

$$PS(x, y, \theta) = \sum_s \sum_t \beta_{c,s} g(x, y, \theta, t, s) \quad (3)$$

where $\beta_{c,s}$ is a weighting coefficient.

Finally, $B(x, y)$, the possibility of the pixel $P(x, y)$ being a boundary point, can be estimated by

$$B(x, y) = \sum_{\theta} PC(x, y, \theta) + \sum_{\theta} PS(x, y, \theta) \quad (4)$$

3.2. Finding the Optimal Set of Gaussian Kernels

It can be found that, in the process of estimating $B(x, y)$, convolution operations (Equations (1) and (3)) using Gaussian kernels with different orientation angles θ cost most of the computation, which can be reduced if a smaller set of Gaussian kernels are used. The traditional UCM algorithms choose fixed size of $\theta = (\theta_1, \theta_2, \theta_3, \dots)$ with 8 or 16 values uniformly distributed from 0 to π . Here we propose to utilize the characteristics of the railway scene to find a much smaller set of useful orientation angles and thus a smaller set of Gaussian kernels. Usually, in railway scene, there is a clear vanishing point (VP), and the boundaries of many local areas are lines passing through the VP. Therefore, if we can automatically adjust the candidate θ for each specific scene to enhance the weights of the line boundary points of the relevant areas, then we will be able to use a smaller set of θ to accelerate the process.

We propose to find the candidate θ by filtering the original image with a Canny kernel [36], and then convert the obtained texture feature into the Hough coordinate system using

$$\rho = x \cos \theta' + y \sin \theta', -\frac{\pi}{2} < \theta' < \frac{\pi}{2} \quad (5)$$

As shown in Figure 4a, each curve in the Hough coordinate system stands for one point in the Cartesian coordinate system. If the curves (colorful curve lines in Figure 4a) have one intersection

point in the Hough coordinate system, then the corresponding points (blue point in Figure 4a) in the Cartesian coordinate system are collinear.

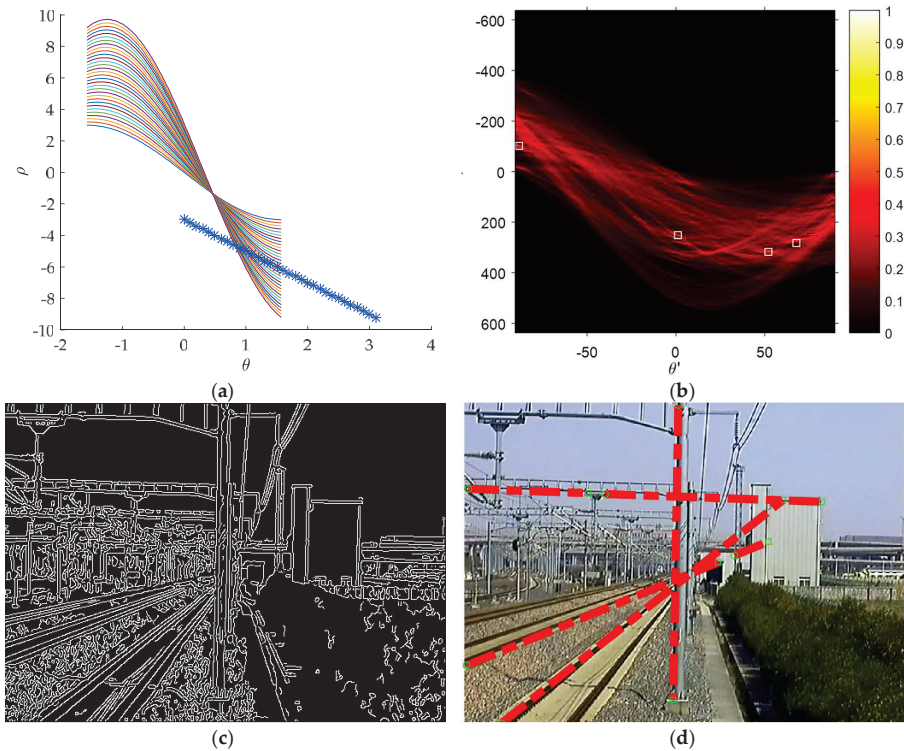


Figure 4. Using Hough transformation to detect the most significant lines in the Hough coordinate system. (a) The intersection point of a group of curves in the Hough coordinate system means there are a group of collinear points in the Cartesian coordinate system. (b) The more the curves intersect in the Hough coordinate system, the lighter the intersection point is, meaning that there are more collinear points along this line in the Cartesian coordinate system. (c) The texture feature maps filtered by canny filter. (d) The top four significant lines.

Let $H(\theta', \rho)$ be the number of curves intersecting at point (θ', ρ) and find the point with maximum $H(\theta', \rho)$, where there are the largest number of points which are collinear on the corresponding line in the Cartesian coordinate system. The line can be expressed as

$$y = -\frac{1}{\tan \theta'}x + \frac{\rho}{\sin \theta'} = kx + b \tag{6}$$

To find a small set of four orientation angles, one can take the top four maximum θ in $H(\theta', \rho)$, e.g., the points with highest ‘lightness’ in Figure 4b: $\theta' = 68^\circ, 52^\circ, 0^\circ$ and -88° . Here we change the $\theta = 90^\circ - \theta' = 22^\circ, 38^\circ, 90^\circ$, and 178° in order to obtain a range of values from 0° to 180° ($0-\pi$). Based on the selected set of orientation angles, the Gaussian kernels can be constructed correspondingly by

rotating the Gaussian. As shown in Figure 5, in the Cartesian coordinate system X-O-Y, point $P(x, y)$ rotates around the point $o(\frac{W}{2}, \frac{W}{2})$ an angle θ to $P'(x', y')$, which can be formulated as

$$\begin{aligned}
 \begin{bmatrix} x & y & 1 \end{bmatrix} &= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5W & 0.5W & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W & 0.5W & 1 \end{bmatrix} \\
 &= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ -0.5W(\cos \theta - \sin \theta - 1) & -0.5W(\sin \theta + \cos \theta - 1) & 1 \end{bmatrix} \quad (7)
 \end{aligned}$$

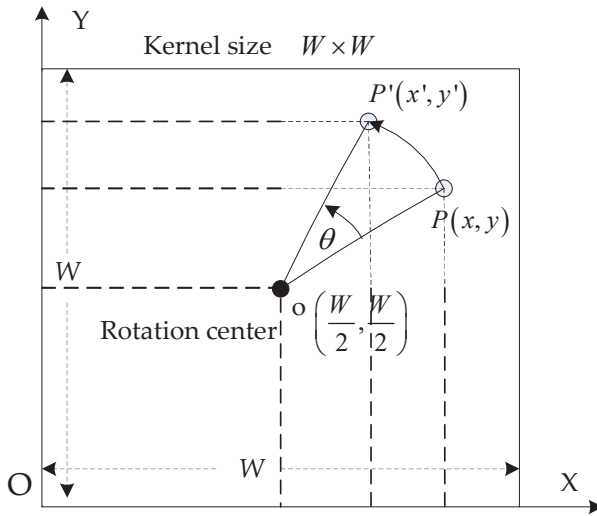


Figure 5. Calculating the rotation matrix of the Gaussian kernel. The rotation center is on the kernel center.

Figure 6 shows several Gaussian kernels rotated by the optimal set of $\theta = 22^\circ, 38^\circ, 90^\circ,$ and 178° obtained above and $\theta = 112.5^\circ$, one of the eight uniformly-distributed values commonly used in traditional UCM algorithms, respectively. The results show that the features of the horizontal catenary bracket, the vertical catenary column and the declining track are strengthened obviously in the first four filters, contrasting with the feature extraction equality in the fifth filter. The universality of using 8 or 16 uniform values in different angle θ causes a redundant calculation when applied to the railway scene. Therefore, adjusting a smaller number of θ adaptively to filter the feature map can accelerate the boundary weighting to generate the fragmented regions.

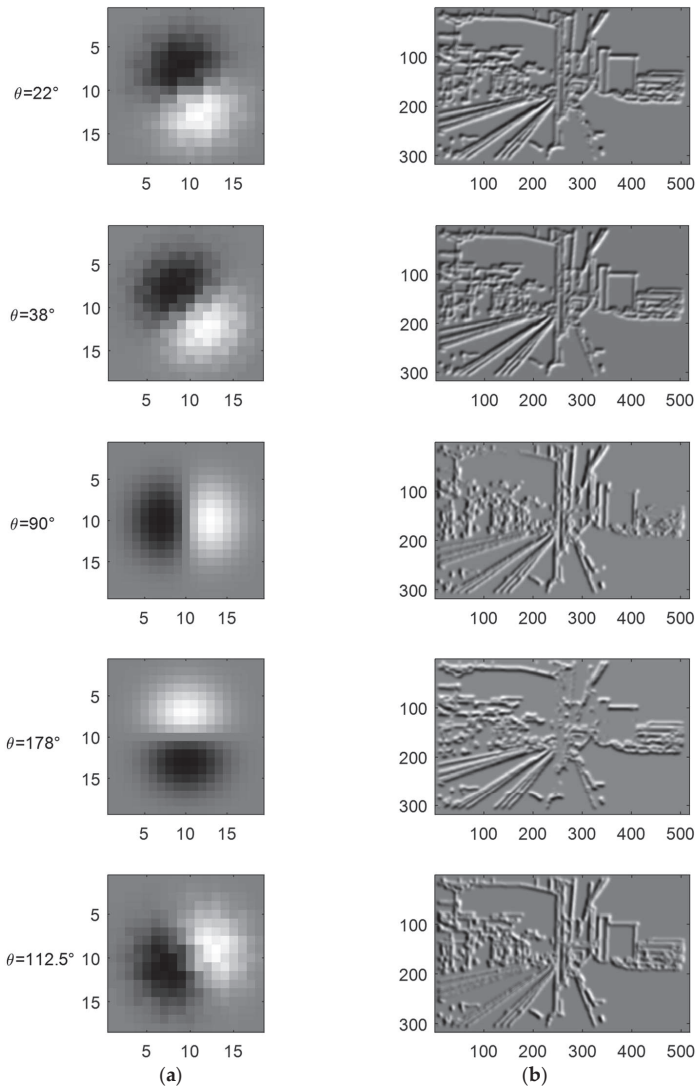


Figure 6. Different kernels and the convolution results of CIE-lab color L channel. (a) First order derivative Gaussian kernels rotated by five angles. (b) Results of the Gaussian convolution.

3.3. Combination Rule

The fragmented regions generated by the adaptive boundary detection are shown in Figure 7a. The higher the boundary weight is, the brighter the point is shown in the gray feature map, indicating that the point is more likely to become a boundary point.

A clustering rule based on both of the boundary weight and the region size is proposed to combine the fragmented regions into local areas. The number of the regions will be reduced in the process of weak boundary point removal by filtration. The smallest remaining region will be combined with its neighbor region, with which it shares the weakest boundary. Repeat this iteration until the statistical parameters meet the requirements. The process is as follows:

1. Let $B(m)$ be the normalized value of the boundary point's weight $B(x_m, y_m)$, where $m = 1, 2, 3 \dots M$, and M is the total number of boundary points:

$$B(m) = \text{sigmoid}(B(x_m, y_m)) = \frac{1}{1 + e^{-B(x_m, y_m)}} \quad (8)$$

2. The statistical distribution of the boundary point weight $B(i)$ is shown in Figure 7b. There are many levels of boundary point weights. Choose the minimum level B as the threshold to delete the weak boundary points $B(m) \leq B$;
3. The fragmented regions will be reduced by reconnecting the breakpoints of the boundary line using expansion and corrosion operations, as shown in Figure 7c. The new regions are shown in Figure 7d;
4. The statistical distribution of region size $f(n)$ is shown in Figure 7e, where $n = 1, 2, 3 \dots N$, and N is the serial number of the regions. Choose the smallest region along its boundary line and find the neighbor region which shares the weakest boundary with it. Then combine them into a new region. As shown in Figure 7d, regions in number 1, 2, 3, and 4 are combined as one new region in Figure 7f;
5. Repeat Step 4 to reduce N until the area of the smallest region is larger than a threshold S , which is used to limit the minimum area of the remained regions;
6. Compare the final N with another threshold Q to limit the minimum quantity of the remained regions. If $N > Q$, select the second minimum level weight B and go back to Step 2;

Figure 7g is the original railway scene image, and the Figure 7h is the result of our segmentation algorithm. The railway scene only contains five categories of areas, and the shape of the area is usually in a large and radial pattern. Therefore, we set the minimum area threshold S to 10% of the whole image and the maximum quantity threshold Q to 10, which will prevent the remained regions from being too fragmented. The remained regions will be adjusted into a standard size of 64×64 and RGB 3 channels, after being classified by the CNN in Section 4, the remaining regions with the same labels will be combined as one local area.

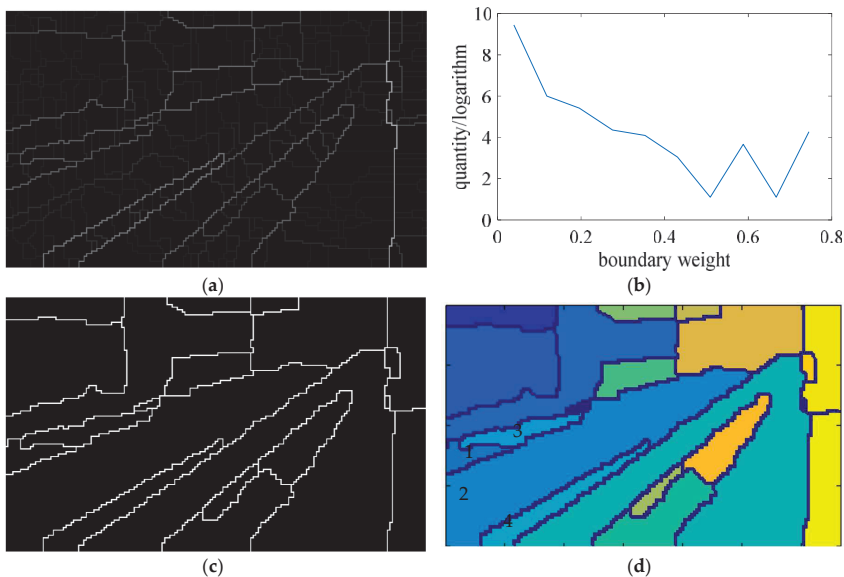


Figure 7. Cont.

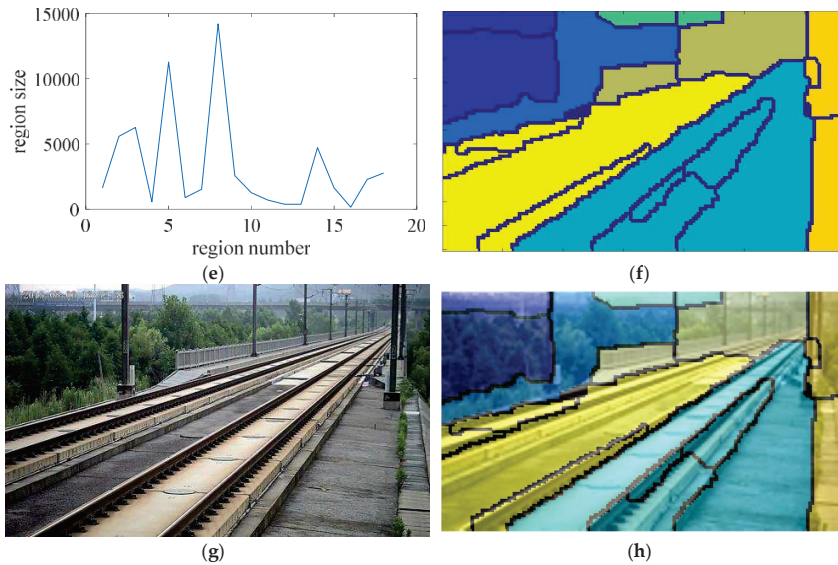


Figure 7. The procedures of combining the fragmented regions into local areas. According to the adjustment and experiments, for the railway scene, the scene image is set to a pixel size of 90×150 , the number of adaptive θ is reduced to 4, the number of reserved areas Q is set to 10, and the smallest fragmented area S is set to 10% of the total size of the image. (a) Boundary with weight. (b) Distribution of boundary weight and quantity. (c) Delete the weak boundary. (d) Fragmented regions. (e) Distribution of the region size and serial number. (f) Local areas after the fragmented regions are combined. (g) The original railway scene image, (h) is the result.

4. Local Area Recognition in Railway Scene

To automatically label the local areas in real time without the help a GPU, we design a simplified CNN with less layers and kernels. To compensate the reduced accuracy, the convolution kernels are pre-trained, and a sparsity penalty term is added into the loss function to enhance the diversity of the feature maps.

4.1. Structure of Simplified CNN

Before designing and applying a simplified CNN, we first construct a dataset of local area images for training it. As shown in Figure 8, there are mainly five basic categories of elements in a typical railway scene, including track area, sky, catenary system, green belt, and ancillary buildings. To sample the dataset, five solid line rectangles are manually defined to cover the five different areas. We program a simple extraction code to take the image patches using the dotted-line box as samples with the same category of the outer rectangle. We set up a group of constraint parameters to control the dotted box to extract the patches at a random position, by a random scale, maintaining inside of each rectangle. The image patches are adjusted into a pixel size of 64×64 and RGB 3 channels to assemble our five-category datasets of railway local area. However, for the specific application of this paper, our target is focused on the track area for judging intrusion behavior, so besides the 'track' label, we merge the other four elements into one category labeled as 'others'. There are 9000 image patches in total, in which 5000 images are used for training our net, 2000 images are used for cross-validation, and 2000 images are used for testing.

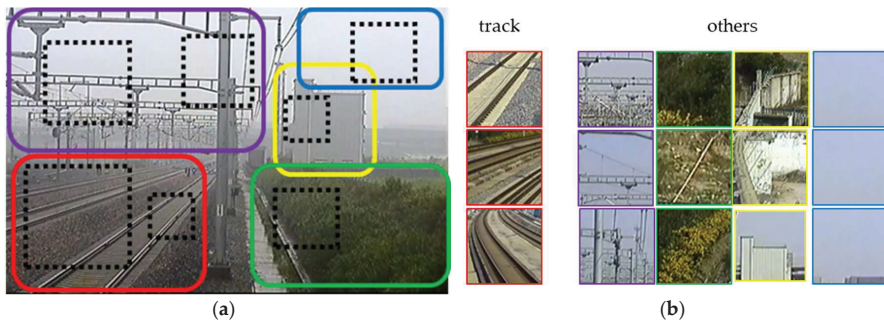


Figure 8. Collecting samples of local areas for CNN training. (a) Solid-line rectangles are delineated by manual with labels, including the track area (red), sky (blue), catenary system (purple), green belt (green), ancillary buildings (yellow). The dotted-line boxes are extractor windows. (b) The dataset containing two categories for training the CNN.

A simplified CNN structure is designed for fast recognition, which consists of an input layer, two convolution layers $C1$ and $C2$, two mean pooling layers $S1$ and $S2$, and a logistic classification layer, as shown in Figure 9.

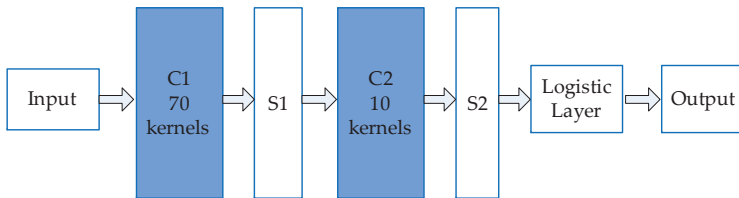


Figure 9. Structure of the simplified CNN. The size of the input image is a pixel size of 64×64 with RGB 3 channels. The output is one of the two category labels.

As shown in Table 1, we conducted five experiments with different kernel quantities and sizes. It can be seen that increasing the kernel size and quantity may increase the accuracy, but the accuracy is still less than 80%. Although the railway scene is very simple, only containing several typical area categories, the shapes, color, and texture features of the area belonging to the same category are still very complex and different. Therefore, the training process must be optimized to increase the accuracy.

Table 1. Experimental results of different CNN network structures.

| Kernel Size | Kernel Quantity | | Calculation Time (s) | Accuracy |
|--------------|-----------------|----|----------------------|----------|
| | C1 | C2 | | |
| 3×3 | 50 | 10 | 0.00372 | 72.25% |
| | 70 | 10 | 0.00495 | 73% |
| | 100 | 10 | 0.00689 | 75% |
| 5×5 | 100 | 10 | 0.0125 | 76% |
| 7×7 | 100 | 10 | 0.0217 | 76.5% |

4.2. Optimization of the Simplified CNN

To increase the accuracy, kernels are pre-trained to extract better low-level features. The pre-training strategy is based on autoencoder-decoder network; and the $W_{i,3 \times 3}^1$ after training in first layer is applied as the convolution kernel in the first convolution layer $C1$, as shown in Figure 10 for the case

with kernel size of 3×3 and in RGB 3 channels. During the training, 3×3 patches in RGB 3 channels are randomly selected from random railway scene images, as shown in Figure 11a. The result of the pre-trained kernels is shown in Figure 11b, where the patches and the kernels are all in RGB 3 channels.

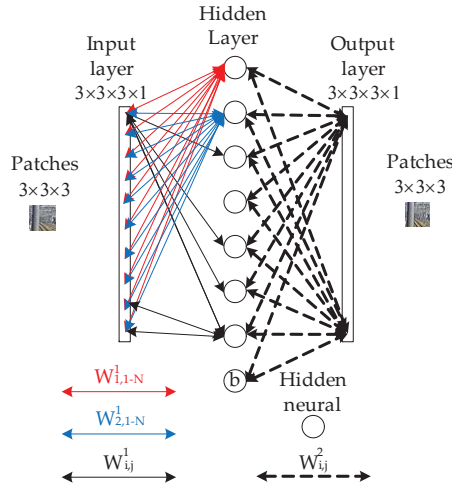


Figure 10. Structure of the autoencoder-decoder network. The hidden layer contains 70 hidden neurons; W denotes the weight associated with the connection between neurons; and the network is trained to produce output the same as its input.

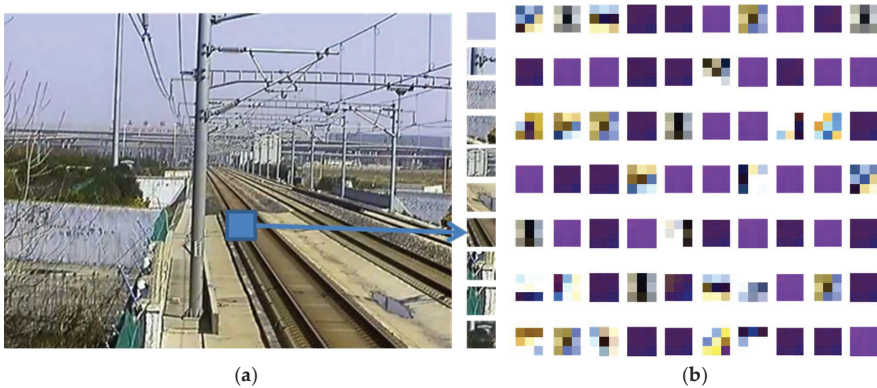


Figure 11. Pre-trained convolution kernels using the autoencoder-decoder algorithm. (a) The image patches are extracted from the left railway scene image for the kernel training. (b) The pre-trained kernels used in convolution layer $C1$.

After pre-training, the input weights of each neuron in the hidden layer are used as the initial weights of kernels in the first convolution layer $C1$ in Figure 9. The rest of CNN in Figure 9 are randomly initialized and then trained by using a backpropagation algorithm (stochastic gradient descent, SGD). To enhance the diversity of the feature maps, a sparsity penalty term is added into the loss function J as

$$J = \left\{ \frac{1}{P} \sum_{p=1}^P \frac{1}{2} [h(e_p) - l_p]^2 \right\} + \tau \sum_{f=1}^{10} \left[\chi \lg \frac{\chi}{\eta_f} + (1 - \chi) \lg \frac{1 - \chi}{1 - \eta_f} \right] \tag{9}$$

where

$$\eta_f = \frac{1}{P} \sum_{p=1}^P \sum_{u=1}^{29} \sum_{v=1}^{29} O_{f, e_p}^{(2)}(u, v) \quad (10)$$

e_p is the p -th input image, l_p is the ground truth label, there are totally P images in the dataset, $h(e_p)$ is the output label, τ is the weight of the sparsity penalty term, χ is the sparsity parameter (a smaller value close to 0, e.g., 0.05), η_f is the average output of the f -th feature map in convolution layer C2 (averaged over the training dataset), and $O_{f, e_p}^{(2)}(u, v)$ is the value at position (u, v) in the f -th feature map of the input e_p in the second convolutional layer C2, the size of the feature map is 29×29 pixels.

In the process of backpropagation, the sparsity penalty item will suppress the average output of all feature maps in the second convolutional layer C2, but enforce the output of one feature map at the same time, so as to enhance the diversity of the feature maps and improve the accuracy. The learning rate is set to 0.1, and the decay of the learning rate is 0.001 after each iteration, the final value of J should be less than 0.05.

4.3. Performance of the Simplified CNN

As shown in Table 2, the accuracies of the simplified CNNs with different structures are all increased by using the proposed optimization method, compared with the results of traditional training method shown in Table 1, e.g., the simplified CNN with 70 kernels (3×3 , 3 channels) in C1 and 10 kernels (3×3 , 70 channels) in C2 is used for the proposed segmentation algorithm. The quantity of the network parameters is only 0.02912M. After the railway scene is segmented and classified, the regions with track labels can be combined together as the final track areas.

Table 2. Experiment results of different CNN network structures after the optimization.

| Kernel Size | Kernel Quantity | | Accuracy |
|--------------|-----------------|----|----------|
| | C1 | C2 | |
| 3×3 | 50 | 10 | 98% |
| | 70 | 10 | 98.5 |
| | 100 | 10 | 98.5% |
| 5×5 | 100 | 10 | 98.75% |
| 7×7 | 100 | 10 | 99.25% |

5. Experiments and Results

5.1. Railway Scene Dataset

We collect images from 16 PTZ cameras at straight lines, curves and bridges in the high-speed railway from Shanghai to Hangzhou, China. For each camera, images are collected from 10 different shooting angles, lenses, and under different illumination conditions from 8:00 a.m. to 5:00 p.m. Examples are shown in Figure 12a. There are totally 1760 scene images in the dataset, in which 1000 images are used in the training dataset, 400 images are used in the cross-validation dataset and 360 images are used in the test dataset. These datasets are used to generate the datasets for our simplified CNN (Section 4.1) and the dataset for training the FCN for the comparison experiments.



Figure 12. Samples in railway dataset. (a) Images from PTZ cameras under different conditions. (b) Ground truth of the track area.

5.2. Modification of the Workflow for the Case of Small Track Portion

For cameras on line sections, track area only takes up a small portion of the scene image, while for the ones at tunnel entrances and bridges over railway line, track area usually takes up most of the scene. As shown in Figure 12b, the red track area takes about 25–70% of the whole scene image for different cameras. That means, the complete-processing workflow (Sections 3 and 4) would waste a lot of time calculating the boundaries between the ‘others’ areas (Figure 13b) rather than focusing on the potential track area as the red dotted line rectangle shown in Figure 13a. In order to find the potential track area and reduce the segmentation calculation furthermore, we design a partial-scanning workflow to locate the potential position of the track area before the segmentation and classification by scanning over the railway scene roughly using the proposed CNN. As shown in Figure 13c, we firstly divide the railway scene image into 6×10 cells (yellow cell); each cell and its peripheral zone (red dotted line rectangle) are resized to 64×64 pixels, define their classified labels as the representation of its central cell (red area in Figure 13c); the proposed CNN is used to classify these cells and the output labels are used to identify the potential track area roughly as the red area shown in Figure 13d; A minimum enclosing dotted line rectangle is used to adjust the potential track area into a regular shape as shown in Figure 13d.

The strategy of the partial-scanning workflow reduces the segmentation area, but spends extra scanning time. Thus, the overall processing time depends on the proportion of the track area to the railway scene, as shown in Table 3, the numbers on the left are the scene images in Figure 12, from the left to the right. If the track area takes over more than 88.1% of the railway scene, the performance of the partial-scanning workflow would be worse than the complete-processing workflow. These two workflows can be chosen for different cameras: for those with short focus lens and focus on the near scene full of track area, the complete-processing workflow should be used; for the ones with long focus lens and track area only take a small part of the scene, the partial-scanning workflow should be used.

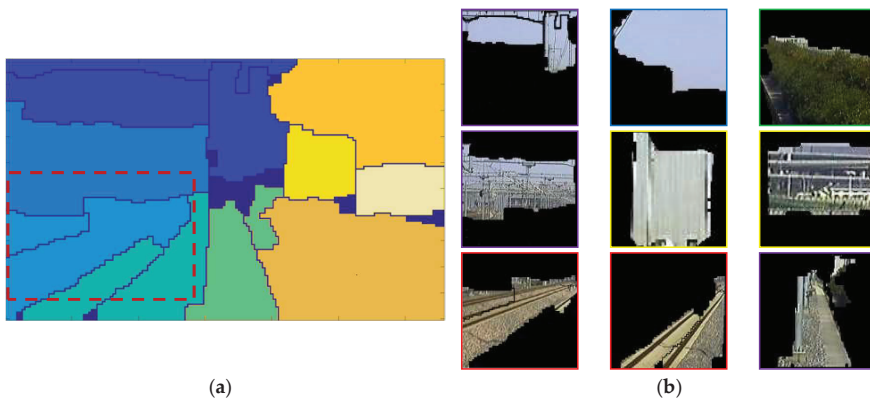


Figure 13. Cont.



Figure 13. Rough scanning over the scene to find the potential track area. (a) Segmentation result of the whole railway scene images. (b) Different local areas. (c) Scanning the railway scene image roughly using proposed CNN. (d) Area in red dotted rectangle is the potential track area, which will reduce segmentation calculation by three-quarters.

Table 3. Calculation time of the comparison experiments with different workflow to segment the railway scene.

| | Partial-Scanning Workflow | | | Complete-Processing Workflow | |
|---|---------------------------|--------------------------|--|------------------------------|----------|
| | Scan Time (s) | Proportion of Track Area | Segmentation and Classification Time (s) | Total (s) | Time (s) |
| 1 | 0.297 | 41.7% | 1.042 | 1.339 | 2.5 |
| 2 | 0.297 | 25% | 0.625 | 0.922 | 2.5 |
| 3 | 0.297 | 75% | 1.875 | 2.172 | 2.5 |
| 4 | 0.297 | 40% | 1 | 1.927 | 2.5 |
| 5 | 0.297 | 30% | 0.75 | 1.047 | 2.5 |

5.3. Metrics

To evaluate the segmentation performance, three criteria are used. The first one is the intersection over union (IU) generally defined as Equation (11), where L represents for the ground truth, R represents the segmentation result; the second one is the pixel accuracy (PA) defined in Equation (12) to evaluate the portion of the area which need to be surveilled are segmented; and the extra pixel (EP) as Equation (13) is used to evaluate the portion of segmented areas which do not need to be surveilled. PA would influence the missing part of the track area which would cause a missing alarm, and the EP would influence the extra part of the track area which would cause a false alarm.

$$IU = \frac{L \cap R}{L \cup R} \quad (11)$$

$$PA = \frac{L \cap R}{L} \quad (12)$$

$$EP = \frac{R - L \cap R}{L} \quad (13)$$

5.4. Performance of the Proposed Segmentation Algorithm

The proposed algorithm is compared with MCG and FCN using images from railway dataset and some examples are shown in Figure 14. In the experiment, the computation platform is equipped with an Intel i5-6500 CPU, 8 GB DDR3 memory, without GPU and MATLAB 2012, and images in the dataset are resized to 90×150 . The MCG method is the pre-trained demo from [17]. The FCN network uses

a standard VGG16 structure trained by VOC2012 dataset for the feature extracting, and upsampled the outputs of the third, fourth, and seventh convolution layers.

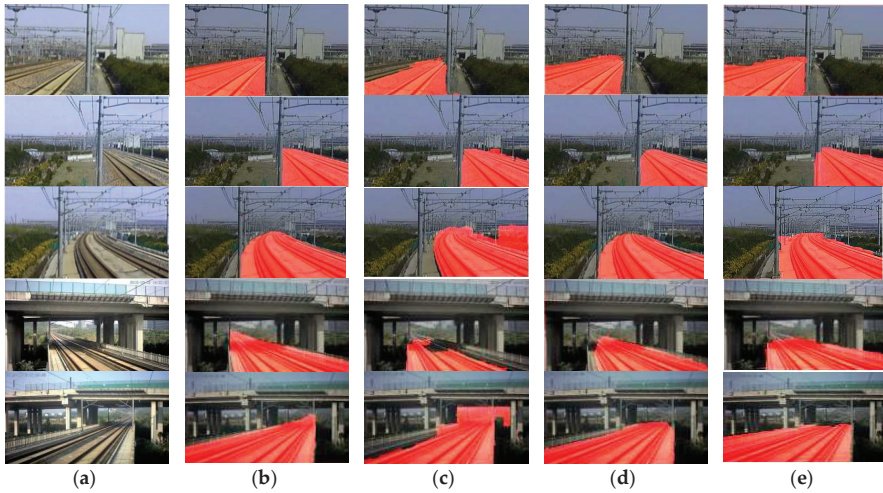


Figure 14. Using different algorithm to detect the track area. (a) The original railway scenes. (b) Ground truth of track areas. (c) Results of the MCG algorithm. (d) Results of the FCN algorithm. (e) Results of our algorithm.

The missing part and the extra part of the segmented track area are shown in Figure 15. For the MCG algorithm, it used the CRFs to combine the fragmented regions into one unified area based on the texture which caused the missing part (as shown in Figure 15e) because of the difference texture between the nearby track and the distant track. The performances of the FCN algorithms were improved slightly from their original results in [19] because of the monotonous railway scene and the small amount of categories; but not too significantly because the shape and color textures of the scene images sampled with different illuminations, weather, and in different seasons were still complex. As shown in Figure 15f,i, the smooth boundary line of the FCN algorithm was not suitable for our railway scene parsing because of the concave and convex shapes at the straight and sharp edge of the region, especially near the area with an acute angle and straight line. Concave and convex shapes caused both a missing part and an extra part of the track area when compared with the ground truth, which would release both the missing alarms and false alarms. For the engineering application, our system would rather release a false alarm than miss a true alarm.

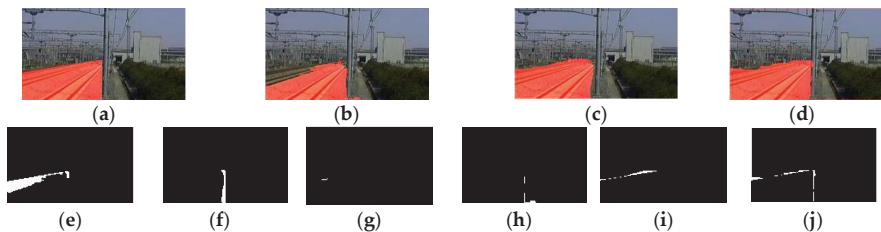


Figure 15. Missing and extra areas of different methods comparing with the ground truth. (a) Manual label of track areas. (b) Results of the MCG. (c) Results of the FCN. (d) Results of our method. (e) Missing part of MCG. (f) Missing part of FCN. (g) Missing part of our method. (h) Extra part of MCG. (i) Extra part of FCN. (j) Extra part of our method.

The performances of the three algorithms are shown in Table 4. It can be found that the proposed algorithm with four optimal Gaussian kernels achieves the highest score in PA, which means that the greatest portion of the surveillance area is found out and thus is preferred for applications.

Table 4. Experimental results of different algorithms.

| Algorithm | | Mean IU | Mean PA | Mean EP | Time (s) |
|---------------|--------------------------------|---------|---------|---------|----------|
| MCG | | 72.05% | 79.94% | 10.63% | 7 |
| FCN | | 89.83% | 91.26% | 16.20% | 41 |
| Our Algorithm | Four optimal Gaussian kernels | 81.94% | 95.90% | 18.17% | 0.9–2.8 |
| | Eight regular Gaussian kernels | 85.23% | 93.85% | 17.56% | 1.1–4.4 |

6. Conclusions

The proposed algorithm uses an adaptive feature distribution extractor for railway track segmentation by making full use of the strong linear characteristics of railway scenes and the typical categories of the local areas. A good balance between segmentation precision, recognition accuracy, calculation time, and complexity of manual operation can be achieved. By using the proposed algorithm, the railway intrusion detection system can automatically and accurately delimit the boundaries of a surveillance scene in real time and greatly improve the efficiency of the system operation. Considering the fact that, in China, there are over 29,000 km of high-speed railways and the average density of cameras on high-speed railway lines is about 2.92 cameras/km, the proposed algorithm is of great significance to improve the efficiency.

The proposed algorithm can be applied into the surveillance system of public places such as airport aprons, highway pavement, and squares. These places share some common characteristics: simple structure full of straight lines—such as airplane runways and different functional areas, vehicles and different lanes, pedestrians and sidewalk lines. Before applying this method, however, the training dataset of the simplified CNN has to include new categories in such scenes, then the proposed algorithm can segment the scene and label each local area.

Author Contributions: Conceptualization, Y.W., L.Z., and Z.Y.; Investigation, Y.W. and B.G.; Methodology, Y.W. and L.Z.; Project administration, Z.Y.; Software, Y.W.; Validation, Y.W.; Writing—original draft, Y.W.; Writing—review & editing, Y.W. and L.Z.

Funding: This research was funded by National Key Research and Development Program of China (2016YFB1200401).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Y.; Yu, Z.; Zhu, L.; Guo, B. Fast feature extraction algorithm for high-speed railway clearance intruding objects based on CNN. *J. Sci. Instrum.* **2017**, *38*, 1267–1275.
2. Hou, G.; Yu, Z. Research on flexible protection technology of high speed railways. *J. Railw. Stand. Des.* **2006**, *11*, 16–18.
3. Cao, D.; Fang, H.; Wang, F.; Zhu, H.; Sun, M. A fiber bragg-grating-based miniature sensor for the fast detection of soil moisture profiles in highway slopes and subgrades. *Sensors* **2018**, *18*, 4431. [[CrossRef](#)] [[PubMed](#)]
4. Zhang, Y. Application study of fiber bragg-grating technology in disaster prevention of high-speed railway. *J. Railw. Signal. Commun.* **2009**, *45*, 48–50.
5. Oh, S.; Kim, G.; Lee, H. A monitoring system with ubiquitous sensors for passenger safety in railway platform. In Proceedings of the 7th International Conference on Power Electronics, Daegu, Korea, 22–26 October 2007; pp. 289–294.

6. Wang, Y.; Shi, H.; Zhu, L.; Guo, B. Research of surveillance system for intruding the existing railway lines clearance during Beijing-Shanghai high speed railway construction. In Proceedings of the 3rd International Symposium on Test Automation and Instrumentation, Xiamen, China, 22–25 May 2010; pp. 218–223.
7. Luy, M.; Cam, E.; Ulamis, F.; Uzun, I.; Akin, S.I. Initial results of testing a multilayer laser scanner in a collision avoidance system for light rail vehicles. *Appl. Sci.* **2018**, *8*, 475. [[CrossRef](#)]
8. Guo, B.; Yu, Z.; Zhang, N.; Zhu, L.; Gao, C. 3D point cloud segmentation, classification and recognition algorithm of railway scene. *Chin. J. Sci. Instrum.* **2017**, *38*, 2103–2111.
9. Zhan, D.; Jing, D.; Wu, M.; Zhang, D.; Yu, L.; Chen, T. An accurate and efficient vision measurement approach for railway catenary geometry parameters. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2841–2853. [[CrossRef](#)]
10. Guo, B.; Zhu, L.; Shi, H. Intrusion detection algorithm for railway clearance with rapid DBSCAN clustering. *J. Sci. Instrum.* **2012**, *33*, 241–247.
11. Guo, B.; Yang, L.; Shi, H.; Wang, Y.; Xu, X. High-speed railway clearance intrusion detection algorithm with fast background subtraction. *J. Sci. Instrum.* **2016**, *37*, 1371–1378.
12. Shi, H.; Chai, H.; Wang, Y. Study on railway embedded detection algorithm for railway intrusion based on object recognition and tracking. *J. China Railw. Soc.* **2015**, *37*, 58–65.
13. Vazquez, J.; Mazo, M.; Lazaro, J.L.; Luna, C.A.; Urena, J.; Garcia, J.J.; Hierrezuelo, L. Detection of moving objects in railway using vision. In Proceedings of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 14–17 June 2004; pp. 872–875.
14. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunck, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [[CrossRef](#)] [[PubMed](#)]
15. Arbeláez, P. Boundary extraction in natural images using ultrametric contour maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, New York, NY, USA, 17–22 June 2006; p. 182.
16. Verbeek, J.; Triggs, B. Region classification with Markov field aspect models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
17. Ladický, L.; Russell, C.; Kohli, P.; Torr, P. Associative hierarchical CRFs for object class image segmentation. In Proceedings of the 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 739–746.
18. Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; Malik, J. Multiscale combinatorial grouping. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 328–335.
19. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. In *Neural Information Processing Systems*; Morgan-Kaufmann: San Francisco, CA, USA, 1990; pp. 396–404.
20. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *39*, 640–651. [[CrossRef](#)] [[PubMed](#)]
21. Ren, X.; Malik, J. Learning a classification model for segmentation. In Proceedings of the 9th International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 1, pp. 10–17.
22. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
23. Zhu, Y.; Luo, K.; Ma, C.; Liu, Q.; Jin, B. Superpixel segmentation based synthetic classifications with clear boundary information for a legged robot. *Sensors* **2018**, *18*, 2808. [[CrossRef](#)] [[PubMed](#)]
24. Liu, Y.; Chen, Y.; Zhang, S. Traffic sign recognition based on pyramid histogram fusion descriptor and HIK-SVM. *J. Transp. Syst. Eng. Inf. Technol.* **2017**, *17*, 220–226.
25. Fang, Z.; Duan, J.; Zheng, B. Traffic signs recognition and tracking based on feature color and SNCC algorithm. *J. Transp. Syst. Eng. Inf. Technol.* **2014**, *14*, 47–52.
26. Liu, K.; Ying, Z.; Cui, Y. SAR image target recognition based on unsupervised K-means feature and data augmentation. *J. Signal Process.* **2017**, *33*, 452–458.
27. Zhang, X.; Fan, J.; Xu, J.; Shi, X. Image super-resolution algorithm via K-means clustering and support vector data description. *J. Image Graph.* **2016**, *21*, 135–144.
28. Ma, G.; Tian, Y.; Li, X. Application of K-means clustering algorithm in color image segmentation of grouper in seawater background. *J. Comput. Appl. Softw.* **2016**, *33*, 192–195.

29. Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [[CrossRef](#)] [[PubMed](#)]
30. Pont-Tuset, J.; Arbeláez, P.; Barron, J.; Marques, F.; Malik, J. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 128–140. [[CrossRef](#)] [[PubMed](#)]
31. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. SSD: Single shot multibox detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 8–16 October 2016; Volume 9905, pp. 21–37.
32. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [[CrossRef](#)] [[PubMed](#)]
33. Couprie, C.; Farabet, C.; Najman, L.; LeCun, Y. Indoor semantic segmentation using depth information. *arXiv*, 2013; arXiv:1301.3572.
34. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from RGB-D Images for object detection and segmentation. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; Volume 8695, pp. 345–360.
35. Petrelli, A.; Pau, D.; Di Stefano, L. Analysis of compact features for RGB-D visual research. In Proceedings of the 18th International Conference on Image Analysis & Processing, Genoa, Italy, 7–11 September 2015; Volume 9280, pp. 14–24.
36. Canny, J.F. A computation approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 769–798.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

A Novel Decentralized Game-Theoretic Adaptive Traffic Signal Controller: Large-Scale Testing

Hossam M. Abdelghaffar ^{1,2} and Hesham A. Rakha ^{3,*}

¹ Department of Computers & Control Systems, Engineering Faculty, Mansoura University, Mansoura, Dakahlia 35516, Egypt; hossamvt@vt.edu

² Center for Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA

³ Charles E. Via, Jr. Dept. of Civil and Environmental Engineering, Director of the Center of Sustainable Mobility, Virginia Tech Transportation Institute, Virginia Tech, Blacksburg, VA 24061, USA

* Correspondence: hrakha@vt.edu

Received: 26 March 2019; Accepted: 13 May 2019; Published: 17 May 2019

Abstract: This paper presents a novel de-centralized flexible phasing scheme, cycle-free, adaptive traffic signal controller using a Nash bargaining game-theoretic framework. The Nash bargaining algorithm optimizes the traffic signal timings at each signalized intersection by modeling each phase as a player in a game, where players cooperate to reach a mutually agreeable outcome. The controller is implemented and tested in the INTEGRATION microscopic traffic assignment and simulation software, comparing its performance to that of a traditional decentralized adaptive cycle length and phase split traffic signal controller and a centralized fully-coordinated adaptive phase split, cycle length, and offset optimization controller. The comparisons are conducted in the town of Blacksburg, Virginia (38 traffic signalized intersections) and in downtown Los Angeles, California (457 signalized intersections). The results for the downtown Blacksburg evaluation show significant network-wide efficiency improvements. Specifically, there is a 23.6% reduction in travel time, a 37.6% reduction in queue lengths, and a 10.4% reduction in CO₂ emissions relative to traditional adaptive traffic signal controllers. In addition, the testing on the downtown Los Angeles network produces a 35.1% reduction in travel time on the intersection approaches, a 54.7% reduction in queue lengths, and a 10% reduction in CO₂ emissions compared to traditional adaptive traffic signal controllers. The results demonstrate significant potential benefits of using the proposed controller over other state-of-the-art centralized and de-centralized adaptive traffic signal controllers on large-scale networks both during uncongested and congested conditions.

Keywords: traffic signal control; game theory; decentralized control; large-scale network control

1. Introduction

Traffic growth and limited available capacity within the roadway system produces problems and challenges for transportation agencies. Traffic congestion affects traveler mobility and has an impact on air quality, and consequently on public health. The stopping and starting in traffic jams burns fuel at a higher rate than the smooth rate of travel, and contributes to the amount of emissions released by vehicles that create air pollution and are related to global warming [1]. Reduction in traffic congestion improves traveler mobility and accessibility, while also reducing vehicle fuel consumption and emissions.

Traffic congestion in 2013 cost Americans \$124.2 billion [2], and this number is projected to rise to \$186.2 billion in 2030. Traffic signal controllers attempt to optimize various traffic variables (e.g., delay, queue length, and energy and emission levels), by optimizing signal control variables, including the cycle length, the phasing scheme and sequence, the phase split, and the offset. Most of the currently

implemented traffic signal systems can be categorized into one of the following categories: fixed-time control (FP), actuated control (ACT), responsive control, or adaptive control [3].

An FP control system is developed off-line using historical traffic data to compute traffic signal timings; real-time traffic data is not taken into account, and the duration and order of all phases stay fixed without any adaptation to real-time traffic demand fluctuations [4]. Previous studies have found this approach to only be appropriate for under-saturated conditions and traffic flows that are stable or relatively stable [5]. By comparison, ACT systems respond to changes in traffic demand patterns by communicating with the controller based on the presence or absence of vehicles as identified by local detectors installed at intersection approach stop lines. While ACT has been proven to generally perform better than FP for very low demand levels, it still offers no real-time optimization to adapt to traffic fluctuations, and may result in long network queues [6]. Adaptive systems have the potential to alleviate traffic congestion by adjusting signal timing parameters in response to real-time traffic fluctuations. These systems use detector inputs, historical trends, and predictive models to predict vehicle arrivals at intersections, and then use the predictions to determine the best gradual changes in cycle length, phase splits, and offsets to minimize vehicle delays or queue lengths [7]. Some examples in this category are: the Split Cycle Offset Optimization Tool (SCOOT) [8], a macroscopic model that minimizes delay and the number of vehicle stops at all intersection approaches, and performs effectively in under-saturated traffic conditions. The Sydney Coordinated Adaptive Traffic System (SCATS) [9] operates in a centralized hierarchical mode, and allocates green times to the phases of greatest need. OPAC [10] optimizes an objective function for a specified rolling horizon using dynamic-programming-based traffic prediction models that require a traffic environment state transition probability model, which can be difficult to generate. TR2 and UTCS-1 [11], optimized off-line, are incapable of handling stochastic variations in traffic patterns.

The operation of actuated and adaptive controllers is constrained by minimum and maximum cycle lengths, green indication durations and offsets, and also require going through a pre-defined sequence of phases. In addition, some systems use hierarchies that either partially or totally centralize decisions, rendering them more susceptible to failures. Hierarchies make scaling these systems up more difficult, relatively more complex to operate, and more expensive [12].

Various computational intelligence-based techniques have been investigated in the domain of traffic signal optimization domain, and are still under continuous research and development, using fuzzy sets, genetic algorithms, reinforcement learning, and neural networks. Genetic algorithms compute the optimal solution using an evolutionary process of possible solutions [13,14]; it solves simple networks and deals with static traffic volumes. However, as the network increases in size, the search space involved in finding effective signal plans increases significantly, and a large amount of centralized computing power is required. Pappis [15] proposed the first signal controller using fuzzy logic for an isolated intersection. Ella [16] proposed a neuro-fuzzy controller, where the parameters of the fuzzy membership functions were adjusted using a neural network. The neural learning algorithm in Ella's work was reinforcement learning, which was found to be successful at constant traffic volumes, but failed when the traffic demand changed rapidly. The choice of the membership functions (building blocks of fuzzy set theory) are important for a particular problem since they affect a fuzzy inference system. As a traffic control system is a complex large-scale system with many interactive factors, it is more appropriate to use fuzzy control for isolated intersections [17].

Several approaches have been proposed for designing traffic signal controllers using neural networks [18,19]. Most of these works are based on a distributed approach, where an agent is assigned to update the traffic signals of a single intersection. Neural networks also adapt very slowly to changing traffic parameters, where on-line learning has to take place continuously. Some networks require multiple models to be maintained for various times within a day. Most intelligence-based approaches are still being researched and are thus under development or have only been implemented and tested on an isolated intersection, so their effectiveness for controlling a large-scale traffic network is also unknown.

Reinforcement learning is inspired by behavioral psychology [20]. It is a machine learning approach which allows agents to interact with the environment, attempting to learn the optimal behavior based on the feedback received from interactions. The feedback may be available right after the action, or several time steps later, which makes the learning more challenging [21]. Abdulhai et al. [22] applied a model-free Q-learning technique to a simple two-phase isolated traffic signal in a two-dimensional road network. Salkham et al. [23] applied a Q-learning strategy that allowed an agent to exchange rewards with its neighbors on 64 signalized intersections. The state-action space was simple and very time coarse. Each agent decided the phase splits every two cycles, which did not capture of the rapid dynamics of congestion–coordination between the agents actions was missing. Studies have considered the use of RL algorithms for traffic control, but they are very limited in terms of network complexity and traffic loadings, so that realistic scenarios, over saturated conditions, and transitions from under saturation to over saturation (and vice versa) have not been fully explored.

Game theory studies the interactive cooperation between intelligent rational decision makers with the specific goal of cooperating and benefiting from reaching a mutually agreeable outcome. It has been widely used in economic, military, communication applications [24,25], model traveler route choice behavior [26], control connected vehicle movements [27], and to in-route guidance [28]. The literature indicates that investigation of game-theoretic traffic signal control is very limited. Bargaining theory is related to cooperative games through the concept of Nash bargaining (NB). A bargaining situation is defined as a situation in which multiple players with specific objectives cooperate and benefit by reaching a mutually agreeable outcome [29]. The bargaining process is the procedure that bargainers follow to reach an agreement (outcome) [30], and the bargaining outcome is the result of the bargaining process [31,32].

Traffic flow is affected by a number of factors, including weather, time-of-day, day-of-week, and unpredictable events, such as special events, incidents, and work zones. Consequently, traffic control strategies could be improved if control systems responded not only to actual conditions, but also adapted their actions to transient conditions. Due to the stochastic nature of traffic flows, an adaptive control strategy that adjusts to stochastic changes is needed. Cycle-free strategies may present an innovative and less restrictive means of accommodating variations in traffic conditions.

Traffic signal controllers can be categorized as centralized or decentralized. Centralized systems require a reliable and direct communication network between a central computer and the local controllers. The main advantage of these systems is that they allow for traffic signal coordination. However, decentralized systems offer many advantages over centralized control systems as they are computationally less demanding and require only relevant information from adjacent intersections/controllers. Robustness is also guaranteed in decentralized control systems, because if one or more controllers fail, the remaining controllers can take over some of their tasks. Decentralized systems are scalable and easy to expand by inserting new controllers into the system. Additionally, decentralized systems are often inexpensive to establish and operate, as there is no essential need for a reliable and direct communication network between a central computer and the local controllers in the field.

To mitigate traffic congestion, a novel de-centralized traffic signal controller, considering a flexible phasing sequence and cycle-free operation, using a NB game-theoretic framework (DNB) is developed. The proposed controller was implemented and evaluated in the INTEGRATION microscopic traffic assignment and simulation software [33–35]. INTEGRATION is a microscopic model that replicates vehicle longitudinal motion using the Rakha–Pasumarthy–Adjerid collision-free car-following model, also known as the RPA model [36]. The RPA model captures vehicle steady-state car-following behavior using the Van Aerde model [37,38]. Movement from one steady state to another is constrained by a vehicle dynamics model described in [39,40]. Vehicle lateral motion is modeled using lane-changing models described in [35]. The model estimates of vehicle delay were validated in [41], while vehicle stop estimation procedures are described and validated in [42]. Vehicle fuel consumption and emissions are modeled using the VT-Micro model [43–45]. The developed controller was compared to the

operation of a decentralized phase split and cycle length controller (PSC) [6], and a fully coordinated adaptive phase split-cycle length and offset optimization controller (PSCO) to evaluate its performance, where PSCO is based on the REALTRAN (REAL-time TRANsynt) controller that emulates the SCOOT system [46,47]. The DNB controller was implemented and evaluated on large-scale networks consisting of 38 (Blacksburg) and 457 (downtown Los Angeles) signalized intersections.

This paper describes the application and the testing of the proposed DNB controller on large-scale networks and is organized as follows. Section 2 describes the developed de-centralized traffic signal controller using a game-theoretic framework. Section 3 presents the experimental setup and results of a large-scale study in the town of Blacksburg, Virginia, consisting of 38 signalized intersections. Section 4 describes the experimental setup and the experimental results of a large-scale study on a downtown network in Los Angeles, California, consisting of 457 signalized intersections. Section 5 presents a summary and conclusions drawn from these studies.

2. Traffic Signal Controller

This section describes the NB solution for two players (Section 2.1), Section 2.2 describes how the NB approach is adapted and extended to control a multi-phase (player) signalized intersection (DNB), and Section 2.3 describes the de-centralized mechanism of the DNB controller over an entire transportation network.

2.1. NB Solution for Two Players

A bargaining situation is defined as a situation in which multiple players with specific objectives cooperate and benefit by reaching a mutually agreeable outcome (agreement). In bargaining theory, there are two concepts: the bargaining process and the bargaining outcome.

The bargaining process is the procedure that bargainers follow to reach an agreement (outcome). Nash adopted an axiomatic approach that abstracts the bargaining process and considers only the bargaining outcome [31]. The bargaining problem consists of three basic elements: players, strategies, and utilities (rewards). Bargaining between two players is illustrated in the bi-matrix shown in Table 1. Each player, namely P_1 and P_2 , has a set of possible actions A_1 and A_2 , whose outcome preferences are given by the utility functions u and v , respectively, as they take relevant actions.

Table 1. Two players matrix game.

| | | P_2 | |
|-------|-------|------------|------------|
| | | A_1 | A_2 |
| P_1 | A_1 | u_1, v_1 | u_2, v_2 |
| | A_2 | u_3, v_3 | u_4, v_4 |

The space (S) shown in Figure 1, is the set of all possible utilities that the two players can achieve; the vertices of the area are the utilities where each player chooses their pure strategy. The disagreement or the threat point $d = (d_1, d_2)$ corresponds to the minimum utilities that the players want to achieve. The threat point is a benchmark, and its selection affects the bargaining solution. Each player attempts to choose their threat point in order to maximize their bargaining position. Subsequently, a bargaining problem is defined as the pair (S, d) where $S \in \mathbb{R}^2$ and $d \in S$ such that S is a convex and compact set, and there exists some $s \in S$ such that $s > d$.

Nash’s theorem states that there exist a unique solution satisfying four axioms (Pareto efficiency, symmetry, invariance to equivalent utility representation, and independence of irrelevant alternatives), and this solution is the pair of utilities (u^*, v^*) that solves the following optimization problem:

$$\begin{aligned} & \max_{u,v} (u - d_1)(v - d_2) \\ & \text{s.t. } (u, v) \in S, (u, v) \geq (d_1, d_2) \end{aligned} \tag{1}$$

The NB solution (u^*, v^*) of this optimization problem can be calculated as the point in the bargaining set that maximizes the product of the players utility gains relative to a fixed threat point.

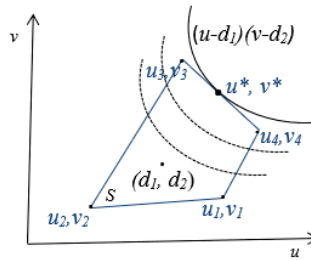


Figure 1. Utility region.

2.2. DNB Traffic Signal Controller for Multi-Players

This section describes the game model and the DNB solution for multi-players (N), and shows how the model is adapted (from Section 2.1) and applied to control a multi-phase signalized intersection. First, a four-phase scheme for a four-legged intersection is used, assuming four players $(N = 4)$, to represent the intersection phases as shown in Figure 2, with protected, leading main street left-turn phases.

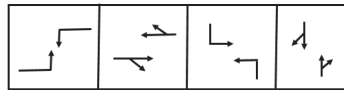


Figure 2. Phasing scheme.

In the game model, the four phases are modeled as four players $P_1, P_2, P_3,$ and P_4 in a four-player cooperative game. For each player (phase), there are two possible actions: maintain (A_1) or change (A_2) . These actions produce the state for the traffic signal. Specifically, the action *maintain* maintains the traffic signal (i.e., if it is displaying a green indication, it will remain green; if it is displaying a red indication, it will remain red). The action *change* entails changing the state of the traffic signal (i.e., if it is displaying a green indication, it will switch its state by first introducing a yellow indication followed by a red indication; if it is red, it will switch to a green indication) in the simulated time interval. The combinations of phases offer four possibilities, where only one player holds the green indication and all others hold red indications [48].

The INTEGRATION software is a microscopic traffic simulation model that traces individual vehicle movements every deci-second. Driver characteristics such as reaction times, acceleration and deceleration levels, desired speeds, and lane-changing behavior are examples of stochastic variables that are modeled in INTEGRATION. The threshold speed is fixed and assigned to the entire network (chosen to be equal to the typical pedestrian speed, $s^{Th} = 4.5$ (km/h)). We continuously check the vehicle speeds when they are within the threat distance from the approach stop bar. If the vehicle (v) speed (s_v^t) is less than the threshold speed (s^{Th}) at time (t), the vehicle is assigned to the queue, and the current queue length associated with the corresponding lane (l) is updated. Once the vehicle's speed exceeds (s^{Th}) the queue length is updated (i.e., shortened by the number of vehicles leaving the queue). This is formulated mathematically as

$$q_l^t = \sum_{v \in v_l^t} q_v^t \tag{2}$$

$$q_v^t = \begin{cases} 1 & \text{if } s_v^{t-1} > s^{Th} \text{ \& } s_v^t \leq s^{Th} \\ -1 & \text{if } s_v^{t-1} \leq s^{Th} \text{ \& } s_v^t > s^{Th} \\ 0 & \begin{cases} \text{if } s_v^{t-1} \leq s^{Th} \text{ \& } s_v^t \leq s^{Th} \\ \text{if } s_v^{t-1} > s^{Th} \text{ \& } s_v^t > s^{Th} \end{cases} \end{cases} \tag{3}$$

q_l^t is the number of queued vehicles in lane l at time t . The index $(t - 1)$ is used to refer to the previous time step. In this case the previous deci-second as the INTEGRATION model tracks vehicle movements at a frequency of 10 hertz.

The utilities (rewards) for each player (phase) in the game can be defined as the estimated sum of the queue lengths in each phase after applying a specific action. The estimated queue length after applying a specific action is calculated according to the following equation:

$$Q_P(t + \Delta t) = \sum_{l \in P} q_l^t + Q_{inl} \Delta t - Q_{outl} \Delta t \tag{4}$$

where Δt is the updating time interval, q_l^t is the current queue length at time t , $Q_P(t + \Delta t)$ is the estimated queue length after Δt for phase P , Q_{inl} is the arrival flow rate (veh/h/lane), and Q_{outl} is the departure flow rate (veh/h/lane).

The NB solution is extended to four players ($N=4$) with a four-dimensional utility space and threat points. The solution for the four-phase NB problem can be formulated as:

$$\begin{aligned} & \max_{(u_1, \dots, u_N)} \prod_{i=1}^N (u_i - d_i) \\ & \text{s.t. } (u_1, \dots, u_N) \in S, (u_1, \dots, u_N) \geq (d_1, \dots, d_N) \end{aligned} \tag{5}$$

The NB solution can be calculated as the vector that maximizes the product of the player’s utility gains relative to a fixed threat point. The threat point represents the maximum number of vehicles that could be accumulated per lane (i.e., the maximum measurable queue length). The objective is to minimize and equalize the queue lengths across the different phases. Hence, the negative queue length is used as the utility of each strategy considering a negative threat point. In other words, the reward (u) is defined to be the negative of the estimated queue length (Q_P), i.e., $u = -Q_P$, and we substitute (d) with a negative number. Consequently, the objective function can be rewritten as follows:

$$\begin{aligned} & \max_{(Q_{P1}, \dots, Q_{PN})} \prod_{i=1}^N (d_i - Q_{Pi}) \\ & \text{s.t. } (Q_{P1}, \dots, Q_{PN}) \in S, (Q_{P1}, \dots, Q_{PN}) \leq (d_1, \dots, d_N) \end{aligned} \tag{6}$$

The block diagram for the DNB controller is shown in Figure 3, where the predefined threat point values are an input to the controller (i.e., the maximum queue size that each player can accommodate). Q_{outl} are generally measured at the approach stop bar, whereas Q_{inl} are measured at a distance from the stop bar equal to the threat point divided by the approach jam density (i.e., the maximum length of the queue assuming all vehicles are stopped).

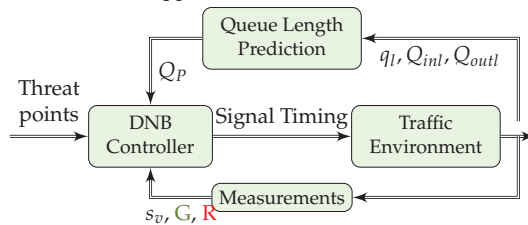


Figure 3. System block diagram.

The flows Q_{inl} and Q_{outl} can be measured using stationary sensors (e.g., loop detectors or through video image processor (VIP) detection obtained from CCTV cameras). The queue length estimates can be obtained using CCTV cameras or via GPS-equipped vehicles that communicate with the traffic signal controller. As such, the proposed controller is technology agnostic.

2.3. DNB Controller for Multi-Intersections

This section presents the DNB controller formulation for a network composed of multiple signalized intersections. For illustration purposes only, we formulate the problem considering three signalized intersections, as shown in Table 2. It should be noted, however that the algorithm can operate on a network of any number of signalized intersections.

Assume we have three signalized intersections (I_1, I_2, I_3), each traffic signal has three phases (Ph_1, Ph_2, Ph_3), where each phase is modeled as a player in a game resulting in a total of nine players where I_1 has three players (P_1, P_2, P_3), I_2 has three players (P_4, P_5, P_6), and I_3 has three players (P_7, P_8, P_9). Each traffic signal has three possible actions (A), where one phase displays a green indication (G) while the others display a red indication (R), as illustrated in Table 2.

Table 2. Multi-player matrix game.

| Intersection | First Intersection (I_1) | | | Second Intersection (I_2) | | | Third Intersection (I_3) | | |
|--------------|------------------------------|--------------|--------------|-------------------------------|--------------|--------------|------------------------------|--------------|--------------|
| Player | Ph1(P_1) | Ph2(P_2) | Ph3(P_3) | Ph1(P_4) | Ph2(P_5) | Ph3(P_6) | Ph1(P_7) | Ph2(P_8) | Ph3(P_9) |
| Action | Ph1(P_1) | Ph2(P_2) | Ph3(P_3) | Ph1(P_4) | Ph2(P_5) | Ph3(P_6) | Ph1(P_7) | Ph2(P_8) | Ph3(P_9) |
| First | G | R | R | G | R | R | G | R | R |
| | A_{11} | | | A_{21} | | | A_{31} | | |
| Second | R | G | R | R | G | R | R | G | R |
| | A_{12} | | | A_{22} | | | A_{32} | | |
| Third | R | R | G | R | R | G | R | R | G |
| | A_{13} | | | A_{23} | | | A_{33} | | |

Consequently, for the three signalized network illustrated in Table 2, there are 27 possible scenarios (action permutations) as shown in Table 3. The optimum overall network performance (NB optimum, Equation (6)) can be computed from Table 3.

Referring to Table 2, and assuming that the first traffic signal (I_1) has action (A_{12}) that optimizes its performance, traffic signal (I_2) has action (A_{21}) that optimizes its performance, and traffic signal (I_3) has action (A_{33}) that optimizes its performance. Consequently, searching in Table 3 for the Nash optimum combination yields scenario 12. This implies that in order to achieve the Nash optimum network performance, it is sufficient to search for the actions that optimize the operations of each signalized intersection. This can be described using the NB optimization problem shown in the following equations.

$$\begin{aligned}
 & \max_{(u_1, \dots, u_9)} \prod_{i=1}^9 (u_i - d_i) \\
 = & \max_{(u_1, \dots, u_9)} \underbrace{\prod_{i=1}^3 (u_i - d_i)}_{I_1} \underbrace{\prod_{i=4}^6 (u_i - d_i)}_{I_2} \underbrace{\prod_{i=7}^9 (u_i - d_i)}_{I_3} \tag{7} \\
 = & \underbrace{\max_{(u_1, \dots, u_3)} \prod_{i=1}^3 (u_i - d_i)}_{I_1} \underbrace{\max_{(u_4, \dots, u_6)} \prod_{i=4}^6 (u_i - d_i)}_{I_2} \underbrace{\max_{(u_7, \dots, u_9)} \prod_{i=7}^9 (u_i - d_i)}_{I_3}
 \end{aligned}$$

The network-wide Nash optimum solution is obtained by maintaining the Nash optimum solution at each signalized intersection. As such, while the proposed NB controller is decentralized (i.e., DNB), it still produces the network-wide Nash-optimum control strategy relying solely on edge computing. The Nash optimum should not be mistaken for the system-optimum solution, where the system optimum might sacrifice the performance of one or more traffic signals to achieve optimum network-wide performance. It should be noted that obtaining the system-optimum solution is impossible given the scale and level of interactions of the various network-wide traffic signal controllers. The DNB controller, thus, provides a scalable and resilient controller that circumvents the problems inherent in complex centralized systems with minimum sacrifices to network-wide performance.

Table 3. All possible Network Actions (Permutations).

| Scenario # | Network Action |
|------------|------------------------|
| 1 | $A_{11} A_{21} A_{31}$ |
| 2 | $A_{11} A_{21} A_{32}$ |
| 3 | $A_{11} A_{21} A_{33}$ |
| 4 | $A_{11} A_{22} A_{31}$ |
| 5 | $A_{11} A_{22} A_{32}$ |
| 6 | $A_{11} A_{22} A_{33}$ |
| 7 | $A_{11} A_{23} A_{31}$ |
| 8 | $A_{11} A_{23} A_{32}$ |
| 9 | $A_{11} A_{23} A_{33}$ |
| 10 | $A_{12} A_{21} A_{31}$ |
| 11 | $A_{12} A_{21} A_{32}$ |
| 12 | $A_{12} A_{21} A_{33}$ |
| 13 | $A_{12} A_{22} A_{31}$ |
| 14 | $A_{12} A_{22} A_{32}$ |
| 15 | $A_{12} A_{22} A_{33}$ |
| 16 | $A_{12} A_{23} A_{31}$ |
| 17 | $A_{12} A_{23} A_{32}$ |
| 18 | $A_{12} A_{23} A_{33}$ |
| 19 | $A_{13} A_{21} A_{31}$ |
| 20 | $A_{13} A_{21} A_{32}$ |
| 21 | $A_{13} A_{21} A_{33}$ |
| 22 | $A_{13} A_{22} A_{31}$ |
| 23 | $A_{13} A_{22} A_{32}$ |
| 24 | $A_{13} A_{22} A_{33}$ |
| 25 | $A_{13} A_{23} A_{31}$ |
| 26 | $A_{13} A_{23} A_{32}$ |
| 27 | $A_{13} A_{23} A_{33}$ |

Note that a single traffic signal cannot be decomposed (i.e., optimize each decision variable independently), as the utilities of the players within the same traffic signal are dependent on each other. Specifically, if one player displays a green indication by default the other players have to display a red indication given that this would result in conflicting movements being discharged simultaneously. Alternatively, each traffic signal controller operates independently. Consequently, decomposition is invalid within a traffic signal but valid between traffic signals, as players within a traffic signal compete for the same resource, namely green time.

3. Blacksburg Town Experiments

This section presents the experimental setup and the results of a testing of the proposed system in the town of Blacksburg, Virginia, illustrated in Figure 4. The simulations were conducted using the morning peak hour (7–8 a.m.) traffic demand. The town of Blacksburg has 38 signalized intersections, 549 stop signs, 30 yield signs, and 1844 links. The minimum free-flow speed on the network was 30 (km/h), and the maximum free-flow speed on the network was 105 (km/h).

The minimum link length was 50 m while the maximum link length was 2932 m. The jam density was set at 160 (veh/km/lane). The traffic signal phasing scheme used in the study was the same as those in the field. These varied between 2 to 4 phases.

3.1. Blacksburg Experimental Setup

The time-dependent static O-D demand matrices were generated every 15 min using the QueensOD software [49–51]. QueensOD estimates the most likely O-D matrix that is as close structurally as a seed matrix while at the same time minimizing the error between the estimated and field observed link flow counts. The time-dependent static O-Ds were then used to compute a dynamic O-D matrix using procedures described in [52]. The final peak-hour dynamic O-D matrix consisted of 23,260 vehicular trips. Vehicles were loaded for one hour, while the simulation continued until all vehicles cleared the network to ensure that the same number of vehicles were used in comparing the performance of the various traffic signal control algorithms.

The performance of the DNB controller was evaluated by comparing its performance to that of the PSC and PSCO controllers. The network-wide average of each of the following measures of effectiveness (MOEs) was calculated to assess the DNB controller's performance: travel time, total delay, stopped delay, queue length, fuel consumption, and emission levels. The INTEGRATION microscopic traffic assignment and simulation software was used to model the network, shown in Figure 4. Three experiments were conducted on the BB network, as discussed in the following sections.

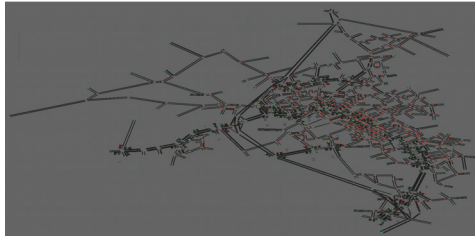


Figure 4. Blacksburg network.

3.2. BB Experimental Results: 1

In this experiment the performance of the DNB controller was compared to the PSC and PSCO controllers. The threat point (d) values per lane for the DNB controller were assigned based on the link's lengths (L), the link's free-flow speeds (U_f), and the updating time intervals (Δt), using the following formula; $d = \min[N(L/2), N(U_f \times \Delta t)]$, where $N(L/2)$ represents the number of vehicles that could be accumulated up to the half length of the link, and $N(U_f \times \Delta t)$ represents the maximum number of stopped vehicles that could be stored in the distance ($U_f \times \Delta t$). Using this distance allowed vehicles to proceed through the intersection in a minimal time without stopping if there was no queue ahead of them. A distance of $L/2$ was used instead of L to get a better estimate of the queue length for each movement because drivers typically moved to their desired lanes as they got closer to the signalized intersection, and to avoid being fully queued (i.e., players will accept a fully occupied (queued) link).

The average MOE values over the entire simulation for the PSC, PSCO, and DNB control scenarios are summarized in Table 4. In addition, Table 4 shows the percent improvement in MOEs using the proposed DNB controller over the PSC and PSCO controllers. The improvement (%) is calculated as:

$$\text{Improvement}(\%) = \frac{\text{MOE}(\text{PSC}/\text{PSCO}) - \text{MOE}(\text{DNB})}{\text{MOE}(\text{PSC}/\text{PSCO})} \times 100 \quad (8)$$

Table 4. Average measures of effectiveness (MOEs) and (%) improvement for game-theoretic framework (DNB) over phase split and cycle length controller (PSC) and phase split-cycle length and offset optimization controller (PSCO) controllers.

| | System | | |
|---------------------------------------|---------|---------|---------|
| MOE | PSC | PSCO | DNB |
| Average Total Delay (s/veh) | 96.234 | 100.197 | 80.323 |
| Improvement % | 16.534 | 19.823 | |
| Average Stopped Delay (s/veh) | 20.285 | 25.649 | 12.1074 |
| Improvement % | 40.314 | 52.7962 | |
| Average Travel time (s) | 306.254 | 310.225 | 290.175 |
| Improvement % | 5.25 | 6.46 | |
| Average Number of Stops | 4.662 | 4.5899 | 4.281 |
| Improvement % | 8.18 | 6.734 | |
| Average Fuel (L) | 0.4142 | 0.4129 | 0.40 |
| Improvement % | 3.38 | 3.07 | |
| Average CO ₂ Emissions (g) | 913.833 | 912.495 | 883.127 |
| Improvement % | 3.36 | 3.22 | |

The simulation results demonstrated a significant reduction in the average travel time of 5.25%, a reduction in the average total delay of 16.5%, and a reduction in the average stopped delay of 40.3% over the PSC controller. In addition, the results indicated significant reduction in the average travel time of 6.5%, a reduction in the average total delay of 19.8%, and a reduction in the average stopped delay of 52.7% over the PSCO controller. These results show that the proposed DNB controller outperforms both the PSC and PSCO controllers.

3.3. BB Experimental Results: 2

This section presents a potential solution to better estimate the queue length considering the driver’s lane changing behavior close to the intersections. A suggested phasing scheme, shown in Figure 5b, where all vehicles on the link discharge in a single phase, might provide a better estimate of the queue length per phase over the currently implemented phase scheme shown in Figure 5a, where each link discharges in two phases. Two simulations were conducted using the DNB controller to evaluate the effectiveness of the two phasing scheme on the MOEs, where the threat point per lane was assigned using the following formula: $d = \min[N(L/2), N(U_f \times \Delta t)]$. The simulation results using the two schemes (Figure 5) are shown in Table 5.

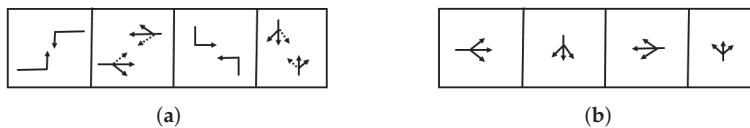


Figure 5. Four phasing scheme. (a) Implemented phasing scheme. (b) Suggested phasing scheme.

Table 5. MOEs using two different phasing schemes.

| | System | | |
|---------------------------------------|--------------------|-----------------------|----------|
| MOE | DNB (Field Scheme) | DNB (Modified Scheme) | Imp. (%) |
| Average Total Delay (s/veh) | 80.323 | 94.712 | −17.913 |
| Average Stopped Delay (s/veh) | 12.107 | 24.381 | −101.374 |
| Average Travel Time (s) | 290.175 | 302.425 | −4.222 |
| Average Number of Stops | 4.281 | 4.417 | −3.177 |
| Average Fuel (L) | 0.40 | 0.41 | −2.274 |
| Average CO ₂ Emissions (g) | 883.127 | 902.277 | −2.168 |

The simulation results demonstrate that the suggested phasing scheme does not improve the network performance.

3.4. BB Experimental Results: 3

This section presents the effect of reducing the number of vehicles that can be accumulated in a lane on the network's performance. The minimum free-flow speed on the network was 30 (km/h), and the maximum free-flow speed on the network was 105 (km/h), with updating time intervals of 10 s. Assigning the detector locations to be the $\min(L/2, U_f \times \Delta t)$, the detectors could be located for long links between 84 m (i.e., 13 veh/lane) to 292 m (i.e., 47 veh/lane). Employing the free-flow speed to determine the threat point ($d = \min[N(L/2), N(U_f \times \Delta t)]$) is a good choice for low traffic demand, as vehicles are not required to stop at the intersection; however, for high traffic demand, long links can accommodate long queues, which causes delays for the vehicles on that link. Hence, reducing the number of vehicles that can accumulate in a lane might enhance the network's performance. To examine the effectiveness of changing the maximum number of vehicles that could be accumulated per lane on the MOEs, a sensitivity analysis was conducted, as shown in Figure 6, with $d = \min[N(L/2), NV]$, where NV presents the maximum number of vehicles that can be stored in a lane; this number ranges between 6 to 32 vehicles.

Analysis of the results in Figure 6 demonstrated that better performance using the DNB controller could be achieved if the threat points are assigned as a minimum of 12 veh/lane and the number of vehicles that could be accumulated in $L/2$, ($d = \min[N(L/2), 12]$).

Table 6 shows the average MOEs values over the entire simulation time and the percent improvement in MOEs using the proposed DNB controller over PSC and PSCO controllers. Simulation results indicate significant reduction in the average total delay of 19.38%, a reduction in the average stopped delay of 51.18%, a reduction in the average travel time of 6.162%, a reduction in the average number of stops of 8.39%, a reduction in the average fuel consumption of 3.89%, and a reduction in the emission levels of 3.84% over the PSC controller. The results show that the proposed DNB approach outperforms both the PSC and PSCO controllers.

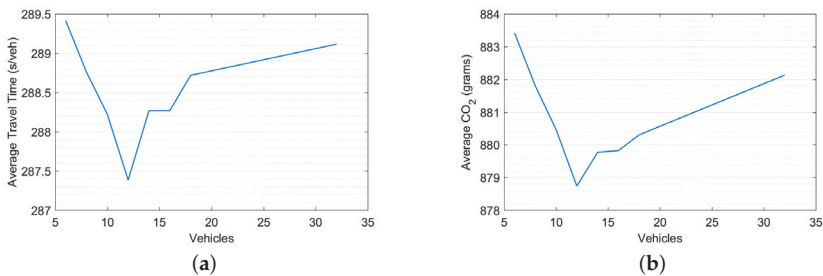


Figure 6. Sensitivity analysis. (a) Average travel time. (b) Average CO₂.

Table 6. Average MOEs and (%) improvement using DNB over the PSC and PSCO controllers.

| MOE | System | | |
|---------------------------------|---------|---------|---------|
| | PSC | PSCO | DNB |
| Average Total Delay (s/veh) | 96.234 | 100.197 | 77.577 |
| Improvement % | 19.3871 | 22.575 | |
| Average Stopped Delay (s/veh) | 20.285 | 25.649 | 9.903 |
| Improvement % | 51.182 | 61.391 | |
| Average Travel Time (s) | 306.254 | 310.225 | 287.384 |
| Improvement % | 6.162 | 7.362 | |
| Average Number of Stops | 4.662 | 4.5899 | 4.271 |
| Improvement % | 8.393 | 6.95 | |
| Average Fuel (L) | 0.4142 | 0.4129 | 0.3981 |
| Improvement % | 3.887 | 3.584 | |
| Average CO ₂ (grams) | 913.833 | 912.495 | 878.739 |
| Improvement % | 3.84 | 3.7 | |

To further investigate the achieved improvements using the DNB controller, it was taken into consideration that the network has 459 stop signs and 30 yield signs, which might conceal the full degree of improvement achieved using the DNB controller on the signalized intersection. Accordingly, we investigated the percent improvement in MOEs using the DNB controller over the PSC controller over only the links that were directly associated with intersections. Table 7 shows the percent improvement in MOEs using the DNB controller over the PSC controller on the 38 intersections.

Table 7 demonstrates an improvement in the travel time on the intersections between 6% to 52%, an improvement in the queue length on the intersections between 8% to 60%, and an improvement in the number of stops on the intersections between 8% to 80%. In addition, Table 7 demonstrates an overall reduction in the average travel time of 23.63%, in the average queued vehicles of 37.66%, in the average number of stops of 23.58%, in the average fuel consumption of 10.44%, in the average CO₂ emitted of 9.84%, and in the average NO_x emitted of 5.4% over the PSC controller. These results revealed that the DNB controller performs significantly better than the PSC controller.

Table 7. Intersections (%) improvement of MOEs using DNB over PSC controller.

| MOEs Int. # | Travel Time | Queue | Num. of Stops | CO ₂ | Fuel | NO _x |
|--------------------|--------------|--------------|---------------|-----------------|-------------|-----------------|
| 1 | 6.15 | 22.02 | 24.31 | 2.65 | 2.57 | 0.16 |
| 2 | 16.41 | 26.80 | 21.18 | 7.71 | 7.71 | 5.86 |
| 3 | 8.49 | 18.23 | 32.78 | 6.03 | 6.45 | 9.04 |
| 4 | 31.11 | 52.87 | 39.56 | 8.17 | 6.60 | 8.76 |
| 5 | 22.23 | 53.88 | 52.91 | 9.36 | 8.96 | 3.31 |
| 6 | 23.18 | 34.44 | 14.24 | 11.59 | 10.72 | 4.75 |
| 7 | 8.97 | 15.88 | 17.83 | 3.89 | 3.60 | 2.27 |
| 8 | 24.06 | 41.87 | 16.11 | 13.75 | 13.48 | 9.16 |
| 9 | 40.71 | 56.27 | 29.85 | 25.25 | 24.65 | 13.84 |
| 10 | 13.40 | 26.35 | 41.44 | 8.63 | 8.65 | 9.77 |
| 11 | 17.63 | 26.34 | 11.80 | 9.01 | 8.35 | 1.35 |
| 12 | 7.64 | 7.97 | 32.65 | 3.48 | 3.37 | 3.48 |
| 13 | 19.41 | 37.91 | 20.92 | 8.99 | 8.75 | 3.76 |
| 14 | 28.50 | 35.50 | 25.36 | 7.85 | 6.62 | 8.15 |
| 15 | 23.87 | 39.63 | 34.58 | 12.55 | 12.27 | 6.17 |
| 16 | 27.55 | 59.10 | 41.88 | 15.11 | 14.79 | 8.84 |
| 17 | 42.00 | 60.00 | 56.97 | 16.90 | 14.83 | 12.84 |
| 18 | 26.26 | 49.88 | 32.72 | 14.49 | 13.41 | 5.70 |
| 19 | 19.68 | 36.53 | 21.10 | 4.96 | 4.25 | 4.98 |
| 20 | 52.24 | 76.08 | 63.09 | 32.97 | 31.76 | 20.16 |
| 21 | 34.82 | 50.16 | 46.27 | 21.57 | 21.39 | 18.27 |
| 22 | 38.27 | 59.40 | 37.47 | 27.63 | 27.28 | 26.53 |
| 23 | 17.19 | 30.86 | 16.27 | 7.60 | 6.92 | 5.26 |
| 24 | 34.67 | 44.00 | 11.27 | 14.63 | 13.34 | 3.24 |
| 25 | 23.48 | 44.59 | 57.38 | 5.76 | 4.50 | 0.09 |
| 26 | 18.03 | 26.03 | 30.50 | 4.02 | 2.48 | 0.75 |
| 27 | 28.13 | 36.34 | 8.57 | 16.77 | 16.19 | 14.48 |
| 28 | 14.53 | 35.05 | 11.90 | 9.46 | 9.85 | 11.61 |
| 29 | 13.13 | 19.12 | 9.60 | 5.35 | 4.99 | 1.14 |
| 30 | 23.63 | 47.38 | 23.22 | 19.33 | 19.41 | 24.77 |
| 31 | 32.76 | 55.70 | 80.38 | 18.00 | 17.27 | 19.33 |
| 32 | 34.76 | 53.07 | 35.46 | 26.64 | 27.05 | 29.31 |
| 33 | 35.98 | 48.47 | 15.26 | 20.35 | 19.56 | 11.67 |
| 34 | 16.68 | 32.68 | 30.34 | 11.27 | 11.15 | 11.76 |
| 35 | 18.01 | 28.95 | 21.58 | 18.24 | 18.67 | 26.12 |
| 36 | 22.59 | 46.51 | 34.33 | 7.68 | 7.03 | 2.47 |
| 37 | 29.31 | 46.50 | 31.49 | 7.40 | 6.68 | 1.08 |
| 38 | 14.32 | 14.55 | 8.06 | 4.67 | 4.17 | 1.14 |
| Overall (%) | 23.63 | 37.67 | 23.59 | 10.44 | 9.84 | 5.39 |

4. Downtown Los Angeles Experiments

This section describes the experimental setup and the experimental results of large scale studies in downtown Los Angeles, California comprised of 457 signalized intersections.

4.1. Los Angeles Experimental Setup

These experiments were large scale studies of a network in downtown Los Angeles (LA), California, including the most congested downtown area, as shown in Figure 7a. The INTEGRATION microscopic traffic assignment and simulation software was used to model the network, as shown in Figure 7b.

Simulations were conducted using the morning peak hour (7–8 a.m.) traffic demand that was calibrated in a previous effort [53]. The downtown LA network has 457 signalized intersections, 285 stop signs, 23 yield signs, and 3556 links. The origin-destination (O-D) demand matrices were generated, as described earlier, using a combination of the QueensOD software, to generate time-dependent static O-D demands, and then converting these static O-D demands to a dynamic O-D

demand. The resulting O-D consisted of a total of 143,957 vehicle trips. Vehicles were loaded for the one-hour period and the simulation continued until all vehicles cleared the network to ensure that all comparisons were made for the same number of vehicles.

The traffic signal phasing schemes varied from 2 to 6 phases, reflecting the field implemented traffic signal settings in downtown LA. The minimum free-flow speed on the network was 15 (km/h), and the maximum free-flow speed on the network was 120 (km/h). The minimum link length on the network was 50 m, and the maximum link length on the network was 4400 m. The jam density of the various network links was set equal to 180 (veh/km/lane).



Figure 7. Downtown Los Angeles network. (a) LA, Google maps. (b) LA, INTEGRATION.

The DNB controller was compared to the PSC controller to evaluate their relative performance. The average of each of the following measures of effectiveness (MOEs) was calculated to assess the performance of the DNB controller: travel time, total delay, stopped delay, queue length, fuel consumption, and emission levels.

4.2. LA Experimental Results: 1

In this experiment, the performance of the DNB controller was compared to that of the PSC controller using the full traffic demand in the morning peak hour. The threat point per lane for the DNB controller was assigned as the minimum of 12 veh/lane and the number of vehicles that could be accumulated on $L/2$ (i.e., $d = \min[N(L/2), 12]$) based on the sensitivity analysis shown in Figure 8.

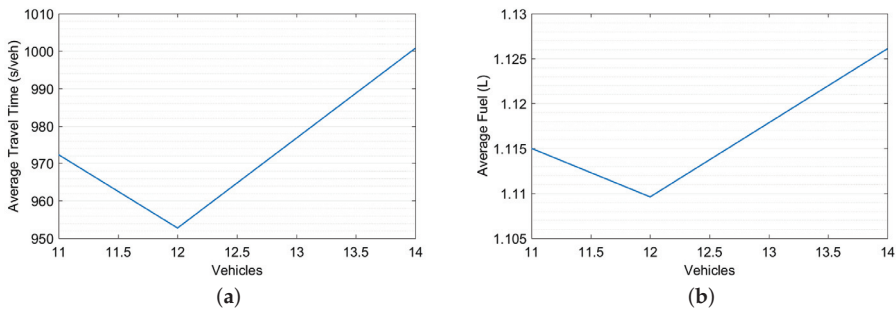


Figure 8. LA Sensitivity Analysis. (a) Average Travel Time. (b) Average Fuel Consumption.

The average MOE values over the entire simulation for the PSC and DNB controllers are shown in Table 8. In addition, Table 8 shows the percent improvement in MOEs using the proposed DNB controller relative to the PSC controller. The simulation results demonstrate a significant reduction in

the average travel time of 7.89%, a reduction in the total delay of 14.55%, a reduction in the average stopped delay of 25.18%, a reduction in the average number of vehicle stops of 12.4%, a reduction in the average fuel consumption of 4.0%, and a reduction in CO₂ emission levels of 4.25%, relative to the PSC controller. Analysis of the results demonstrated that the proposed DNB controller outperforms current state-of-the-art de-centralized traffic signal controllers.

Table 8. Average MOEs and the (%) improvement using DNB controller over PSC controller (100% Demand).

| MOE | System | | |
|---------------------------------|---------|---------|--------------|
| | PSC | DNB | DNB Imp. (%) |
| Average Total Delay (s/veh) | 557.46 | 476.35 | 14.55 |
| Average Stopped Delay (s/veh) | 256.77 | 192.12 | 25.18 |
| Average Travel Time (s) | 1034.27 | 952.73 | 7.89 |
| Average Number of Stops | 7.41 | 6.49 | 12.40 |
| Average Fuel (L) | 1.16 | 1.11 | 4.00 |
| Average CO ₂ (grams) | 2482.13 | 2376.59 | 4.25 |

The improvements produced by the DNB controller, only at the signalized intersections, were further analyzed. Accordingly, we investigated the percent improvement in MOEs using the DNB controller over the PSC controller over only the links that were directly associated with signalized intersections.

Table 9 demonstrates a reduction in the average travel time of 35.16%, a reduction in the average queued vehicles of 54.67%, a reduction in the average number of stops of 44.03%, a reduction in the average fuel consumption of 9.97%, a reduction in the CO₂ emissions of 9.92%, and a reduction in the NO_x emissions of 11.78% relative to the PSC controller. These results revealed that the DNB controller has significantly better performance potential than the PSC controller.

Table 9. Average (%) improvements of MOEs using DNB controller over PSC controller (100% Demand), over the links that are directly associated with intersections.

| Int. # | MOEs | | | | | |
|----------------------|-------------|-------|---------------|-----------------|------|-----------------|
| | Travel Time | Queue | Num. of Stops | CO ₂ | Fuel | NO _x |
| Overall 457 Int. (%) | 35.16 | 54.66 | 44.03 | 9.97 | 9.92 | 11.77 |

4.3. LA Experimental Results: 2

A simulation was conducted for lower levels of traffic congestion by scaling the demand down by 90% (i.e., 10% of the peak demand) to investigate the performance potential using the DNB controller. Table 10 shows a reduction in the average travel time of 7.1%, a reduction in the average total delay of 36.79%, a reduction in the average stopped delay of 90.26%, a reduction in the average number of vehicle stops of 34.66%, a reduction in the average fuel consumption of 4.8%, and a reduction in CO₂ emission levels of 4.79%, relative to the PSC controller.

Table 10. Average MOEs and the (%) improvement using DNB over PSC controller (10% Demand).

| MOE | System | | |
|---------------------------------|---------|---------|--------------|
| | PSC | DNB | DNB Imp. (%) |
| Average Total Delay (s/veh) | 84.94 | 53.69 | 36.79 |
| Average Stopped Delay (s/veh) | 19.97 | 1.95 | 90.26 |
| Average Travel Time (s) | 450.11 | 418.18 | 7.10 |
| Average Number of Stops | 4.48 | 2.92 | 34.66 |
| Average Fuel (L) | 0.85 | 0.81 | 4.80 |
| Average CO ₂ (grams) | 1830.27 | 1742.53 | 4.79 |

Once more, to further investigate the achieved improvements using the DNB controller, we investigated the improvement in MOEs over only the links that were directly associated with

signalized intersections, as shown in Table 11. Table 11 demonstrates a reduction in the average travel time of 19.19%, a reduction in the average queued vehicles of 49.84%, a reduction in the average number of stops of 53.71%, a reduction in the average fuel consumption of 54.16%, a reduction in the average CO₂ emitted of 16.09%, and a reduction in the average NO_x emitted of 25.94% over PSC controller.

These results demonstrate that the DNB controller performed significantly better than the PSC controller in both congested and uncongested conditions, however, produced more savings as the traffic demand decreased.

Table 11. Average (%) improvements of MOEs using DNB over PSC controller (10% Demand) over the links directly associated with intersections.

| Int. # | MOEs | Travel Time | Queue | Num. of Stops | CO ₂ | Fuel | NO _x |
|----------------------|------|-------------|-------|---------------|-----------------|-------|-----------------|
| Overall 457 Int. (%) | | 19.19 | 49.84 | 53.71 | 54.16 | 16.09 | 25.94 |

The results show that the DNB controller yielded significant improvements in the average values of all MOEs, demonstrating improved system efficiency.

5. Summary & Conclusions

The research presented in this paper develops and evaluates a Nash bargaining de-centralized flexible phasing cycle-free traffic signal controller (DNB controller) on large-scale networks. The controller was implemented and tested in the INTEGRATION microscopic traffic assignment and simulation software. The performance of the DNB controller was compared to a decentralized phase split and cycle length optimization controller based on the HCM procedures (PSC) and a fully-coordinated adaptive phase split, cycle length and offset optimization controller (PSCO), in the town of Blacksburg, Virginia and in downtown Los Angeles, California.

Several simulations were conducted on the Blacksburg network using different threat point values and phasing schemes to determine their effect on the controller's performance. The results show significant reductions in the network-wide average travel time of 6.1% and 7.3%, a reduction in the average total delay of 19.3% and 22.6%, a reduction in the stopped delay of 51% and 61%, and a reduction in CO₂ emission levels of 3.8% and 3.7%, over the PSC and PSCO controllers, respectively. In addition, the results show significant reductions on the intersection approach average travel time of 23.6%, a reduction in the average queue length of 37.6%, a reduction in the average number of vehicle stops of 23.6%, a reduction in the fuel consumption of 9.8%, a reduction in the CO₂ emissions of 10.4%, and a reduction in NO_x emissions of 5.4%.

In addition, the DNB controller's performance was tested in downtown Los Angeles, California, and compared to the performance of the de-centralized PSC controller. The results show significant improvements in various network-wide measures of performance. Specifically, a reduction in the average travel time of 8%, a reduction in the average total delay of 14.5%, a reduction in the stopped delay of 25.1%, a reduction in the average number of vehicle stops of 12.4%, and a reduction in CO₂ emissions of 4.25%, over the PSC controller. Moreover, the results show significant improvements in the signalized intersection operations with a reduction in the average travel time of 35.1%, a reduction in the average queue length of 54.7%, a reduction in the average number of vehicle stops of 44%, a reduction in the fuel consumption and CO₂ emissions of 10%, and a reduction in NO_x emissions of 11.7%. Furthermore, simulations conducted for lower traffic demand levels showed significant network-wide improvements with a reduction in the average total delay of 36.7%, a reduction in the stopped delay of 90.2%, and a reduction in the average number of stops of 35% over the PSC controller. As these results indicate, the DNB controller can generate major performance improvements at lower demands. The results demonstrate significant potential benefits of using the proposed controller over other state-of-the-art centralized and de-centralized controllers on large scale networks.

In summary, a novel traffic signal controller is developed that offers a number of unique features. First, the controller adapts signal timings dynamically to changing traffic conditions without using historical data, which tends to be inaccurate, resulting in inefficient traffic signal plans. Second, the developed controller is de-centralized, which increases both the scalability and robustness of the system, to avoid the problems inherent with complex centralized communication. Decentralized systems are often inexpensive to establish and operate, as there is no essential need for a reliable and direct communication network between a central computer and the local controllers in the field. Third, the controller, while de-centralized, does not sacrifice in system-wide performance and computes the network-wide Nash optimum solution. Finally, the controller is designed to operate with current traffic signal controllers. This controller should increase the traffic handling capacity of roads, and reduce unnecessary stop-and-go vehicular movement, which will reduce fuel consumption and, accordingly, air pollution.

Author Contributions: The work described in this article is the collaborative development of all authors, conceptualization, H.M.A. and H.A.R.; methodology, H.M.A. and H.A.R.; software, H.M.A. and H.A.R.; validation, H.M.A. and H.A.R.; formal analysis, H.M.A. and H.A.R.; investigation, H.M.A. and H.A.R.; writing—review and editing, H.M.A. and H.A.R.

Funding: This effort was funded by the US Department of Transportation through the University Mobility and Equity Center (Award 69A3551747123).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Elbakary, M.I.; Abdelghaffar, H.M.; Afrifa, K.; Rakha, H.A.; Cetin, M.; Iftekharuddin, K.M. Aerosol Detection Using Lidar-Based Atmospheric Profiling. In *SPIE Optics and Photonics for Information Processing XI*; SPIE: Bellingham, WA, USA, 2017; doi:10.1117/12.2275626.
2. *The future economic and environmental costs of gridlock in 2030*; Technical Report; Center for Economics and Business Research: London, UK, 2014.
3. Dai, Y.; Zhao, D.; Zhang, Z. Computational Intelligence in Urban Traffic Signal Control: A Survey. *IEEE Trans. Syst. Man, Cybern.* **2012**, *42*, 485–494, doi:10.1109/TSMCC.2011.2161577. [[CrossRef](#)]
4. Turkey, A.M.; Ahmad, M.S.; Yusoff, M.; Hammad, B.T. Using genetic algorithm for traffic light control system with a pedestrian crossing. In Proceedings of the 4th International Conference, RSKT, Gold Coast, Australia, 14–16 July 2009; doi:10.1007/978-3-642-02962-2_65.
5. Yang, X. Comparison Among Computer Packages in Providing Timing Plans for Iowa Arterial in Lawrence, Kansas. *J. Transp. Eng.* **2001**, *127*, 311–318, doi:10.1061/(ASCE)0733-947X(2001)127:4(311). [[CrossRef](#)]
6. Roess, R.; Prassas, E.S.; McShane, W.R. *Traffic Engineering*, 4th ed.; Pearson Higher Education, Inc.: Upper Saddle River, NJ, USA, 2010.
7. French, L.J.; French, M.S. *Benefits of Signal Timing Optimization and ITS to Corridor Operations*; Technical Report; French Engineering, LLC: Spring, TX, USA, 2006.
8. Hunt, P.B.; Robertson, D.I.; Bretherton, R.D.; Winton, R.I. *SCOOT-A Traffic Responsive Method of Coordinating Signals*; Technical Report; Transport and Road Research Laboratory: Wokingham, UK, 1981.
9. Sims, A.G.; Dobinson, K.W. SCAT-The Sydney Co-ordinated Adaptive Traffic System Philosophy and Benefits. In Proceedings of the International Symposium on Traffic Control Systems, Berkeley, CA, USA, 6–9 August 1979.
10. Gartner, N.H. OPAC: A demand-responsive strategy for traffic signal control. *Transp. Res. Rec. J. Transp. Res. Board* **1983**, *906*, 75–81.
11. MacGowan, J.; Fullerton, I.J. Development and testing of advanced control strategies in the urban traffic control system. *Public Roads* **1979**, *43*, 97–105.
12. Evans, M.R. *Balancing Safety and Capacity in an Adaptive Signal Control System—Phase 1*; Technical Report FHWA-HRT-10-038; Federal Highway Administration: Washington, DC, USA, 2010.
13. Ceylan, H.; Bell, M.G.H. Traffic signal timing optimization based on genetic algorithm approach, including driver's routing. *Transp. Res. Part B* **2004**, *38*, 329–342, doi:10.1016/S0191-2615(03)00015-8. [[CrossRef](#)]

14. Chin, Y.; Yong, K.; Bolong, N.; Yang, S.; Teo, K. Multiple Intersections Traffic Signal Timing Optimization with Genetic Algorithm. In Proceedings of the IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 25–27 November 2011; doi:10.1109/ICCSCE.2011.6190569.
15. Pappis, C.; Mamdani, E. A Fuzzy Logic Controller for a Traffic Junction Systems. *IEEE Trans. Man Cybern.* **1977**, *7*, 707–717, doi:10.1109/TSMC.1977.4309605. [[CrossRef](#)]
16. Bingham, E. *Neurofuzzy Traffic Signal Control*; Helsinki University of Technology: Espoo, Finland, 1998.
17. Liu, Z. A Survey of Intelligence Methods in Urban Traffic Signal Control. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 105–112.
18. Spall, J.; Chin, D. A model-free approach to optimal signal light timing for system-wide traffic control. In Proceedings of the 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994; doi:10.1109/CDC.1994.411110.
19. Srinivasan, D.; Choy, M.C.; Cheu, R.L. Neural Networks for Real-Time Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 261–272, doi:10.1109/TITS.2006.874716. [[CrossRef](#)]
20. Shoham, Y.; Powers, R.; Grenager, T. *Multi-Agent Reinforcement Learning: A Critical Survey*; Technical Report; Computer Science Department, Stanford University: Stanford, CA, USA, 2003.
21. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: Cambridge, MA, USA; London, UK, 2012.
22. Abdulhai, B.; Pringle, R.; Karakoulas, G.J. Reinforcement learning for true adaptive traffic signal control. *J. Transp. Eng.* **2003**, *129*, 278–285, doi:10.1061/(ASCE)0733-947X(2003)129:3(278). [[CrossRef](#)]
23. Salkham, A.; Cunningham, R.; Garg, A.; Cahill, V. A collaborative reinforcement learning approach to urban traffic control optimization. In Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, 9–12 December 2008; doi:10.1109/WIAT.2008.88.
24. Park, H.; van, M. Bargaining strategies for networked multimedia resource management. *IEEE Trans. Signal Process.* **2007**, *55*, 3496–3511, doi:10.1109/TSP.2007.893755. [[CrossRef](#)]
25. Han, Z.; Liu, K.J.R. Fair Multiuser Channel Allocation for OFDMA Networks Using Nash Bargaining Solutions and Coalitions. *IEEE Trans. Commun.* **2005**, *53*, 1366–1376, doi:10.1109/TCOMM.2005.852826. [[CrossRef](#)]
26. Chen, J. Game-Theoretic Formulations Of Interaction Between Dynamic Traffic Control and Dynamic Traffic Assignment. *Transp. Res. Rec.* **1998**, *1617*, 179–188, doi:10.3141/1617-25. [[CrossRef](#)]
27. Elhenawy, M.; Elbery, A.A.; Hassan, A.A.; Rakha, H.A. An Intersection Game-Theory-Based Traffic Control Algorithm in a Connected Vehicle Environment. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 343–347, doi:10.1109/ITSC.2015.65. [[CrossRef](#)]
28. Jun, L. *Study on Game-Theory-Based Integration Model for Traffic Control and Route Guidance*; Tian Jin University: Tianjin, China, 2003.
29. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. Isolated Traffic Signal Control using Nash Bargaining Optimization. *Glob. J. Res. Eng. B Automot. Eng.* **2016**, *16*, 27–36.
30. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. Isolated traffic signal control using a game theoretic framework. In Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1496–1501, doi:10.1109/ITSC.2016.7795755. [[CrossRef](#)]
31. Han, Z.; Niyato, D.; Saad, W.; Basar, T.; Hjørungnes, A. *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*; Cambridge University Press: New York, NY, USA, 2012.
32. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. A Novel Game Theoretic De-Centralized Traffic Signal Controller: Model Development and Testing. In Proceedings of the 97th Annual Meeting of Transportation Research Board, Washington, DC, USA, 7–11 January 2018.
33. Aerde, M.V.; Rakha, H.A. *INTEGRATION© Release 2.40 for Windows: User's Guide—Volume I: Fundamental Model Features*; Technical Report; Center for Sustainable Mobility, Virginia Tech Transportation Institute: Blacksburg, VA, USA, 2012.
34. Aerde, M.V.; Rakha, H.A. *INTEGRATION© Release 2.40 for Windows: User's Guide—Volume II: Advanced Model Features*; Technical Report, Center for Sustainable Mobility, Virginia Tech Transportation Institute: Blacksburg, VA, 24060, USA, 2013.

35. Rakha, H.A.; Zhang, Y. The INTEGRATION 2.30 Framework for Modeling Lane-Changing Behavior in Weaving Sections. *Transp. Res. Rec. J. Transp. Res. Board* **2004**, *1883*, 140–149, doi:10.3141/1883-16. [[CrossRef](#)]
36. Rakha, H.A.; Pasumarthy, P.; Adjerid, S. A Simplified Behavioral Vehicle Longitudinal Motion Model. *Transp. Lett. Int. J. Transp. Res.* **2009**, *1*, 95–110, doi:10.3328/TL.2009.01.02.95-110. [[CrossRef](#)]
37. Rakha, H.A. Validation of Van Aerde's Simplified Steady-state Car-following and Traffic Stream Model. *Transp. Lett. Int. J. Transp. Res.* **2009**, *1*, 227–244, doi:10.3328/TL.2009.01.03.227-244. [[CrossRef](#)]
38. Wu, N.; Rakha, H.A. Derivation of Van Aerde Traffic Stream Model from Tandem-Queueing Theory. *Transp. Res. Rec. J. Transp. Res. Board* **2009**, *2124*, 18–27, doi:10.3141/2124-02. [[CrossRef](#)]
39. Rakha, H.A.; Lucic, I.; Demarchi, S.; Setti, J. Vehicle Dynamics Model for Predicting Maximum Truck Accelerations. *J. Transp. Eng.* **2001**, *127*, 418–425, doi:10.1061/(ASCE)0733-947X(2001)127:5(418). [[CrossRef](#)]
40. Rakha, H.A.; Snare, M.; Dion, F. Vehicle Dynamics Model for Estimating Maximum Light Duty Vehicle Acceleration Levels. *Transp. Res. Rec. J. Transp. Res. Board* **2004**, *1883*, 40–49, doi:10.3141/1883-05. [[CrossRef](#)]
41. Dion, F.; Rakha, H.A.; Kang, Y.S. Comparison of Delay Estimates at Under-saturated and Over-saturated Pre-timed Signalized Intersections. *Transp. Res. Part B Methodol.* **2004**, *38*, 99–122, doi:10.1016/S0191-2615(03)00003-1. [[CrossRef](#)]
42. Rakha, H.A.; Kang, Y.S.; Dion, F. Estimating Vehicle Stops at Under-Saturated and Over-Saturated Fixed-Time Signalized Intersections. *Transp. Res. Rec.* **2001**, *1776*, 128–137, doi:10.1016/S0191-2615(03)00003-1. [[CrossRef](#)]
43. Ahn, K.; Rakha, H.A.; Trani, A.; Aerde, M.V. Estimating Vehicle Fuel Consumption and Emissions Based on Instantaneous Speed and Acceleration Levels. *J. Transp. Eng.* **2002**, *128*, 182–190. doi:10.1061/(ASCE)0733-947X(2002)128:2(182). [[CrossRef](#)]
44. Rakha, H.A.; Ahn, K.; Trani, A. Development of VT-Micro Framework for Estimating Hot Stabilized Light Duty Vehicle and Truck Emissions. *Transp. Res. Part D Transp. Environ.* **2004**, *9*, 49–74, doi:10.1016/S1361-9209(03)00054-3. [[CrossRef](#)]
45. Rakha, H.A.; Ahn, K. INTEGRATION Modeling Framework for Estimating Mobile Source Emissions. *J. Transp. Eng.* **2004**, *130*, 183–193, doi:10.1061/(ASCE)0733-947X(2004)130:2(183). [[CrossRef](#)]
46. Aerde, M.V.; Rakha, H.A. REALTRAN: An Off-line Emulator for Estimating the Impacts of SCOOT. In Proceedings of the 74th Transportation Research Board Annual Meeting, Washington, DC, USA, 22–28 January 1995; pp. 124–128
47. Rakha, H.A.; Aerde, M.V.; E.R.. Experiments in Incremental Real-Time Optimization of Phase, Cycle, and Offset Times Using an On-Line Adaptation of TRANSYT-7F. In Proceedings of the Engineering Foundation Conference on Traffic Management: Issues and Techniques, Palm Coast, FL, USA, 1–6 April 1991.
48. Abdelghaffar, H.M.; Yang, H.; Rakha, H.A. Developing a De-centralized Cycle-free Nash Bargaining Arterial Traffic Signal Controller. In Proceedings of the 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, IEEE MT-ITS, Naples, Italy, 26–28 June 2017; doi:10.1109/MTITS.2017.8005732.
49. Aerde, M.V.; Rakha, H.A. *QUEENSOD Rel. 2.10—User's Guide: Estimating Origin—Destination Traffic Demands from Link Flow Counts*; Technical Report, Center for Sustainable Mobility, Virginia Tech Transportation Institute: Blacksburg, VA, 24060, USA, 2010.
50. Rakha, H.A.; Paramahamsan, H.; Aerde, M.V. Comparison of Static Maximum Likelihood Origin-Destination Formulations. In *Transportation and Traffic Theory: Flow, Dynamics and Human Interaction*, Elsevier: College Park Maryland, United States; July, 2005; pp. 693–716, doi:10.1016/B978-008044680-6/50037-4. [[CrossRef](#)]
51. Aerde, M.V.; Rakha, H.A.; Paramahamsan, H. Estimation of O-D Matrices: The Relationship between Practical and Theoretical Considerations. *Transp. Res. Rec.* **2003**, *1831*, 122–130, doi:10.1016/B978-008044680-6/50037-4. [[CrossRef](#)]

52. Yang, H.; Rakha, H.A. A Novel Approach for Estimation of Dynamic from Static Origin-Destination Matrices. *Transp. Lett. Int. J. Transp. Res.* **2019**, *11*, 219–228, doi:10.1080/19427867.2017.1336353. [[CrossRef](#)]
53. Du, J.; Rakha, H.A.; Elbery, A.; Klenk, M. Microscopic Simulation and Calibration of a Large-Scale Metropolitan Network: Issues and Proposed Solutions. In Proceedings of the 97th Annual Meeting of Transportation Research Board, Washington, DC, USA, 7–11 January 2018.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).



Article

Efficient Traffic Video Dehazing Using Adaptive Dark Channel Prior and Spatial–Temporal Correlations

Tianyang Dong, Guoqing Zhao, Jiamin Wu, Yang Ye and Ying Shen *

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China; dty@zjut.edu.cn (T.D.); m15695719625@163.com (G.Z.); hzwujiamin@163.com (J.W.); yeyang80@zjut.edu.cn (Y.Y.)

* Correspondence: shenyings@zjut.edu.cn

Received: 6 March 2019; Accepted: 26 March 2019; Published: 2 April 2019

Abstract: In order to restore traffic videos with different degrees of haziness in a real-time and adaptive manner, this paper presents an efficient traffic video dehazing method using adaptive dark channel prior and spatial-temporal correlations. This method uses a haziness flag to measure the degree of haziness in images based on dark channel prior. Then, it gets the adaptive initial transmission value by establishing the relationship between the image contrast and haziness flag. In addition, this method takes advantage of the spatial and temporal correlations among traffic videos to speed up the dehazing process and optimize the block structure of restored videos. Extensive experimental results show that the proposed method has superior haze removing and color balancing capabilities for the images with different degrees of haze, and it can restore the degraded videos in real time. Our method can restore the video with a resolution of 720×592 at about 57 frames per second, nearly four times faster than dark-channel-prior-based method and one time faster than image-contrast-enhanced method.

Keywords: image dehazing; traffic video dehazing; dark channel prior; spatial-temporal correlation; contrast enhancement

1. Introduction

Today, traffic video analysis plays a very important role in intelligent transportation systems. It has become a common way to help people track a vehicle, as well as locate and judge an accident. Because the images captured by outdoor cameras are often affected by different weather conditions, they suffer from poor visibility and lack of contrast. In the literature, there are many enhancements and dehazing algorithms that improve different images, such as traffic videos, underwater images, and satellite imagery [1–3]. The hazy weather that happens frequently all over the world is becoming a video analysis killer. The haze captured in the video degrades the contrast and color information and reduces the visibility. Therefore, the problem of how to efficiently and effectively remove the haze in traffic videos has attracted broad attention from both academia and industry. In general, when dealing with haze removal in traffic videos, the existing dehazing algorithms often exhibit poor real-time performance, overstretched contrast, and even fail to remove dense haze. The key issue of these problems is how to deal with images in different scenes with different degrees of haze, thus an adaptive algorithm that can remove haze based on the image characteristics is needed. Moreover, the existing video-dehazing methods are almost universal for all videos and do not consider the characteristics of videos in particular scenarios. For traffic videos, the time continuity, lane space structure, and camera spatial locations can be effectively used to decrease computational cost.

In order to restore traffic videos with different degrees of haziness in a real-time and adaptive manner, this paper presents an efficient traffic video dehazing method using adaptive dark channel prior and spatial-temporal correlations. This method can avoid overstretched contrast after haze removal and obtain satisfactory restored results for dense hazy videos by using a novel approach

involving adaptive transmission estimation. This method also takes full advantage of the temporal and spatial correlations in traffic videos to meet the requirements of real-time dehazing, such as using time continuity to set the time slice, refining transmission by characteristics of block structure, decreasing restored area according to the lane space, and simplifying the calculation of parameters by using multi-camera distribution.

2. Related Works

Essentially, videos are composed of frames, thus the haze removal method for images can be used for videos. The image dehazing method is the most common way to restore hazy images. This method considers the inverse process of image degradation and describes the image degradation process in detail through an established physical model. The most critical step of this method is to obtain the parameters of the degradation model. Oakley et al. [4] improved the image quality by using the physical model and estimated the degradation model parameters based on a statistical model. This method is not widely used because it is only useful for gray-scale images, and the acquiring parameters require calibrated radar to get depth information. Narasimhan et al. [5] proposed a method to estimate the depth information by comparing two images of the same scene in different weather conditions. Chen et al. [6] used a sunny image and a foggy image for reference images to calculate parameters. Both of these methods need to receive eligible images in advance, which increases the difficulty of image acquisition.

To obtain the parameters of the degradation model effectively, some dehazing methods based on prior knowledge or assumptions were proposed, and they do not need to get reference images in advance or use an additional hardware device. Therefore, these methods have better adaptability than previous methods. Based on the assumption that a haze-free image has a higher contrast than a hazy image, Tan [7] proposed a haze removal approach by maximizing the contrast of recovered scene radiance. This approach can produce a satisfactory result for haze removal in single images, but it tends to overcompensate for the reduced contrast and leads to halo effects. Fattal [8] decomposed scene radiance of an image into the albedo and shading and then estimated the scene radiance based on independent component analysis, assuming that transmission shading and surface shading are locally uncorrelated. However, this method cannot generate impressive results when the captured image is heavily obscured by fog. He et al. [9] presented a single image haze removal method by using dark channel prior, which can estimate the transmission map directly. However, when a large white area without shading exists in the images, or the images have uneven illumination, this method takes a long time to restore the hazy images. In addition, the use of the soft matting algorithm makes this a complex computation. Then, Lai et al. [10] presented a haze removal method based on the difference-structure-preservation prior. In this method, the difference-structure-preservation dictionary is learned such that the local consistency features of the transmission map can be well preserved after coefficient shrinkage. Zhu et al. [11] presented a simple but effective Color Attenuation Prior (CAP) algorithm similar to Dark Channel Prior (DCP) using the difference in brightness and saturation to estimate the haze concentration to build a depth model for dehazing. Up until now, other researchers have improved their dehazing algorithms based on the dark channel prior. Yeh et al. [12] introduced a haze removal algorithm based on region decomposition and feature fusion, which is especially suitable for hazy images with large sky regions. Li et al. [13] proposed a novel haze removal method based on sky segmentation and dark channel prior to restore images. In this method, the average image intensity of the sky region is chosen as the atmospheric light value. Wang et al. [14] designed a new method of selecting atmospheric light values to weaken the area where the dark channel priority does not work effectively. A visibility restoration method was introduced by Huang et al. [15], which consists of three modules: (i) depth estimation module based on dark channel priority, (ii) color analysis module that repairs depth estimation distortion, and (iii) visibility restoration module that generates repair results. Riaz et al. [16] proposed a new and efficient method for transmission estimation with bright-object handling capability, which uses a local average haziness value to compute the

transmission of such surfaces based on the observation that the transmission of a surface is loosely connected to its neighbors.

Usually, traffic video dehazing algorithms are proposed based on single-image dehazing algorithms. However, the computational complexity makes it difficult to apply single-image dehazing algorithms directly to video dehazing. Most existing research on video dehazing is to speed up the process of dehazing. Sun et al. [17] proposed a real-time haze removal method based on bilateral filtering to reduce the processing time of 320×240 images to a speed of 20 frames per second. However, this method cannot satisfy the requirements of high-definition videos. Wang et al. [18] proposed a method based on Retinex theory that enhances image contrast in YUV color space and can process an image of 704×576 in 0.055 s. Kumari et al. [19] proposed an approach for dehazing images and videos based on a filtering method. The use of a gray-scale morphological operation made the approach faster, and it took only 80% of the execution time compared to a fast bilateral filter. Berman et al. [20,21] proposed a new method via calculating the air-light to dehaze fogs, which was based on a non-local prior. Their algorithm relies on the assumption that colors of a haze-free image are well approximated by a few hundred distinct colors that form tight clusters in RGB space. It performs well on a wide variety of images. However, these methods take every frame in videos as a single image, and they are completely based on image dehazing methods.

The characteristics of videos can be applied in specific video dehazing algorithms. Tarel et al. [22] proposed a video dehazing method for onboard video systems. This method can separate moving objects and driveway regions in videos and only update the depth information of moving objects. Zhang et al. [23] proposed a method based on spatial and temporal correlation that uses spatial and temporal similarity between frames to optimize the estimation of a scene depth map. Shin et al. [24] proposed an effective video dehazing technique to reduce flicker artifacts by using adaptive temporal average. However, these methods cannot remove the haze from videos in real time. Therefore, Kim et al. [25] proposed an image-dehazing method based on the image degradation model and kept a balance between image contrast enhancement and image information loss. To improve the speed of video dehazing, they adopted a video dehazing method by using temporal correlation, which can reach a speed of 30 frames per second for videos with a resolution of 640×480 . However, this method adopts a fixed initial transmission value that cannot be adapted to images with different degrees of haze, and it cannot efficiently remove dense haze in videos. Our method uses an adaptive initial transmission value based on image characteristics to handle different degrees of hazes; meanwhile, it can reduce the processing time through lane space separation.

3. Single-Image Dehazing Using Adaptive Dark Channel Prior

3.1. Framework of Single-Image Dehazing Method

The most common dehazing model is based on atmospheric optics [26], which can describe the degradation process of a hazy image. In [27], the modeling function is simplified, and it is represented by Equation (1).

$$I(p) = J(p)t(p) + A(1 - t(p)) \quad (1)$$

where p is a pixel in the image, $I(p)$ and $J(p)$ are the observed and haze-free image, respectively, A is the global atmospheric light, and $t(p) \in [0, 1]$ is the transmission map for each pixel that describes the proportion of the light arriving at a digital camera without scattering.

The process of haze removal for every frame of a traffic video can be divided into three steps: calculating atmospheric light, estimating the transmission map, and restoring the image. In this paper, we present a novel adaptive method for transmission map estimation, thus the dehazing algorithm can be applied to images with different degrees of haze. The framework of the single-image dehazing algorithm is shown in Figure 1.

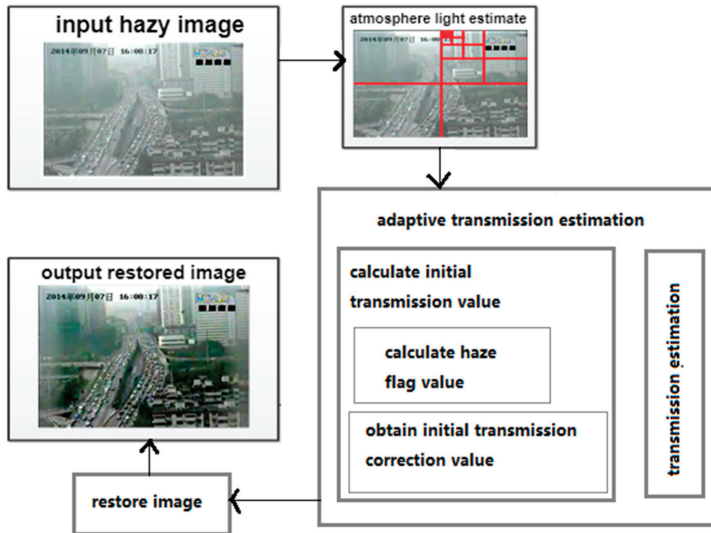


Figure 1. Framework of single-image dehazing method.

We use a hierarchical searching method based on quad-tree subdivision [25] to find the areas least affected by haze and to get the brightest pixel in this area. The detailed steps are as follows:

- Step 1: Divide an input image into four rectangular regions.
- Step 2: Define the score of each region as the average pixel value subtracted from the standard deviation of the pixel values within the region.
- Step 3: Select the region with the highest score and divide it further into four smaller regions.
- Step 4: Repeat Steps 1 through Step 3 until the size of the selected region is smaller than a prespecified threshold. The prespecified threshold in this paper is 200, which is that the height * width of the selected region is smaller than 200.

At last, we choose the color vector, which minimizes the distance $\|(I_r(p), I_g(p), I_b(p)) - (255, 255, 255)\|$, where $I(p)$ is the value of pixel p in the selected region as the atmospheric light.

3.2. Transmission Estimation for Enhancing the Contrast of Blocks

In general, a hazy block yields low contrast, and the contrast of a restored block increases as the value of the estimated transmission decreases. We adopt the image-contrast-enhanced method [18] to maximize the contrast of the restored blocks and get the best estimated transmission value.

Mean squared error contrast (CMSE) [28] can define the contrast of a restored block, which is represented by Equation (2):

$$C_{MSE} = \sum_{p=1}^N \frac{(J_c(p) - \bar{J}_c)^2}{N} \quad (2)$$

where $J_c(p)$ represents the RGB color channel of pixel p in a block of input image, $c \in \{r, g, b\}$, \bar{J}_c is the average value of $J_c(p)$, and N is the number of pixels in a block.

According to the assumption that the scene depths are locally similar [8,12,16], the dehazing algorithm in this paper determines a single transmission value for each block of size 32×32 , and then gets the fixed optimal transmission value t for each block. For a pixel p in a block, $t(p)$ in Equation (1)

can be replaced with the fixed estimated transmission t of its block. Hence, $J_c(p)$ is represented by Equation (3).

$$J_c(p) = \frac{I_c(p) - A}{t} + A \quad (3)$$

If Equation (3) is put into Equation (2), C_{MSE} can be represented by Equation (4):

$$C_{MSE} = \sum_{p=1}^N \frac{(I_c(p) - \bar{I}_c)^2}{t^2 N} \quad (4)$$

where \bar{I}_c is the average value of $I_c(p)$ in the input block. According to Equation (4), we can find that the mean squared error contrast is a decreasing function of t . Thus, we can select a small value of t to increase the contrast of a restored block. However, the value of t influences the pixel's restored image value according to Equation (3).

However, when a block contains dense haze, it has a relatively narrow value range for input pixels. Thus, even though it is assigned a small t value, most of the input values are not truncated, and the block can be correctly restored. On the contrary, a block without haze usually has a broad range of values for input pixels and should be assigned a larger t value to reduce the information loss due to the truncation. Thus, we should not only enhance the contrast but also reduce the information loss.

Therefore, we need to set quantitative evaluations for contrast and information integrity. The contrast cost $E_{contrast}$ and the information loss cost E_{loss} were proposed by Kim [25] to evaluate the contrast and information integrity, respectively.

$$E_{contrast} = - \sum_{c \in \{r,g,b\}} \sum_{p \in B} \frac{(J_c(p) - \bar{J}_c)}{N_B} = - \sum_{c \in \{r,g,b\}} \sum_{p \in B} \frac{(I_c(p) - \bar{I}_c)}{N_B} \quad (5)$$

where \bar{J}_c and \bar{I}_c are the average values of $J_c(p)$ and $I_c(p)$ in block B , respectively, and N_B is the number of pixels in B . Thus, we can maximize the mean squared error contrast by minimizing the value of $E_{contrast}$.

$$E_{loss} = \sum_{c \in \{r,g,b\}} \sum_{p \in B} \{(\min\{0, J_c(p)\})^2 + (\max\{0, J_c(p) - 255\})^2\} \quad (6)$$

where $\min\{0, J_c(p)\}$ and $\max\{0, J_c(p) - 255\}$ denote the truncated values for output pixels due to the underflow and overflow, respectively.

If we want to get a better restored image, the image contrast should be smoother, and the color information should be maintained as much as possible. Thus, these two factors should be taken into consideration synthetically, and the overall cost function is described as Equation (7).

$$E = E_{contrast} + \lambda_L E_{loss} \quad (7)$$

where λ_L is a weight coefficient that controls the relative importance of the contrast cost and the information loss cost [18]. The minimum value of E represents the most suitable contrast for restored images, and the color loss is as small as possible. Finally, for each block in a hazy image, we can get an optimal transmission t^* by minimizing the value of E . The value of t^* is the transmission we use while dehazing.

3.3. Adaptive Estimation of Initial Transmission

3.3.1. Calculating Image Haziness Flag

We present a haziness flag T to measure the degree of haze in an image. The dark channel prior [9] can estimate the transmission of a block, which represents the luminosity of objects. The transmission has a close relationship with the degree of haze. Therefore, we can adopt the average value of

transmission as the haziness flag T of an image. The haziness flag T is concerned with the effects of the degree of haze in images.

The dark channel prior is based on the observation that most local blocks in haze-free outdoor images contain some pixels that have very low intensities in at least one color channel. In other words, the dark channel value of a haze-free image is close to zero [9]. For any input image J , dark channel J_{dark} can be expressed as Equation (8).

$$J_{dark}(p) = \min_{y \in \Omega(p)} \left(\min_{c \in \{r,g,b\}} J_c(y) \right) \quad (8)$$

where $c \in \{r, g, b\}$ and $\Omega(p)$ represent a local block centered at p , and y is a pixel in the local block $\Omega(p)$. A dark channel is the outcome of two minimum operators: $\min_c J_c(y)$ is performed on each pixel, and $\min_{y \in \Omega(p)}$ is a minimum filter [9].

Assuming that the atmospheric light A_c is given, we can normalize the haze imaging Equation (1) by A_c [9]:

$$\frac{I_c(p)}{A_c} = t(p) \frac{J_c(p)}{A_c} + 1 - t(p) \quad (9)$$

Since the transmission $t(p)$ is a constant $\tilde{t}(p)$ in local block, and the value of A_c is given, the dark channel operation can be given by the following equations [9].

$$\min_{y \in \Omega(p)} \left(\min_c \frac{I_c(y)}{A_c} \right) = \tilde{t}(p) \min_{y \in \Omega(p)} \left(\min_c \frac{J_c(y)}{A_c} \right) + 1 - \tilde{t}(p) \quad (10)$$

Using the concept of a dark channel [9], if J_c is an outdoor haze-free image except for the sky region, the intensity of dark channel is low and tends to be zero, which leads to:

$$\min_{y \in \Omega(p)} \left(\min_c \frac{J_c(y)}{A_c} \right) = 0 \quad (11)$$

Putting Equation (11) into Equation (9), we can eliminate the multiplicative term and estimate the transmission $\tilde{t}(p)$ simply by

$$\tilde{t}(p) = 1 - \min_{y \in \Omega(p)} \left(\min_c \frac{I_c(y)}{A_c} \right) \quad (12)$$

where $\tilde{t}(p)$ is the predicted value of transmission of a block [9]. We need to calculate the average transmission for all blocks to obtain the average transmission T for the whole image, which is the value of image haziness flag.

3.3.2. Correction of Initial Transmission

According to our experimental results, in a hazy image, the range of T is generally between 0.4 and 0.6. Although the image haziness flag T can characterize the nature of the image, taking T as the initial transmission value to get the optimal transmission leads to an excessive value of t^* . Thus, we set a correction value X , and set $T * X$ as the initial transmission value to decrease this initial value.

The structural similarity (SSIM) index is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. To guarantee that the restored images are closer to ground truths, we adopted the SSIM index [29] to measure the similarity between the ground truths and restored images. Because the traffic video is captured by a fixed camera, we can get a haze-free image of the same scene as a reference image in advance and compare the restored image with the reference image. The initial value of T can be obtained directly because it is relevant to the nature of images, whereas the unknown value X is calculated by the SSIM. In our experiments, we set X as a series of values between 0.3 and 1.2, and the interval is 0.02. Then, we take every X in this range multiplied by T , that is, $T * X$, as the initial value of transmission

and get the corresponding restored image. At last, we find a restored image that is closest to the haze-free image based on the maximum value of the SSIM index. Thus, the value of transmission is the optimal initial value, and the corresponding correction value X is the optimal correction value of initial transmission.

However, this method needs a haze-free image to get the optimal correction value X . This method is limited in practical applications, thus it is necessary to get the correction value according to the image characteristics. After analyzing the image contrast and the haze in images, we find the relationship between the correction value of initial transmission and the image characteristics. Therefore, a relatively reasonable initial transmission correction value can be obtained directly from hazy images.

If the relatively reasonable correction value of initial transmission is X' , we take $T * X'$ as the initial transmission value. Because the dehazing algorithm is based on the concept of enhancing the image contrast to the greatest degree, the contrast is the important indicator. The value of image haziness flag T represents the degree of haze that degrades the image contrast. Thus, the image contrast and haziness flag value should be considered simultaneously. We set C as the image contrast and set $T * C$ as a quantitative value representing the image characteristics. The constant value X' depends on the range of value $T * C$.

Table 1 shows the values of X' for different ranges of $T * C$. In Table 1, X is the optimal correction value obtained by the method with reference images, and X' is the relatively reasonable correction value obtained by the ranges of $T * C$. In the dehazing algorithm, the initial transmission value is the key factor that affects the dehazing result. Table 1 shows the values of $T * X$ and $T * X'$, which are the initial transmission value derived by optimal correction of initial X and relatively reasonable correction value X' , respectively. Figure 2 shows the histogram of $T * X$ and $T * X'$, where the values of $T * X$ and $T * X'$ in the same group have similar values, and the difference of the values in the same group does not affect the dehazing results significantly. Therefore, our method can determine the optimal initial transmission value using only the nature of images and then obtain a more adaptive transmission value.

Table 1. The value of x' for different ranges of $T * C$.

| Image No. | T | C | $T * C$ | X | The Range of $T * C$ | X' | $T * X'$ | $T * X$ |
|-----------|--------|---------|---------|------|----------------------|------|----------|---------|
| 1 | 0.4032 | 3.8224 | 1.5414 | 0.50 | $T * C < 10$ | 0.5 | 0.2016 | 0.2016 |
| 2 | 0.4006 | 6.3436 | 2.5410 | 0.52 | | 0.5 | 0.2003 | 0.2083 |
| 3 | 0.4177 | 8.4845 | 3.5437 | 0.50 | | 0.5 | 0.2088 | 0.2088 |
| 4 | 0.4113 | 13.4080 | 5.5151 | 0.50 | | 0.5 | 0.2056 | 0.2057 |
| 5 | 0.4329 | 13.2774 | 5.7476 | 0.46 | | 0.5 | 0.2164 | 0.1991 |
| 6 | 0.4444 | 17.6432 | 7.84004 | 0.46 | | 0.5 | 0.2222 | 0.2044 |
| 7 | 0.4211 | 19.7160 | 8.3039 | 0.52 | | 0.5 | 0.2160 | 0.2190 |
| 8 | 0.4584 | 22.1363 | 10.1480 | 0.54 | $10 \leq T * C < 15$ | 0.6 | 0.2750 | 0.2476 |
| 9 | 0.4275 | 25.5289 | 10.9141 | 0.64 | | 0.6 | 0.2565 | 0.2736 |
| 10 | 0.4732 | 26.9131 | 12.7346 | 0.62 | | 0.6 | 0.2839 | 0.2934 |
| 11 | 0.4370 | 31.9037 | 13.9419 | 0.76 | | 0.6 | 0.2622 | 0.3321 |
| 12 | 0.4862 | 31.3389 | 15.2359 | 0.66 | $15 \leq T * C < 20$ | 0.7 | 0.3403 | 0.3209 |
| 13 | 0.4469 | 38.3871 | 17.1555 | 0.84 | | 0.7 | 0.3128 | 0.3754 |
| 14 | 0.4987 | 35.6754 | 17.7904 | 0.62 | | 0.7 | 0.3491 | 0.3092 |
| 15 | 0.4555 | 44.9152 | 20.4609 | 0.80 | $20 \leq T * C < 25$ | 0.8 | 0.3644 | 0.3644 |
| 16 | 0.4625 | 50.9075 | 23.5422 | 0.86 | | 0.8 | 0.3700 | 0.3977 |
| 17 | 0.4724 | 57.3643 | 27.1012 | 0.94 | $25 \leq T * C < 30$ | 0.9 | 0.4252 | 0.4441 |
| 18 | 0.4812 | 63.6731 | 30.6395 | 1.00 | $T * C \geq 30$ | 1.0 | 0.4812 | 0.4812 |
| 19 | 0.4909 | 70.3751 | 34.5454 | 1.06 | | 1.0 | 0.4909 | 0.5203 |

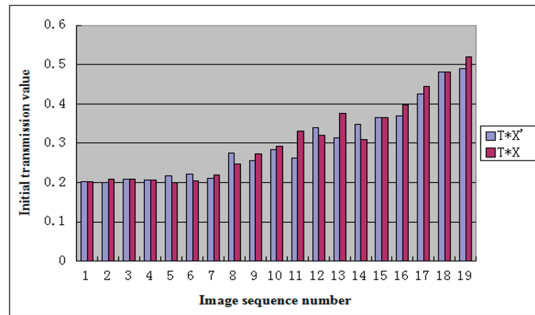


Figure 2. The histogram of $T * X'$ and $T * X$.

4. Adaptive Traffic Video Dehazing Method Using Spatial–Temporal Correlations

Compared with static traffic images, traffic videos have some unique characteristics. First, a traffic video is a collection of images with time continuity. Second, the cameras are fixed on the road and capture videos of the same scene over a long time, thus the videos are consistent in space. Therefore, we can use the correlations of spatial-temporal information to speed up traffic video dehazing.

4.1. Time Continuity of Traffic Videos

Because the cameras are fixed, the scenes in traffic videos barely change over a long period of time, and the influence of haze is stable. In our experiments, we use the traffic videos from ZhongHe elevated freeways in Hangzhou City, set a cycle of five minutes, and regard the frames in one cycle as a collection of images with the same characteristics. Figure 3 shows images whose interval is 1 min in a 5 min cycle, and the difference of T is very small, usually less than 0.04. Figure 4 presents the difference in restored images by using different T values where the results have no obvious influence on visibility with the difference of T less than 0.04. Therefore, if the videos are captured at the same scene, the values of T for these video images in a 5 min cycle are at the same level, and the cycle of 5 min is reasonable in practical application.

After setting the 5 min cycle, we can take the first frame of a video segment as a reference frame. We can determine the image haziness flag value T and the relatively reasonable initial transmission correction value X' from the reference frame and then calculate the optimal transmission I^* . In this way, we can speed up the dehazing processing for the traffic video. This method can avoid incorrect transmission estimation, which is caused by the changes in atmospheric light, and eliminate the discontinuity of videos after dehazing.



Figure 3. The difference of T for the images in a 5 min cycle. The images come from different scenes (a,b).

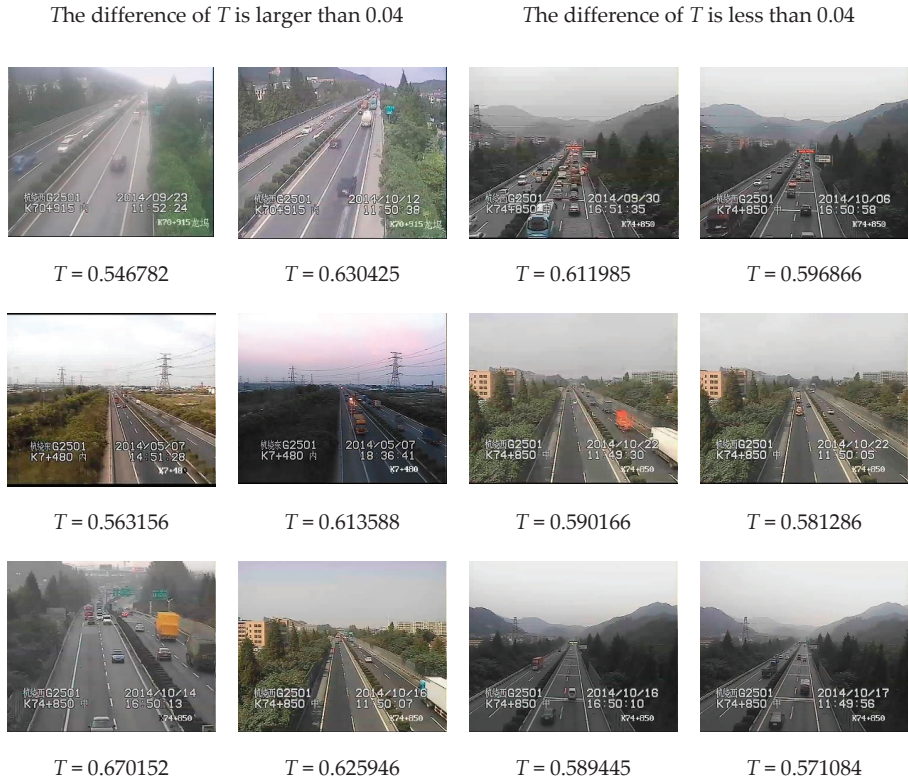


Figure 4. The images with different T values.

4.2. Transmission Refinement Based on Spatial Structure

We estimate the optimal transmission based on the assumption that all pixels in a block have the same transmission. However, scene depths may vary spatially within a block, and the block-based transmission map usually has a blocking-artifact problem. Therefore, an edge-preserving filter is adopted to refine the block-based transmission map.

The single-image dehazing method using dark channel prior [9] employs the soft matting technique [30] to refine the large block size in the transmission map, which causes an enormous computational burden. In this paper, the guided filter method [31] is adopted to refine the transmission map, which has less computational cost. The filtered transmission $\hat{t}(p)$ is an affine combination of the guidance image $I(p)$, as show in Equation (13):

$$\hat{t}(p) = s^T I(p) + \psi \quad (13)$$

where $s = (s_r, s_g, s_b)^T$ is a scaling vector, and ψ is an offset determined by the size of block. For a block in one image, the optimal parameters of s^* and ψ^* can be obtained by minimizing the difference between the transmission $t(p)$ and the filtered transmission $\hat{t}(p)$ using the least squares method as Equation (14):

$$(s^*, \psi^*) = \operatorname{argmin}(s, \psi) \sum_{p \in \Omega} (t(p) - \hat{t}(p))^2 \quad (14)$$

If the transmission is too small, the noise will be enhanced in the restored image [9]. Thus, the lower limit of the transmission is set to 0.1. If a window slides pixel by pixel over the entire image,





there will be multiple windows that overlap at each pixel position. Therefore, we adopt the centered window scheme, which sets the final transmission values as the average of all associated refined transmission values at each pixel position. However, the average transmission value in this scheme will cause blurring in the final transmission map, especially around object boundaries, where the depths change abruptly. To overcome this problem, the shiftable window scheme [32] is employed instead of the centered window scheme. The centered window scheme overlays a window on each pixel so that the window contains multiple objects with different depths, which leads to unreliable depth estimation. In the shiftable window scheme, the window is shifted within a block of 40×40 . The optimal shift position is selected depending on the smallest change of pixel values within the window. Even though a shiftable window is selected for a specific pixel, the number of overlapping windows usually varies at different positions. The windows in smooth regions are selected more frequently than those in rough boundary regions. Thus, the shiftable window scheme can reduce the effects of unreliable transmission values derived from rough boundary regions, thereby alleviating the blurring artifacts.

4.3. Lane Separation for Traffic Videos

After analyzing the spatial characteristics of traffic video, we found that the traffic lane is an obvious structure. In a traffic video detection system, the detected objects are mostly concentrated in the driveway regions. The areas outside lanes are not the regions of interest in traffic video processing. Therefore, we can process haze removal only in the driveway region of traffic video to reduce computing time.

However, the estimations of atmospheric light and transmission are based on the whole image. If these values are achieved only through the driveway regions, it may cause some deviations, especially when the sky occupies a large area of the image, such as the cases shown in Table 2. The larger the sky region is, the greater the deviation for the value of $T * X$ is. Therefore, the separated lane can be used in the last step to restore the pixels only for the driveway regions.

Table 2. Global image and driveway.

| | | Case 1 | Case 2 | Case 3 | Case 4 | | |
|--------------|------------|---|---|---|--|----------|---------|
| Regions | Parameters |  |  |  |  | | |
| | | Driveway Region | T | 0.590856 | 0.704105 | 0.839763 | 0.83898 |
| | | contrast | 47.7547 | 49.0273 | 54.0312 | 62.208 | |
| | | X' | 0.90000 | 1.0000 | 1.0000 | 1.0000 | |
| | $T * X'$ | 0.53200 | 0.70400 | 0.8400 | 0.8390 | | |
| Global Image | T | 0.563265 | 0.632323 | 0.773405 | 0.563549 | | |
| | contrast | 48.8811 | 49.2619 | 57.5056 | 127.7800 | | |
| | X' | 0.9000 | 1.0000 | 1.0000 | 1.0000 | | |
| | $T * X'$ | 0.5070 | 0.6320 | 0.7730 | 0.5660 | | |

We adopt a straight-line extraction algorithm based on the Hough transform to detect the lanes and separate the driveway region from the global image. The process of haze removal combined with the driveway region separation is described as follows:

1. Calculate the global atmospheric light A , the value of haziness flag T , and the image contrast C , then estimate the optimal transmission map for each block in an image.
2. Get the driveway region, as shown in Figure 5.

- Step 1: Obtain the edge information in the video through edge detection.
- Step 2: Remove obviously wrong-angle lines by Hough linear fitting, and obtain lane candidates, as shown in Figure 5b.
- Step 3: Find the far left lane and the far right lane, and set them as the driveway boundaries, then find the intersection of these two lines, as shown in Figure 5c.
- Step 4: Identify a rectangular area as the driveway region, which is composed of the boundary of the image and a horizontal line across the intersection, as shown in Figure 5c. If the intersection is outside the image, take the whole image area as the driveway region.
- Use the original pixel values and the optimal transmission of driveway region in the dehazing model to restore the image in the driveway region.

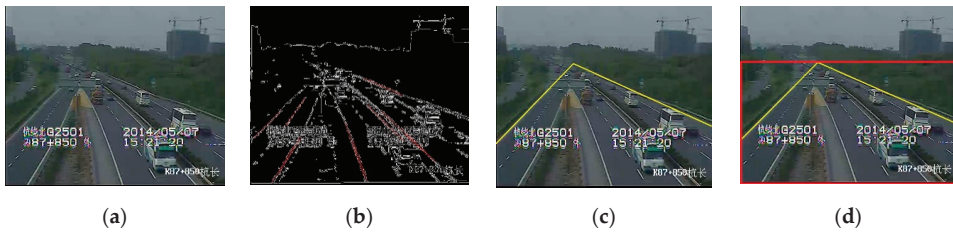


Figure 5. Lane space separation: (a) original Image; (b) lane candidates; (c) driveway boundary; (d) result for lane separation.

In a traffic video detection system, each camera is located at a fixed position and captures the same traffic scenes for a long time. Based on the time continuity, the result of lane space separation for the initial frame of a traffic video can be used over a long time period. Lane space separation can decrease the area of haze removal and improve the efficiency of the dehazing algorithm. Figure 6 shows the haze removal results with and without lane separation. In this scene, the dehazing of 2000 frames needs 35.301 s without lane separation and 32.74 s with lane separation (lane space separation takes 0.182 s). Although lane separation requires some time, the operation just occurs in the first frame. Thus, the time for lane separation can be shared by all frames of a traffic video. With an increasing number of frames, the efficiency of the dehazing algorithm with lane separation will be improved more significantly. Hence, if the driveway region is a larger portion of a whole image, the processing time can be decreased obviously. When real-time processing is required, a little reduction in processing time has been of practical significance.



Figure 6. Results for video dehazing with lane separation: (a) before haze removal; (b) haze removal without lane separation; (c) haze removal with lane separation.

4.4. Optimization Based on Spatial Distribution of Cameras

With an increasingly complex layout of transportation networks, the number of traffic monitoring cameras also increases gradually, and sometimes there are multiple cameras in the same section of road. These cameras located in close physical proximity usually have the same hardware indicators. In a traffic video detection system, multiple cameras are connected to one system. These cameras have similar characteristics according to their spatial distribution. The weather is also an index with spatial characteristics, that is, the degrees of haze are similar in nearby regions. Thus, we can use the spatial distribution information of cameras to speed up dehazing and optimize the performance of the traffic video detection system.

Figure 7 shows the images captured by four surveillance videos of DE-elevated freeways in Hangzhou City at the same time. The locations of these cameras are shown in Figure 8, where the distance between the cameras is about 500 to 600 m. Table 3 shows the initial transmission values of these four videos. The haziness flag values T calculated from each video are shown in the first column of Table 3. We obtain relatively proper initial transmission correction value X' by using the method proposed in Section 3, and then determine the initial transmission value $T * X'$. According to the results, these initial transmission values are very numerically similar, thus there may be no obvious influence on the restored images.

In traffic video dehazing, the cameras are divided into different regions according to their locations, and one camera in a region is set as the calibration camera. The images from the calibration camera are used to calculate the initial transmission value, which is also applied to other cameras in the same region. Therefore, we can avoid repeatedly calculating the values of T , C , and X' for other cameras, thus improving the efficiency of haze removal. The results of haze removal with the initial transmission value obtained by calibration cameras is shown in Figure 9b, and the result directly using the initial transmission value obtained by the image itself is shown in Figure 9c. It is obvious that the results are very similar in these two ways. It takes 0.033 s to calculate the initial transmission value, which can be saved by using that of the calibration camera.

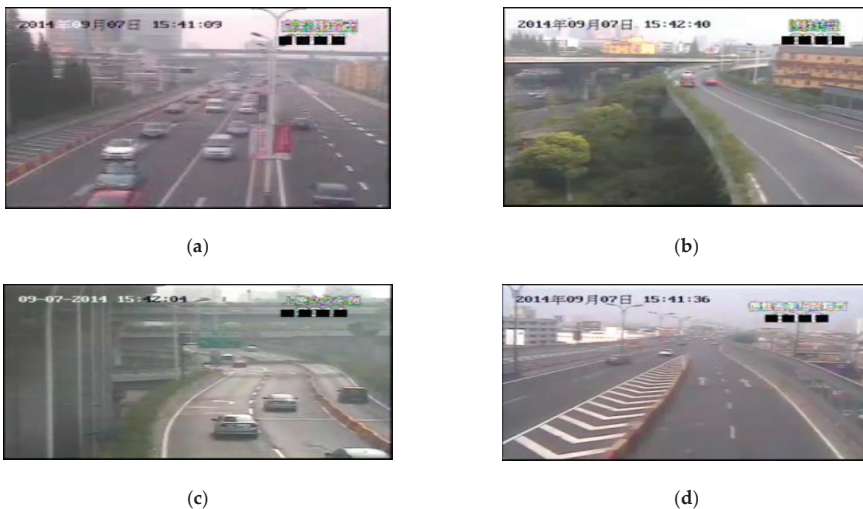


Figure 7. Example images of the nearby regions.

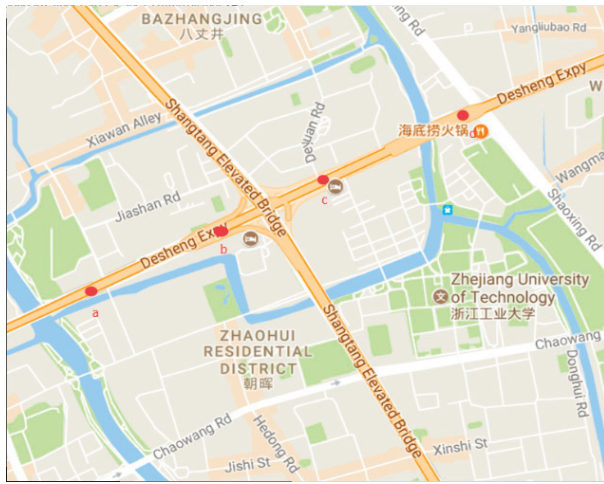


Figure 8. The locations of cameras.

Table 3. Initial transmission values for videos in nearby regions.

| Cases | Haze Flag Value T | Initial Transmission Correction Value X' | Initial Transmission Value $T * X'$ |
|-------|---------------------|--|-------------------------------------|
| a | 0.524188 | 1 | 0.524 |
| c | 0.580732 | 1 | 0.581 |
| b | 0.569918 | 1 | 0.570 |
| d | 0.517431 | 1 | 0.517 |



Figure 9. Results of haze removal with and without calibration camera: (a) original image; (b) initial transmission value for calibration camera is 0.596; (c) initial transmission value for image itself is 0.578.

5. Results

In the efficient traffic video dehazing method using adaptive dark channel prior and spatial-temporal correlations, a video sequence is converted into YUV color space where Y represents the luminance and U/V represents the chromaticity. Human eyes are more sensitive to high-frequency signals than low-frequency signals and more sensitive to changes in visibility than changes in color. The U and V components are less affected by haze than the Y component. Thus, we can only adopt the luminance (Y) component to reduce computational complexity. In our experiments, we implemented each method with Opencv and C/C++ language. The source codes were compiled with Microsoft Visual Studio 2010 and run on an Intel Core I5-2400 processor and 4 GB of main memory running a Windows 7 system.

5.1. Results for Single Image Dehazing

Our adaptive method can determine the initial transmission according to the image characteristics, thus it can produce a more satisfactory dehazing result than the method with fixed initial transmission. Figure 10 shows the restored images using our adaptive method, and there are four different initial transmission values, 0.1, 0.2, 0.3, and 0.4. It is obvious from the experimental results that the smaller initial transmission values may lead to some blocks in the images with overstretched contrast, therefore the optimal initial transmission for the first image is between 0.2 and 0.3, the value for the second image is between 0.3 and 0.4, and the value for the third and fourth images is above 0.4. The $T * X'$ values for the images obtained by our method are all located in the range of the optimal initial transmission. Therefore, our method is adaptable for images with different degrees of haze.

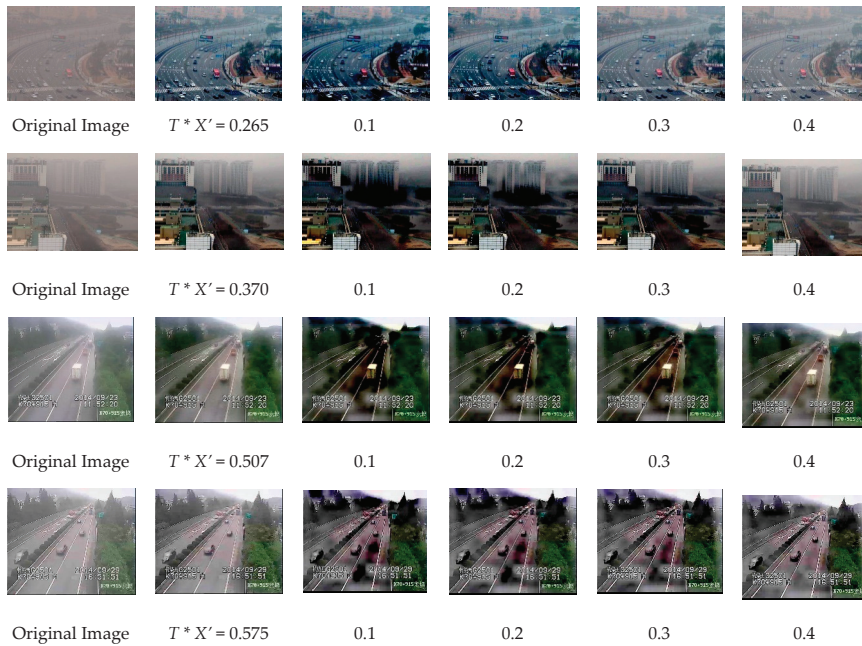


Figure 10. Results for different initial transmission using our adaptive method.

Figure 11 shows four images from Foggy Road Image Database (FRIDA) [33] and restored these images using the dark-channel-prior-based method [9,31], the visibility enhancement algorithm [34], the image-contrast-enhanced method [25], the non-local image dehazing method [20,21], and our method. The SSIM values in Figure 7 are the average values of three channels of RGB. In FRIDA [33], each image without fog is associated with some hazy images, and different kinds of fog are added in each image—uniform fog, heterogeneous fog, cloudy fog, and cloudy heterogeneous fog. According to the experimental results, the dark-channel-prior-based method does not have satisfactory results for haze removal in heterogeneous fog and cloudy heterogeneous fog, while the image-contrast-enhanced method and our method achieves more satisfactory results for these two cases. In addition, our method obtains the highest SSIM for the restored images compared to the first three methods, thus the restored images using our method are more similar to ground truth. As to the results of non-local image dehazing method [20,21], the SSIM for some restored images may be higher than those of our method. However, the non-local image dehazing method takes longer processing time, as shown in Table 4. Table 4 provides the overall processing times of these methods. Our method is faster than the dark-channel-prior-based method [9,31] and visibility enhancement algorithm [34]. However,

our method takes more time than the image-contrast-enhanced method [25] because it spends some time in calculating the image haziness flag value and the initial transmission correction value. However, the results for haze removal using the proposed method are better than the results of the image-contrast-enhanced method. Although the non-local image dehazing method can get more satisfactory restored images, it is too slow to be used in real-time scenarios. In addition, it usually needs to manually set the parameters to different scenes, which is not suitable for real-time traffic video processing. Further still, we can spread this part of the computation time over all frames in video dehazing and reach a faster dehazing speed through the fusion of spatial and temporal information.









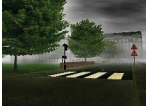







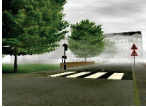



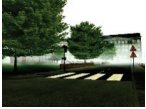







| | | | | |
|--|---|---|---|--|
| Original Images |  |  |  |  |
| Ground Truths |  |  |  |  |
| Dark-channel-prior-based method [9,31] |  |  |  |  |
| SSIM | 0.6468 | 0.6586 | 0.6205 | 0.6397 |
| Visibility Enhancement Algorithm method [34] |  |  |  |  |
| SSIM | 0.5978 | 0.5210 | 0.5412 | 0.5622 |
| Image-contrast-enhanced method [25] |  |  |  |  |
| SSIM | 0.7120 | 0.6970 | 0.7177 | 0.6801 |
| Non-local Image Dehazing [20,21] |  |  |  |  |
| SSIM | 0.6797 | 0.7931 | 0.8035 | 0.7956 |
| Our method |  |  |  |  |
| $T * X'$ | 0.5080 | 0.2740 | 0.2390 | 0.2540 |
| SSIM | 0.7630 | 0.7072 | 0.7391 | 0.7087 |

Figure 11. Comparison of the restored images using different methods; * SSIM = structural similarity.

Table 4. Processing times for single-image dehazing.

| Image Resolution | Dark-Channel-Prior Method [9,31] | Visibility Enhancement Algorithm [34] | Image-Contrast-Enhanced Method [25] | Dehazing Only Using Adaptive Dark Channel Prior | Non-Local Image Dehazing [20,21] | Our Method |
|------------------|----------------------------------|---------------------------------------|-------------------------------------|---|----------------------------------|------------|
| 640 × 480 | 0.897 s | 1.014 s | 0.396 s | 0.506 s | 2.546 s | 0.433 s |
| 480 × 400 | 0.516 s | 0.895 s | 0.165 s | 0.301 s | 2.387 s | 0.252 s |
| 320 × 240 | 0.173 s | 0.348 s | 0.057 s | 0.262 s | 2.024 s | 0.211 s |

5.2. Results for Traffic Video Dehazing

To get better restored images, we restore the whole image for the first frame of a time slice and use the area outside the lane space of the restored frame to replace those areas of the following frames. Moreover, we adopt the parallel programming tools SIMD [35] and OpenMP [36] for rapid calculation. Figure 12 presents a comparison of three approaches for traffic video dehazing, where Figure 12a shows the original videos; Figure 12b shows the results for the dark-channel-prior-based method with guided filtering [9,31], which uses the transmission map obtained from the first frame to filter the following frames; Figure 12c shows the results for the image-contrast-enhanced method [25], whose initial transmission is a constant value 0.3; Figure 12d shows the results produced by our method. Experimental results demonstrate that the image-contrast-enhanced method leads to some blocks with overstretched contrast, such as the images in groups (1), (3), and (4). For some urban scenes, the color is not obviously different between the driveway and background, such as the examples in group (1) with medium haze and group (2) with dense haze. Our method can restore these videos in a manner more similar to the haze-free scenes, and the driveway and the vehicles can be seen more clearly. However, the dark-channel-prior-based method cannot deal with these videos. For the suburban scenes where the trees and road surface are obviously different in color, such as images in group (3) that were captured in daytime and images in group (4) that were captured in dense haze with vehicle headlights on, our method achieves better restored results than the other two methods. For the restored images using our method in group (3), the driveway color is more uniform. For the restored images using our method in group (4), there are no blocks with overstretched contrast, and the color of trees with hierarchical structure is more realistic. Therefore, our method can maintain the image details and restore images that are more similar to the real scene with proper contrast.

As we can see from the experiment results, our method produces better haze removal results by determining parameters according to image characteristics. It is also applicable to dense fog or a variety of fog densities. Moreover, it makes the restored images more similar to the real scene and avoids the problem that the restored images exhibit overstretched contrast. Therefore, it can solve the general problems in the existing dehazing algorithms—contrast distortion after video dehazing and failure to remove dense haze.

In addition, our method adopts the spatial correlation, time continuity, lane separation, and spatial distribution of cameras to improve computational efficiency. Besides the processing time, the performance parameters of frames per second (fps) and SSIM of different methods for the video dehazing in Figure 12 are shown in Table 5. In order to meet the actual traffic scenarios, we process the video frame by frame, and the data show the total processing time for 1000 frames. Our method uses the initial frame in a time slice to calculate the transmission map and atmospheric light and adopts the lane separation to decrease the dehazing areas. Compared with other methods, the time of dehazing in our method decreases when the time slice increases. According to the experiment results, our method can obviously speed up video dehazing, especially if the video has high resolution or the driveway is only a small part of the whole image. Our method can restore the video with a resolution of 720 × 592 at about 57 fps, nearly four times faster than dark-channel-prior-based method and one time faster than image-contrast-enhanced method. Furthermore, our method obtains the highest SSIM for the restored videos compared with other existing methods, thus the restored videos using our method are more similar to ground truth. Therefore, the proposed method not only has superior haze removing and color balancing capabilities but also restores and enhances the degraded videos in real time.

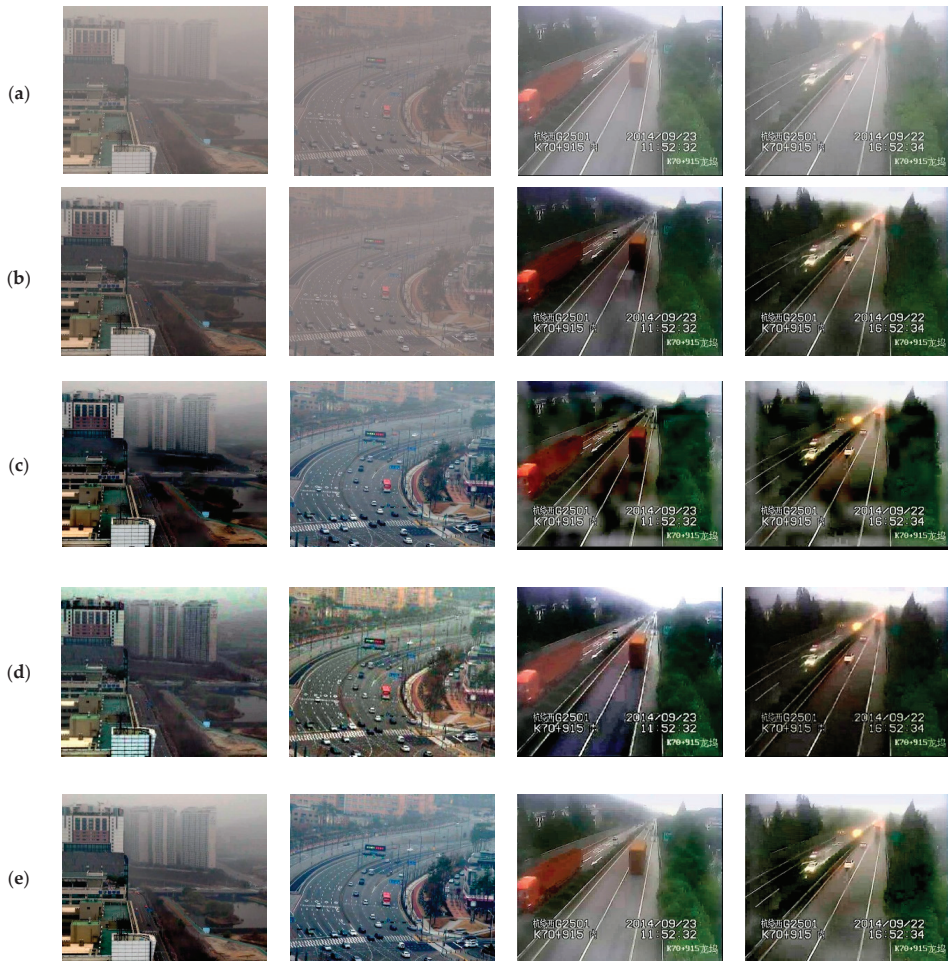


Figure 12. Comparison of restored videos. (a) Original Videos; (b) Dark-channel-prior-based method; (c) Image-contrast-enhanced method; (d) Non-local Image Dehazing; (e) Our method.

Table 5. Comparing the performance parameters.

| Case | Image Resolution | He et al. [9,31] | | | Kim et al. [25] | | | Our Method | | |
|------|------------------|------------------|------|--------|-----------------|------|--------|------------|------|--------|
| | | Time | fps | SSIM | Time | fps | SSIM | Time | fps | SSIM |
| (1) | 640 × 480 | 66.787s | 15.0 | 0.6870 | 35.359 s | 28.3 | 0.6990 | 17.507 s | 57.1 | 0.7012 |
| (2) | 640 × 480 | 64.576 s | 15.4 | 0.7002 | 34.471 s | 29.0 | 0.7079 | 18.005 s | 55.5 | 0.7232 |
| (3) | 720 × 592 | 95.638 s | 10.5 | 0.6155 | 37.858 s | 26.4 | 0.6322 | 17.604 s | 56.8 | 0.6488 |
| (4) | 720 × 592 | 90.911 s | 11.0 | 0.5932 | 39.855 s | 25.1 | 0.6011 | 16.925 s | 59.0 | 0.6155 |

6. Conclusions

Traditional haze removal methods fail to restore the images with different degrees of haziness in a real-time and adaptive manner under most circumstances. To solve this problem, we propose an efficient traffic video dehazing method using adaptive dark channel prior and spatial-temporal correlations. The dark channel prior is based on the statistics of outdoor haze-free images, but it cannot adaptively estimate the initial transmission value based on the degree of haze and contrast

of images. Therefore, we adopt the image-contrast-enhanced method to obtain the best estimated transmission value as the initial transmission value of dark channel prior. The image dehazing method using adaptive dark channel prior can overcome the shortcomings of existing dehazing algorithms that overstretch contrast after haze removal and deal with images with dense haze to a satisfactory level. Additionally, we introduce the temporal-spatial correlation of traffic videos to speed up the traffic video dehazing using the time continuity to set a time slice, the characteristics of block structure to refine transmission, lane space structure to decrease the restored area, and multi-camera distribution to simplify the calculation of parameters. The experiment results show that our method can restore satisfactory image appearance, which can remove dense haze effectively and does not produce results with overstretched contrast. The temporal and spatial characteristics can reduce the computation time, especially for dehazing multiple videos.

However, the dark channel prior is a kind of statistic, and it may not work for some particular traffic videos. When there are rapidly changing hazes in the videos, the dark channel of the scene radiance has a great difference at different times. In addition, if the scene objects are inherently similar to the atmospheric light and no shadow is cast on them, the adaptive dark channel prior is invalid. The dark channel of the scene radiance has bright values near such objects. As a result, our method may underestimate the transmission of these objects and overestimate the haze layer.

Author Contributions: Formal analysis, G.Z. and J.W.; methodology, T.D., Y.Y. and Y.S.; project administration, T.D.; validation, Y.Y.; literature search, J.W. and G.Z.; writing-original draft, T.D. and J.W.; writing-review and editing, G.Z. and Y.S.

Funding: This work is supported by National Natural Science Foundation of China (No. 61672414, 61572437).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pyka, K. Wavelet-Based Local Contrast Enhancement for Satellite, Aerial and Close Range Images. *Remote Sens.* **2017**, *9*, 25. [[CrossRef](#)]
2. Li, R.; Pan, J.; Li, Z.; Tang, J. Single Image Dehazing via Conditional Generative Adversarial Network. In Proceedings of the CVPR Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 July 2018; pp. 8202–8211.
3. Mangeruga, M.; Bruno, F.; Cozza, M.; Agrafiotis, P.; Skarlatos, D. Guidelines for Underwater Image Enhancement Based on Benchmarking of Different Methods. *Remote Sens.* **2018**, *10*, 1652. [[CrossRef](#)]
4. Oakley, J.P.; Satherley, B.L. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *IEEE Trans. Image Process.* **1998**, *7*, 167–179. [[CrossRef](#)]
5. Narasimhan, S.G.; Nayar, S.K. Removing weather effects from monochrome images. In Proceedings of the CVPR Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; pp. 186–193.
6. Chen, G.; Wang, T.; Zhou, H. A Novel Physics-based Method for Restoration of Foggy Day Images. *J. Image Graph.* **2008**, *13*, 888–893.
7. Tan, R.T. Visibility in bad weather from a single image. In Proceedings of the CVPR Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
8. Fattal, R. Single image dehazing. In Proceedings of the ACM Siggraph, Los Angeles, CA, USA, 11–15 August 2008; pp. 1–9.
9. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. In Proceedings of the CVPR Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1956–1963.
10. Lai, Y.; Chen, Y.; Chiou, C.; Hsu, C. Single-Image Dehazing via Optimal Transmission Map Under Scene Priors. *Circuits Syst. Video Technol.* **2015**, *25*, 1–14.
11. Zhu, Q.; Mai, J.; Shao, L. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Trans. Image Process.* **2015**, *24*, 3522–3533.
12. Yeh, C.; Kang, L.; Lee, M.; Lin, C. Haze effect removal from image via haze density estimation in optical model. *Opt. Express* **2013**, *21*, 27127–27141. [[CrossRef](#)] [[PubMed](#)]
13. Li, B.; Wang, S.; Zheng, J.; Zheng, L. Single image haze removal using content-adaptive dark channel and post enhancement. *IET Comput. Vis.* **2014**, *8*, 131–140. [[CrossRef](#)]

14. Wang, J.; He, N.; Zhang, L.; Lu, K. Single image dehazing with a physical model and dark channel prior. *Neurocomputing* **2015**, *149*, 718–728. [[CrossRef](#)]
15. Huang, S.; Chen, B.; Wang, W. Visibility Restoration of Single Hazy Images Captured in Real-World Weather Conditions. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1814–1824. [[CrossRef](#)]
16. Riaz, I.; Fan, X.; Shin, H. Single image dehazing with bright object handling. *IET Comput. Vis.* **2016**, *10*, 817–827. [[CrossRef](#)]
17. Sun, K.; Wang, B.; Zhou, Z. Real time image haze removal using bilateral filter. *Trans. Beijing Inst. Technol.* **2011**, *31*, 810–814.
18. Wang, D.; Fan, J.; Liu, Y. A foggy video images enhancement algorithm of monitoring system. *J. Xian Univ. Posts Telecommun.* **2012**, *5*, TP391.41.
19. Kumari, A.; Sahdev, S.; Sahoo, S.K. Improved single image and video dehazing using morphological operation. In Proceedings of the IEEE International Conference on VLSI Systems, Architecture, Technology and Applications, Bangalore, India, 8–10 January 2015; pp. 1–5.
20. Berman, D.; Treibitz, T.; Avidan, S. Non-Local Image Dehazing. In Proceedings of the CVPR Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1674–1682.
21. Berman, D.; Treibitz, T.; Avidan, S. Air-light Estimation using Haze-Lines. In Proceedings of the IEEE 13th International Conference on Intelligent Computer Communication and Processing, Stanford, CA, USA, 12–14 May 2017; pp. 5178–5191.
22. Tarel, J.; Hautière, N.; Cord, A.; Gruyer, D.; Halmaoui, H. Improved visibility of road scene images under heterogeneous fog. In Proceedings of the IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 478–485.
23. Zhang, J.; Li, L.; Zhang, Y.; Yang, G.; Cao, X.; Sun, J. Video dehazing with spatial and temporal coherence. *Vis. Comput.* **2011**, *27*, 749–757. [[CrossRef](#)]
24. Shin, D.K.; Kim, Y.M.; Park, K.T.; Lee, D.; Choi, W.; Moon, Y.S. Video dehazing without flicker artifacts using adaptive temporal average. In Proceedings of the IEEE International Symposium on Consumer Electronics, Jeju Island, Korea, 22–25 June 2014; pp. 1–2.
25. Kim, J.; Jang, W.; Sim, J.Y.; Kim, C.S. Optimized contrast enhancement for real-time image and video dehazing. *J. Vis. Commun. Image Represent.* **2013**, *24*, 410–425. [[CrossRef](#)]
26. Narasimhan, S.G.; Nayar, S.K. Vision and the Atmosphere. *Int. J. Comput. Vis.* **2002**, *48*, 233–254. [[CrossRef](#)]
27. Pan, X.; Xie, F.; Jiang, Z.; Yin, J. Haze Removal for a Single Remote Sensing Image Based on Deformed Haze Imaging Model. *IEEE Signal Process. Lett.* **2015**, *22*, 1806–1810. [[CrossRef](#)]
28. Peli, E. Contrast in complex images. *J. Opt. Soc. Am. A* **1990**, *7*, 2032–2040. [[CrossRef](#)]
29. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]
30. Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 228–242. [[CrossRef](#)] [[PubMed](#)]
31. He, K.; Sun, J.; Tang, X. Guided image filtering. In Proceedings of the Springer ECCV European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; pp. 1–14.
32. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: New York, NY, USA, 2010.
33. Foggy Road Image Database FRIDA. Available online: <http://www.lcpc.fr/english/products/image-databases/article/frida-foggy-road-image-database> (accessed on 8 June 2012).
34. Huang, S.; Chen, B.; Cheng, Y. An Efficient Visibility Enhancement Algorithm for Road Scenes Captured by Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2321–2332. [[CrossRef](#)]
35. Patterson, D.A.; Hennessy, J.L. *Computer Organization and Design: The Hardware/Software Interface*; Morgan Kaufmann Publishers: Burlington, MA, USA, 1998.
36. Chapman, B.; Jost, G.; van der Pas, R. *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*; MIT Press: Cambridge, MA, USA, 2008.





Review

A Review of Data Analytic Applications in Road Traffic Safety. Part 1: Descriptive and Predictive Modeling

Amir Mehdizadeh ^{1,†}, Miao Cai ^{2,†}, Qiong Hu ¹, Mohammad Ali Alamdar Yazdi ³, Nasrin Mohabbati-Kalejahi ⁴, Alexander Vinel ¹, Steven E. Rigdon ² and Karen C. Davis ⁵ and Fadel M. Megahed ^{6,*}

¹ Department of Industrial and Systems Engineering, Auburn University, Auburn, AL 36849, USA; azm0127@auburn.edu (A.M.); qzh0011@auburn.edu (Q.H.); alexander.vinel@auburn.edu (A.V.)

² College for Public Health and Social Justice, Saint Louis University, St. Louis, MO 63103, USA; miao.cai@slu.edu (M.C.); steve.rigdon@slu.edu (S.E.R.)

³ Carey Business School, Johns Hopkins University, Baltimore, MD 21202, USA; yazdi@jhu.edu

⁴ Jack H. Brown College of Business and Public Administration, California State University at San Bernardino, San Bernardino, CA 92407, USA, nasrin.mohabbati@csusb.edu

⁵ Department of Computer Science and Software Engineering, Miami University, Oxford, OH 45056, USA; karen.davis@miamioh.edu

⁶ Farmer School of Business, Miami University, Oxford, OH 45056, USA

* Correspondence: fmegahed@miamioh.edu

† These authors contributed equally to this work.

Received: 5 January 2020; Accepted: 12 February 2020; Published: 18 February 2020

Abstract: This part of the review aims to reduce the start-up burden of data collection and descriptive analytics for statistical modeling and route optimization of risk associated with motor vehicles. From a data-driven bibliometric analysis, we show that the literature is divided into two disparate research streams: (a) predictive or explanatory models that attempt to understand and quantify crash risk based on different driving conditions, and (b) optimization techniques that focus on minimizing crash risk through route/path-selection and rest-break scheduling. Translation of research outcomes between these two streams is limited. To overcome this issue, we present publicly available high-quality data sources (different study designs, outcome variables, and predictor variables) and descriptive analytic techniques (data summarization, visualization, and dimension reduction) that can be used to achieve safer-routing and provide code to facilitate data collection/exploration by practitioners/researchers. Then, we review the statistical and machine learning models used for crash risk modeling. We show that (near) real-time crash risk is rarely considered, which might explain why the optimization models (reviewed in Part 2) have not capitalized on the research outcomes from the first stream.

Keywords: crash risk modeling; data visualization; descriptive analytics; highway safety; predictive analytics

1. Introduction

Despite the significant technological advances in motor vehicle sensing technologies (e.g., lane departure detection and collision mitigation sensing systems), road crashes have remained a pressing global health issue. The World Health Organization estimated that road injuries are the 8th leading cause of death worldwide, resulting in 1.4 million deaths annually [1]. Perhaps more importantly, the incidence of such crashes and their severity are on the rise. By 2030, traffic-related deaths are predicted to become the 7th leading cause of death worldwide [1]. The increase in annual deaths is seen

in low- and high-income countries alike. For example, in the U.S., an estimated 37,133 people died in road crashes in 2017 [2], which constituted a 7.5% increase from the average annual deaths recorded in 2012–2016 [3]. In addition to the massive loss of life, motor vehicle (which is used to capture passenger cars, motorcycles, buses and trucks) crashes cause significant economic losses. According to the World Health Organization [4], “road traffic crashes cost most countries 3% of their gross domestic product.” In the U.S., it is estimated that the total value of societal harm from motor vehicle crashes exceeds \$830 billion annually [5], which is equivalent to $\approx 4.4\%$ of the country’s gross domestic product [6].

Consequently, there are multiple diverse streams of research dedicated to curbing such driving-related risks. This review focuses on data analytics approaches, which revolve around the idea of using data to characterize and predict traffic risk in order to prescribe better (safer) routes, driver assignments, rest breaks, etc. With the advances in information technology it is possible to collect ever increasing amounts of relevant data, such as comprehensive incident databases, real-time driving data feeds, or relevant factor characteristics (e.g., detailed historical and forecasted weather and traffic reports). Further, there has been a tremendous improvement in the variety and capabilities of data analytics tools and methods that can be applied to all steps of modeling (data collection, processing, prediction, or prescription). The goal of this study then, is to pull together and categorize the existing literature on different aspects of research relevant to enabling data-driven analytics approaches to traffic safety.

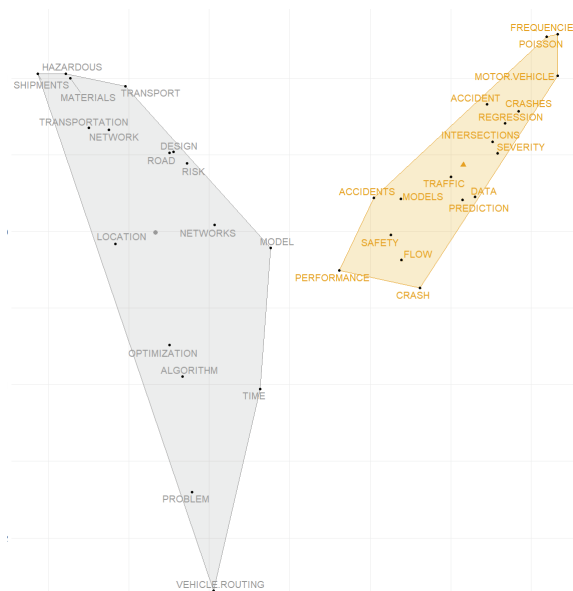
The study was inspired by an observation that there exists an apparent disconnect between two essential facets of pertinent research efforts: statistical modeling of crash risk on one hand and prescriptive modeling for decision making on the other. For example, it is very common in operations research (OR) literature to assume that the crash probability is time-invariant [7,8], and is, in fact, in the range of 10^{-8} to 10^{-6} per mile [9]. This contradicts the findings from the predictive stream of research, with multiple efforts studying the effect of real-time crash risk factors (traffic and weather conditions) on the likelihood of a crash. According to the reviews in [10,11] different traffic and weather conditions would result in different crash risk profiles, bringing into question the effectiveness of the methods often used by OR community for considering risk in decision-making process.

In order to further examine this apparent gap we have conducted a more formal bibliographic study. Based on the keywords and search strategy described in the Supplementary Materials Section, we identified 856 relevant documents (i.e., published articles, proceeding papers, and book chapters). To categorize these documents for this review, a text/bibliometric analysis was performed using the *bibliometrix* R package [12], with the goals of: (a) examining the co-occurrences of keywords within documents since this shows a link between the topics captured by these keywords; and (b) constructing a conceptual structure map of the literature based on a more streamlined keywords list (“Keyword Plus”, see [13] for a detailed introduction). The results are shown in Figure 1a,b, respectively.

In the keyword co-occurrence network, induced by the documents found, a pair of keywords is connected by a link, if they appear in the same document (the links are weighted according to the number of co-occurrences). This network is then clustered with K-means clustering algorithm (all parameters selected automatically by *bibliometrix* package). The clusters and most important links (corresponding to more than four co-occurrences) are depicted on Figure 1a with the black and red links depicting within-cluster and between-cluster connections respectively. The conceptual structure map (Figure 1b) aims at identifying the common emerging concepts in the expanded “Keyword Plus” network. Here, dimensionality reduction technique (multidimensional scaling) is applied to the concept co-occurrence network in order to project it to two dimensions, and the result then clustered with a K-means clustering algorithm. More details on the precise implementation can be found in [12].



(a) A keyword co-occurrence network of the literature, depicting the 60 most used keywords. The nodes correspond to the keywords, with node size reflecting relative frequency. The links are limited to keywords that co-occurred at least five times (black and red lines correspond to between and within clusters, respectively). The network plot divides the literature into two clusters: prescriptive modeling (left), and explanatory /predictive modeling (right).



(b) A data-driven conceptual structure map based on “Keywords Plus” (keywords tagged by the ISI or SCOPUS database scientific experts) and the application of multiple correspondence analysis and *k*-means clustering. The nodes are limited to keywords that have occurred ≥ 5 times, and the gray circle and orange triangle depict the corresponding cluster center. Similar to Figure 1a, the concept map also divides the literature into the same two clusters.

Figure 1. A bibliographic analysis of the literature using the *bibliometrix* package in R.

Based on Figure 1, two important observations can be made. First, the literature can indeed be grouped into two main groups: (a) an explanatory/predictive modeling stream, where the keywords emphasize the collected data (loop detector data), predictors (traffic, weather, time and/or infrastructure), models used (regression, spatial-analysis, Poisson-gamma and negative binomial), and model outcomes (rates, crash frequencies, and crash prediction); and (b) a prescriptive modeling stream, where the focus is on developing algorithms to manage risk, particularly for hazardous materials (hazmat) trucking, through the selection of paths and routes. Second, the cluster agreement between the keyword co-occurrence network and the concept map generated using the Web of Science's Keywords Plus field implies that there is a clear division between the two research streams, despite the fact that the outputs from the first stream should be inputs for the optimization models used for prescriptive decision-making. Based on the second insight and a separate thorough examination of the relevant operations research (OR) literature we can then conclude that the prescriptive literature largely ignores the recent results on factors influencing crash risk.

Against this backdrop, the primary purpose of this review is to help bridge the gap between the different research streams that relate to the modeling and minimization of crash risk. Our goal is to bring the research into better focus and to encourage future work that crosses the siloed divisions within the literature. To achieve this goal, we divide this review into two parts. Part 1 covers the sensing, data acquisition, data exploration, and explanatory/predictive modeling, i.e., focuses on the first research stream. Part 2 reviews the prescriptive modeling component (i.e., second stream), provides a simple case study for how both streams can be integrated, and presents ideas for future research. Note that the research presented in Part 2 primarily targets hazardous materials (hazmat) trucking operations, where optimization models are used to minimize crash risk through path/route selection and/or rest-break scheduling, while meeting delivery requirements. On the other hand, in Part 1, the research relates to both commuters and commercial drivers since the unit of analysis is a "road segment".

This paper is structured to follow the standard data analytics framework: data collection → data exploration → predictive modeling. The final part—prescriptive modeling—is discussed in Part 2 of this effort. We would like to emphasize that in addition to the need for connecting siloed research streams identified above, there also may exist a relatively high "start-up cost" for initiating new efforts in this area. Specifically, as we survey in the remainder of this paper, there exist multitudes of disparate datasets, data processing approaches and statistical methods that all may be relevant. Hence, the goal of this review is to attempt to reduce this burden by categorizing the existing efforts. The remainder of the first part of this review is structured utilizing a data analytic framework (data collection → data exploration → predictive modeling). We present an overview of the sensors and data collection mechanisms used in these studies in Section 2. In Section 3, we provide a taxonomy and review of the commonly utilized data exploration and summarization techniques. Then, we synthesize the explanatory/predictive modeling techniques used for crash risk modeling in Section 4. We offer our concluding remarks in Section 5, and provide links for our code and analysis in the Supplementary Materials Section.

2. Data Acquisition Protocols: An Overview of the Types of Collected Data and Their Associated Sensing Systems

In this section, we provide an overview of the data acquisition strategies typically used in motor vehicle safety studies as well as a brief introduction to the corresponding sensing systems. The ability to extract such data is an indispensable component in any crash risk prediction study, yet it is typically under-described. Thus, we view this section as an important practical contribution of our review since a potential reason for the gap between the predictive and prescriptive analytic research streams can be attributed to the "large start-up burden", associated with the lack of sufficient/targeted documentation for collecting quality data. While we primarily focus on U.S.-based systems, the protocols described here can be extended to many transportation locales. To facilitate and encourage the collection of

data pertaining to important factor sets (per the reviews of Theofilatos and Yannis [10] and Roshandel et al. [11]) in future prescriptive studies, we provide **R** code that can be used to scrape data for many important crash risk predictors (see the link in our Supplementary Materials Section).

It must be emphasized that both data sources needed and data acquisition methods used to access these sources depend on the design of the study in question. Specifically, since this review is focused on the literature dedicated to models for quantifying crash risks, the corresponding studies can generally be divided into two main study designs: (a) retrospective case-control studies in which police crash reports are used, and (b) prospective naturalistic driving studies (NDS), in which a pre-specified set of drivers is followed for a certain period of time. As one can expect, the choice of study design affects the data collection mechanism (as well as the statistical methodologies used for analysis, which are discussed in Section 4). For the sake of completeness, we provide some background on each of these two design strategies in the following subsection.

2.1. Background: Study Designs

Most research on motor vehicle safety has assumed that the sampling unit is a spatiotemporal snapshot of a highway, i.e., researchers typically study a given section of a highway for a pre-specified time period. Note that it is not sufficient to study the conditions under which crashes tend to occur; one must also study the conditions under which crashes do not occur, and compare the two. The problem is analogous to that faced by epidemiologists when investigating the cause(s) of a disease, where they examine the prior behavior of individuals with and without a disease and attempt to identify differences in their prior behavior. The most common design that epidemiologists use is the case-control design. A number of individuals with the disease are first identified, representing the cases. The demographic and behavioral characteristics (e.g., age, sex, race, smoking status, body mass index, etc.) for the cases are then determined/computed. A control group, as similar as possible to the case group, is then identified. In a matched pair case-control study, each case is matched with one or more control subjects.

In motor vehicle highway safety applications, these retrospective case-control studies are typically conducted using police crash reports. In the U.S., crash reports include information pertaining to number of vehicles, involvement of pedestrians, number of injuries/fatalities, road type, crash location, date-time, intersection type, presence of a nearby work zone, weather conditions, and road surface conditions [14,15]. While a lot of information can be captured in these reports, case-control studies are inherently limited for two main reasons. First, the information captured in the crash reports combines: (a) factual information, e.g., type of road and number of vehicles involved in the crash; (b) information that is estimated by the police officer, e.g., classifying weather into one of pre-defined categories; and (c) information captured from witnesses which is subject to recall and/or information bias, e.g., it is often hard to gauge the veracity of information extracted from drivers involved in the crash. Second, the inference from case-control studies can be limited when the denominator (e.g., non-crashes or healthy individuals) is unknown to the researchers [16]. In highway safety research, traffic flows can be captured using cameras and on-the-road sensors; however, such information is not typically available for every road segment (e.g., in rural local roads and/or for all highway exits). Thus, this is a prevalent issue in existing case-control highway safety studies.

To alleviate the limitations in case-control studies, there has been an increasing number of prospective naturalistic driving studies (NDSs) in the past decade. Contrary to the case-control studies, the information is captured via one or more sensors that are mounted in the vehicle in an effort to collect [17]: (a) high-resolution real-time driving data under real-world circumstances; (b) location/GPS, speed, and multiple views of the driver/road; and (c) naturalistic/individualized driving behaviors that can help explain differences if a crash is observed during the study period. Compared to traditional case-control studies, NDSs resemble prospective cohort studies, where a pre-specified set of drivers is followed for a certain period of time. The sampling units here are the drivers instead of road segments, and all the events or non-events of the sample drivers are collected.

Therefore, it is possible to compare the rates of events in NDSs. In addition, the data are automatically collected using sensors, which minimizes the impact of police/witnesses' judgement in imputing the data and/or estimating values for certain predictors.

2.2. Outcome Variables Used in Crash Risk Modeling

In retrospective case-control studies, the most frequently used outcome variable is crash counts. In the U.S., historical crash data are hosted by different Department of Transportation (DoT) divisions depending on: (a) the types of vehicles involved, i.e., commercial vehicles or personal commuter vehicles; and (b) whether the crash resulted in any fatalities. When these models are utilized/deployed for predictive purposes, real-time traffic data can often be used as model inputs. In the U.S., such data can be obtained from state specific reporting systems. For example, the 511 reporting system highlighted in Figure 2, is the predominately used sensing system in the U.S. since it is used by more than 45 states [18]. On the other hand, in prospective NDSs, the use of safety-critical events (SCEs) as a proxy outcome variable is more common since: (a) NDSs do not focus on crash-prone highways, (b) SCEs have a much higher incidence rate than crashes, and (c) they are assumed to be positively correlated with the incidence of crashes [16,19]. SCEs are defined as events that avoid crashes by last-second evasive maneuver(s) [16]. The most commonly studied SCE is “hard brakes”, which can be detected using accelerometers/inertial measurement units mounted in the vehicle or through a driver's smart phone. The identification of a “hard break” is threshold dependent; for example, several studies equate a “hard break” to a deceleration higher than 3.0 m/s^2 [20,21]. Several detailed reviews have been published on surrogate indicators using in the field of traffic safety [22–24]. It is important to note that, while SCE has been extensively used as the outcome variable in NDSs, its validity and causal relationship with crashes have not yet been conclusively confirmed [25,26]. We provide a visual summary of the hierarchical nature of the described outcome variables in Figure 2.

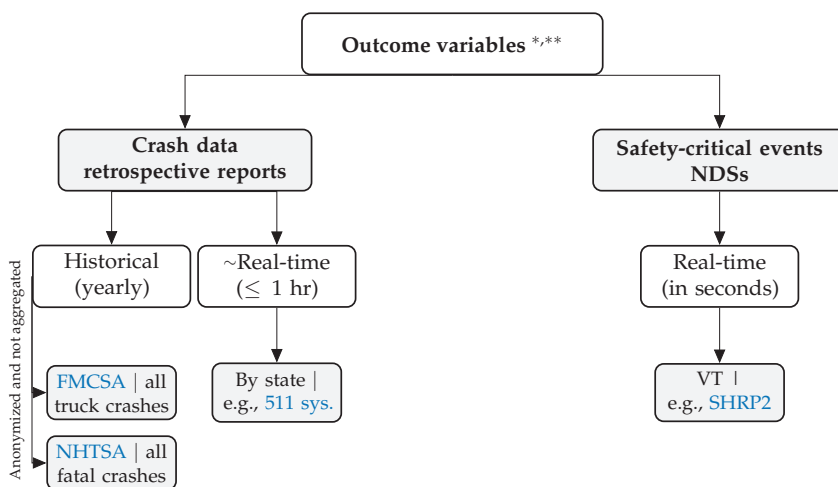


Figure 2. A hierarchical view of outcome variables in crash risk modeling studies. The first level captures the data type, the second level shows the frequency, and the third level highlights examples and sources. * Acronyms: FMCSA = Federal Motor Carrier Safety Administration, NHTSA = National Highway Traffic Safety Administration, VT = Virginia Tech. ** Code: To simplify the data collection process, we present the R code needed to scrape and clean these different data sources at: <https://caimiao0714.github.io/TrafficSafetyReviewRmarkdown/>.

2.3. Predictor Variables Used in Crash Risk Modeling

Factors that have been shown in the literature to contribute to motor vehicle crash risk are discussed in detail in Section 4. Here we concentrate on strategies and sensing technologies used to obtain relevant data. From a data acquisition viewpoint, the sensors can be divided into [27]: (a) intra-vehicular sensing platforms, where conditions extracted from the vehicle are captured, and (b) urban sensing platforms, where the sensors are integrated in the road infrastructure. Intra-vehicular sensors can capture driver behavior, vehicle speed, traffic environment, etc. [28], and are widely used in NDS studies. On the other hand, urban sensing platforms are more commonly utilized in case-control studies. We can categorize such platforms into the following three categories: (a) traffic sensing systems (e.g., traffic cameras, inductive loop detectors, infrared sensors), which can be used to estimate traffic flow, speed, occupancy, and volume [27]; (b) weather sensing systems, which can be used to compute/estimate important factors for both explanatory/predictive (e.g., visibility, rain/snow accumulation, and potential for icy conditions) and prescriptive modeling (e.g., wind direction and speed which are important considerations in hazardous material routing since they are used in predicting the severity of a possible crash through estimating the radius of dispersion of toxic materials); and (c) geometric road descriptors (e.g., number of lanes, speed limit information, longitudinal grade, road shoulder width, and whether the road segment of interest contains a straight, merge, and/or diverge sections), which are typically tagged in geographic information systems (GIS) and can be accessed using popular application programming interfaces (APIs) such as *OpenStreetMaps* [29,30]. A visual summary of predictor variables extracted from urban sensing systems is provided in Figure 3.

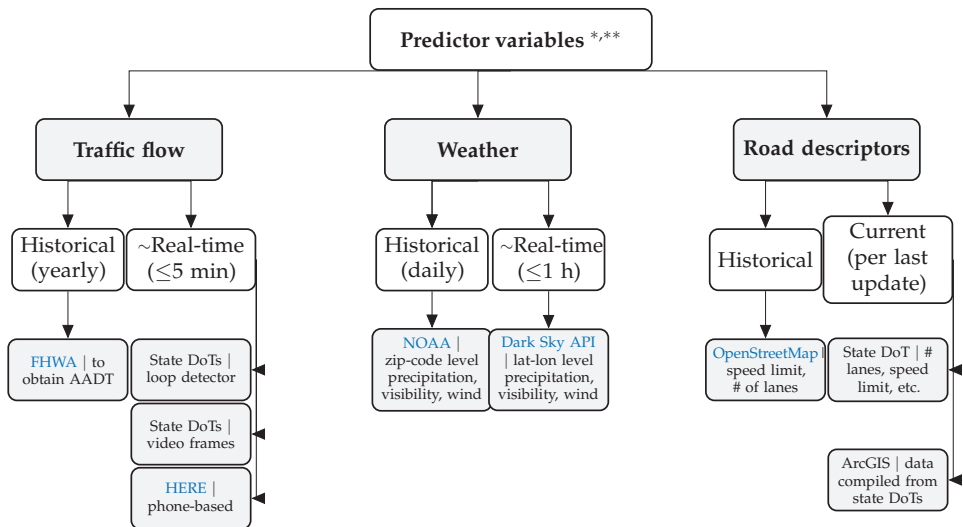


Figure 3. A hierarchy of predictor variables used in modeling crash risk. The first level captures the data type, the second level shows the frequency, and the third level highlights examples and sources. * Acronyms: AADT = Annual Average Daily Traffic, FHWA = U.S. Federal Highway Administration, DoT = U.S. Department of Transportation, and NOAA = U.S. National Oceanic & Atmospheric Administration. ** Code: To simplify the data collection process, we present the R code needed to scrape and clean these different data sources at: <https://caimiao0714.github.io/TrafficSafetyReviewRmarkdown/>.

3. Descriptive Analytic Tools Used for Understanding Crash Data

In this section, we review the exploratory data analysis (EDA) techniques used to examine transportation datasets prior to the explanatory/predictive modeling stage. EDA is an especially

important pre-processing steps when dealing with large datasets, where predictive modeling and optimization can be computationally intensive. In Figure 4, we depict the two major goals of EDA as well as the methodologies used to achieve these goals. Note that these methods may not be mutually exclusive and can be used to complement each other.

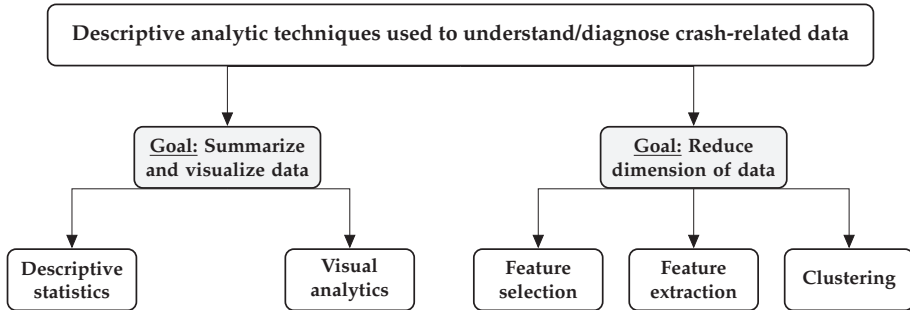


Figure 4. Exploratory data analysis (EDA) goals and their associated techniques/methodological frameworks.

3.1. Data Summarization and Visualization

Data summarization include both univariate (e.g., central tendency, dispersion, etc.) and multivariate tools (e.g., correlation). We assume that both predictive and prescriptive modeling researchers are well-versed with these methods, and thus we will not discuss them here (see Washington et al. [31] for a detailed introduction). As a complement to data summarization, data visualization is a succinct approach to understanding trends, patterns, and anomalies in data. In a survey paper on the application of visualization techniques for traffic datasets, Chen et al. [32] categorized visualization approaches based on four data types: (a) temporal data, (b) spatial data, (c) spatiotemporal data, and (d) multivariate data. This framework can be extended to more comprehensive crash modeling studies where traffic, weather and other predictor sets are combined. Table 1 presents an overview of the appropriate/recommended visualization techniques for each data type, with example references from the literature. In the following subsections, we discuss each of these groups in further detail.

Table 1. Categorizing visualization techniques for transportation data, adapted from Chen et al. [32].

| Variable Type (Main Group) | Subgroup | Visualization Techniques | Examples |
|----------------------------|---------------|---|------------|
| Time-series data | Linear time | Line and stacked graphs | [33–38] |
| | Periodic time | Radial layout and cluster-and-calendar based visualization | [38,39] |
| Spatial | Point-based | Symbol maps | [40] |
| | Line-based | Line maps, edge bundling, and kernel density estimation charts (KDE) | [41,42] |
| | Region-based | Radial metaphor charts, choropleth, proportional symbol maps, and heat maps | [43–47] |
| Spatiotemporal | - | Space-Time-Cube (STC), animated maps, GeoTime, and stacking-based STC | [48–52] |
| Multiple properties | - | Parallel coordinates plot, trellis plot, and multidimensional scaling | [45,53–59] |

3.1.1. Visualization of Time-Oriented Data

Line graphs are the most frequently used visualization technique for time-oriented data, where the x -axis represents time and y -axis demonstrates transportation-related variable. There are numerous applications of line graphs in traffic/crash visualizations, for example, visualizations of tips per trip and fare per miles-driven among New York City taxi drivers [36], carbon monoxide pollution over the course of the day in London [60], traffic volumes in Beijing, China [33] and Porto, Portugal [34], or the effects of road surface conditions and time of day on traffic volumes [35]. Since line graphs can become visually overwhelming as the number of variables increases. Other time-series based graphs can be considered in this case, such as *ThemeRiver stacked chart* [61], which uses a flowing river metaphor to capture changes in several variables of interest over time. This chart was used by Guo et al. [37] for understanding traffic volume patterns.

When the data are inherently periodic or cyclic, three charts can be applied [32]: radial layout, cluster- and calendar-based (where line graphs are used for showing cluster averages over time, and calendar-based charts are used to show cluster membership per day) [62], and statistically derived charts. Pu et al. [39] used the radial layout chart to depict traffic volumes in different days and times. Tsai et al. [38] showed how the cluster- and calendar-based charts can be effective in understanding traffic flows in the state of Alabama. In their case study, they showed that the data exhibited eight distinct clusters of daily traffic volumes (at hourly intervals within each day). Two of the clusters were somewhat unexpected, where one captured game-day traffic for college football, and the other captured travel patterns around different holidays (including Fourth of July, Thanksgiving, and Christmas). Statistically derived plots (based on time-series analysis techniques) can be used to quantify the periodic/seasonal nature of the data. From a time-series analysis perspective, the data can be decomposed into: (a) seasonal, (b) trend, and/or (c) cyclical components within a season. These components can be visualized, along with the autocorrelation function (ACF) and the partial autocorrelation function (PACF) for the differenced series to provide an understanding of what type of time-series models to use. The reader is referred to [31] for a detailed coverage of time-series modeling applied to transportation data analyses.

3.1.2. Visualization of Spatial and Spatiotemporal Data

Crash datasets provide rich spatial information including the location of vehicles, construction sites, road closures, and crashes. Visualizing them spatially gives insight(s) on the geographical patterns and clusters, which may improve the decisions made when setting up the dataset for predictive/prescriptive modeling. Chen et al. [32] presented three visualization options (point-based, line-based, and region-based visualizations), which should be selected based on the dataset's aggregation level.

In point-based visualizations, each symbol on a map represents the position of an object at a given point in time. An example of such a visualization is the motor vehicle fatality symbol map, which is used by NHTSA to depict fatalities [40]. We provide a screenshot of their dashboard in Figure 5, showing the location of vehicle occupants killed in speed-related crashes on Saturdays in December 2016.

Popularized by the ubiquity of modern navigation applications, a line map visualizes travel routes and traffic flow. An example can be found at [42], who presented the trip patterns in Bristol, England. They used the "line width" to encode the number of trips and "color" to encode active travel percents. Given the widespread use of navigation applications, we do not discuss other examples in this review.

Region-based spatial visualizations include three popular visualization techniques. The first is the "proportional symbols map" [43], where the size of a point/symbol in a map is proportional to the number of observations in that location. This can be seen as an extension to the point-based visualization, where the point-position on the map is now used to encode count. The second technique is based on "choropleth maps" [44–46], where areas/regions in maps are shaded, colored, or patterned relative to the value of the metric of interest. These maps are common when comparing crash/fatality rates between larger geographic regions (e.g., counties, states, or countries). The third, and least

commonly used visualization is the “radial metaphor”. One existing application was provided by Zeng et al. [47], who used a “radial metaphor” chart to visualize interchanging traffic patterns among different regions of a city.

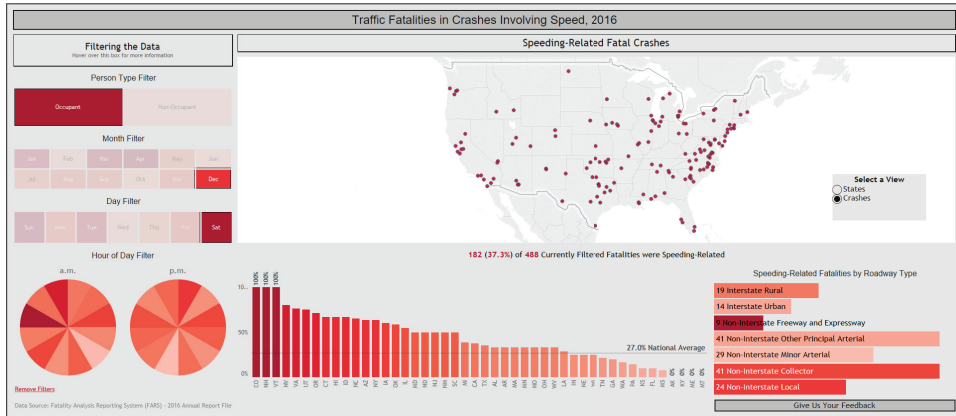


Figure 5. Symbol map showing the location of vehicle occupants killed in speeding-related crashes in the US in December, 2016. The dashboard is available at [40].

For spatiotemporal visualizations, there are two overarching strategies that can be used. The first strategy is intended for web-based visualizations, where a time effect is added to the map by animation or transition effects. Examples can be found in [50,51]. On the other hand, the second strategy is intended for print and utilizes dedicated visualization methodologies. Space-time-cube (STC) visualizations, are the most commonly utilized approach, where the x and y axes are used to capture spatial information, while the temporal information is shown on the z axis [48]. Applications of such technique include: (a) traffic analysis where the changes in a traffic-related variable of multiple vehicles across time and space is shown by stacking-based STC [52]; and (b) crash analysis where crashes are displayed and tracked based on their spatiotemporal information by an enhanced version of standard STC [49,63]. Despite their perceived utility for showing spatiotemporal patterns in a 2-dimensional screen/paper, we do not recommend this approach since the actual values cannot be easily shown and comparisons depend on one’s ability to estimate the patterns over space and time. Instead, we would recommend the use of either panel visualizations (i.e., trellis/ small multiples), or a tabulated representation of the results to show the time component.

3.1.3. Visualization of High-Dimensional Datasets

For high-dimensional data, visualization requires more data cleaning and curation. On the lower end of the spectrum, Parallel coordinates plots (PCP) and trellis (small multiples of bar charts or scatter plots) are commonly used fast plotting tools and require less data preprocessing. For example, PCP can be applied to visualize the correlation/interaction among several crash descriptors including: cars involved, day/month effects, incident type, and road condition [45,53,55]. Additionally, the trellis plot was used by Cottrill and Thakuriah [54] to visualize variations in the number of crashes by different census tracts. On the upper end of the analytical spectrum, visualizations are preceded with the application of projection methods to reduce the problem’s dimensionality. Examples include: (a) Van Huysduyven et al. [57] where cluster analysis and multidimensional scaling were used to produce a 2-dimensional (2D) plot of the relationship between the different constructs and types of drivers examined in the study; (b) Das et al. [59] who utilized multiple correspondence analysis (MCA) to present a proximity map of key factors contributing to wrong-way driving in a 2D space;

(c) Liu et al. [58] where the multivariate time-series data capturing the driver behavior were reduced to a 3D feature space using deep learning techniques, and then visualized using a driving color map.

3.2. Dimension Reduction

In the previous subsection, we highlighted how projection methods can be used to reduce the data dimensionality and assist in its visualization. Here, we discuss how dimension reduction techniques can be used to prepare the data for the predictive modeling stage. In general, there are three main goals for dimension reduction: (a) feature selection, where important variables are identified and selected; (b) feature extraction/generation, where the variable set is projected into lower subspace without losing significant information and; and (c) clustering, where similar observations are grouped together. Since researchers could combine these approaches in their analysis, we classified dimension reduction methods according to their goals.

3.2.1. Feature Selection

One of the recommended steps before the use of statistical and machine learning models is to identify and use only the variables/features deemed important for the analysis since this [64]: (a) avoids over-fitting, (b) reduces the computational complexity in the analysis, and (c) leads to better prediction performance. This step is often referred to as variable or feature selection. In the context of crash prediction models, variable selection plays an important role since there are many potential predictors (e.g., traffic, weather, road geometry related variables) which may have effect on the probability of a crash. In addition, in order to capture the spatial and temporal effects of these variables, new variables need to be introduced in the model. For instance, Shi and Abdel-Aty [65] developed a crash prediction model where each traffic-related variable is collected prior to the crash from two upstream and two downstream sensors. This means that the information for each traffic variable is divided across four variables, and that these variables contain some redundant information within them. In such cases, feature/variable selection will improve model performance [66–70]. For the sake of conciseness, hereafter we use the term feature selection to denote feature and variable selection methods.

Feature selection methods can be classified into three groups: filter, wrapper, and embedded methods [71]. In the filter methods, the process of selecting a subset of features is independent from the statistical and machine learning model used, i.e., a subset of features will be selected according to an algorithm (e.g., Pearson correlation or Mutual Information Criterion), and then the selected features will be the inputs to the explanatory/predictive model. Advantages of filter methods include: (a) simplicity, (b) computational efficiency, (c) speed, and (d) reduction of the risk of over-fitting. However, they can ignore the dependency between features and do not guarantee the selection of an optimal set of features [71,72]. In contrast, wrapper methods consider the prediction performance of the classifier (while accounting for the dependencies/interactions between features) and subsets the feature space using heuristic searching algorithms such as genetic algorithms [73] and particle swarm optimization [74]. While they can improve performance when compared to filter methodologies, they are computationally inefficient. In addition, they also do not guarantee optimality and may over-fit [71,72]. To avoid such problems, feature selection is a part of the model training process in embedded approaches, which makes them the preferred approach in many crash risk modeling scenarios. Random forest (RF) was widely used in the literature as a feature selection method and to determine variable importance [69,70,75]. For more information about the feature selection methods and their applications, we refer the reader to Saeys et al. [72], Guyon and Elisseeff [76], Jović et al. [77].

3.2.2. Feature Extraction

Feature extraction methods offer an alternative approach to dimension reduction by projecting input space to a more efficient dimension space. The projection can combine input variables, reduce the problem complexity, and present a useful abstraction of the data [78]. Thus, feature extraction

differs from feature selection as the focus is not on dropping unimportant variables, but rather to combine the information across the variables through a mathematical transformation. Principal Component Analysis (PCA) is the most commonly used feature extraction method in the crash prediction literature [79–84]. Through an orthogonal transformation, PCA transforms the original variables into a set of linearly uncorrelated variables (i.e., principal components, PCs). Typically, the variation in the data can be explained with a few PCs, which reduces the dimensionality of the problem with minor loss of information. The determination of the number of PCs to retain is often determined through a scree plot or a threshold for the eigenvalues [85]. Since PCA was originally designed for numeric variables that can be linearly combined, there are several extensions to PCA which do not require such assumptions. These include: (a) probabilistic PCA [86], (b) non-linear PCA [78], and (c) kernel-based PCA [87]. These methods have also been implemented extensively in the literature [78].

3.2.3. Clustering

Contrary to feature selection and extraction, clustering is an unsupervised machine learning method that attempts to group observations together with the goals of maximizing the similarity within a cluster (i.e., minimizing distance between observations) and minimizing the similarity between clusters (i.e., maximizing the distance between cluster centers/centroids) [88,89]. Clustering approaches can be divided into: partitioning-based, hierarchical-based, density based, grid-based, and model-based methodologies [88,90].

Crash risk modeling datasets have a number of characteristics that make clustering a viable and useful approach for dimension reduction. For example, if you consider traffic datasets, the goal is typically to understand the impact of traffic conditions on crash likelihood, which is typically achieved through: (a) classifying traffic into different states, and then (b) evaluating the impact of each traffic state (e.g., congested or not congested) on the crash likelihood [10]. Historically, step (a) was achieved through an analysis of traffic flow characteristics (e.g., see [91–93]). A limitation of such an approach is that the modeling can be influenced by researchers' biases and perceptions. Alternatively, one can use an assumption-free, data-driven approach to identify how observations can be clustered. Tsai et al. [38] showed how clustering can be used to identify logical, but hard to model, groupings of the data. Applications of clustering include, but are not limited, to: (a) traffic categorization [38,94,95], (b) identifying accident clusters [96–98], and (c) grouping of weather conditions [99]. To demonstrate how an optimal number of clusters (k^*) can be obtained, we provide a detailed example in the Supplementary Materials where we use k -means clustering and the elbow method to determine the k^* clusters for traffic data.

4. Explanatory/Predictive Models for Crash Risk

This section focuses on two aspects: the risk factors that affect crash risk and statistical/machine learning models. In the risk factors part, we specifically consider the effects of fatigue, distracted driving, and environmental variables including traffic and weather on traffic safety. For the statistical part, we will review how some of the research that has been done to analyze those factors and build predictive models.

4.1. Risk Factors for Traffic Safety

Roshandel et al. [11] discussed five sets of factors that affect crash risk: (a) behavioral characteristics of the driver—e.g., impairment, fatigue, distractions; (b) vehicle condition; (c) traffic conditions—e.g., traffic speed, density and variation in speed between vehicles; (d) geometric characteristics of the road, i.e., type of road, number of lanes, curvature, nearby ramps/intersections, etc.; and (e) weather conditions—e.g., rain, visibility, ice/sleet/snow, etc.

4.1.1. Sleep and Fatigue

Early work on the study of fatigue and the risk of adverse outcomes such as crashes relied on sample surveys of drivers. For example, Crum et al. [100] conducted face-to-face interviews with approximately 500 truck drivers at five rest stops on interstates spread across the United States. The three outcomes were “close calls,” “perception of fatigue,” and “crash involvement.” All of these were based on driver recall from survey responses. They identified three sets of variables that could affect drivers’ fatigue, with self-reported measures. These measures included truck driving environments, economic pressures, and carrier support for safety. Three specific variables, all from the truck driving environment category, were identified as influencing fatigue, including: (a) drive regular or irregular shifts; (b) short or long load wait time; and (c) start the work week tired (or not). Crum et al. [100] ran a regression analysis with these factors as predictors, with each of the responses described above. The first variable (drive regular or irregular shifts) was measured by determining how many six-hour times periods the drivers routinely drove. They found that starting the work week tired was a significant predictor for all three outcome measures described above. Long wait times were positively associated with close calls and self-perception of fatigue. Paradoxically, the number of time periods driven per day was negatively associated with close calls.

In another early study, Crum and Morrow [101] conducted a stratified sample of trucking companies based on their safety record. They selected a sample from each of three strata defined as the bottom quartile (poorest safety performers), the middle two quartiles, and the top quartile (the highest safety performers). After taking a sample of carriers within each stratum they sent seven questionnaires to be filled out by various employees in the company, including the executive, the safety director, two dispatchers and three drivers. They also arranged focus groups within each company. Using the same three sets of variables as in [100] they concluded that the most significant variable in predicting fatigue was “starting the workweek tired.” Other significant factors were “difficulty finding a place to rest” and “shipper and receiver scheduling practices and requirements.”

Garbarino et al. [102] conducted a cross-sectional study of truck drivers in Italy to determine the risk factors for accidents and near misses. Data on sleep apnea, sleep debt, daytime sleepiness, frequency of naps, and frequency of rest breaks, as well as the accident responses were conducted from survey questionnaires and medical exams. They found that obstructive sleep apnea, sleep debt, and excessive daytime sleepiness were positively correlated with accidents; these yielded odds ratios of 2.32, 1.45, and 1.73, respectively. Naps and rest breaks were negatively associated with accidents, having odds ratios of 0.59 and 0.63 respectively. All of these odds ratios had confidence intervals that excluded the null value of 1.0.

With automatic data collection systems that can detect events like accidents, hard-brakes (sudden deceleration caused by braking), lane departures, and others. Mollicone et al. [21] studied hard braking as safety critical events, which are highly correlated with crashes [103]. Their model used a predicted fatigue model of McCauley et al. [104] and McCauley et al. [105] to develop a Poisson regression model having the number of hard brakes as the response. The predictor variables included the predicted fatigue and six variables for the time of day. They found that there is an increasing and concave up relationship between the predicted fatigue and the relative risk of a hard brake.

In a recent study, Stern et al. [106] reviewed the research related to fatigue of commercial motor vehicle drivers. Because of the difficulty of running a controlled experiment by imposing treatments, most research designs are observational studies, that is, they compare the effects of variables that are observed, not imposed. One exception to this is a *randomized encouragement design* where drivers are randomized to receive some sort of incentive to apply some treatment, but are not forced to do so. If an effect is observed, we would conclude that it is due to the incentive, not necessarily to the actual treatment. Many studies use a cohort design or a case-control study. In a cohort design, a number of drivers is identified and studied across time. In a case-control study, a number of cases (e.g., crashes) are identified and are matched with controls; focus is then placed on the differences between the cases and controls. Both cohort studies and case-control studies can be useful in assessing safety.

Recently, Bowden and Ragsdale [107] developed an optimization algorithm for driver scheduling. The algorithm, denoted FAST (Fatigue Avoidance Scheduling Tool) was designed to minimize the trip duration subject to a minimum fatigue level along with other constraints, such as the maximum driving hours under United States law. The algorithm assumes the three process model of alertness (TPMA) developed by Åkerstedt and Folkard [108] and Åkerstedt et al. [109].

4.1.2. Distracted Driving

Other researchers have looked at the effect of distracted driving. The problem of mobile phone usage and distracted driving has been noticed by the World Health Organization [110]. They noted that world-wide use of cell phones has increased by up to 11% in the past 5 to 10 years. Their data suggest that cell phone usage increases the chance of a crash by a factor of four, and this is similar for hand-held phones and hands-free devices. Young et al. [111] noted that at the time, about one fourth of all crashes (trucks and personal vehicles combined) were due at least in part to distractions, particularly mobile phones and navigational systems. They reviewed much of the literature available at the time of their writing. Wilson and Stimpson [112] reviewed trends in distracted driving accidents and noted that deaths due to distracted driving had increased 28% from 2005 to 2008 when the rate was nearly 6000 deaths per year.

Olson et al. [113] studied distracted driving in 203 commercial drivers. The data involved 4452 critical events, such as crashes, near-crashes, and unintentional lane departures, along with 19,888 time periods that involved no special events. They found that 71% of all crashes and 46% of near crashes involved drivers who were engaged in tasks not related to driving. Overall, 60% of critical events occurred while the driver was performing non-driving tasks. Klauer et al. [114] conducted a study in which 42 young drivers (16.3 to 17.0 years of age) who had just received their driver's license and 109 experienced drivers were studied. Here the unit of measurement is the driver. Equipment, such as accelerometers and cameras, were used to detect distracted motion while driving. They found that distracting events like eating or cell phone dialing or texting led to an increased risk of accident, with odds ratios often exceeding 3.0.

In terms of safety optimization, the choice here is clear. Distracted driving, such as hands-on cell phone use and texting, should not be allowed. From a general public perspective, these have translated into driving laws in many countries as well as have been translated into company policies for many commercial transportation firms. In addition, there are several smart phone-based applications that disables texting while driving and/or encourage safe driving behavior. From a commercial driving perspective, there are wearable technologies (e.g., headsets embedded with sensors that are linked to a smartphone application) that are used by professional drivers that provide voice-alerts when their mirror-check rate deviates from a pre-set standard. This information is also shared with dispatchers to schedule rest-breaks as an intervention. While these smart-phone applications/technologies seem promising, there is not a large body of literature that examines the effectiveness of these interventions.

4.1.3. Weather, Traffic Conditions, and Road Geometry

In Sections 4.1.1 and 4.1.2, we have discussed driver-related factors. In many cases, the crash likelihood and severity can be impacted by non-driver/external factors. Variables/features capturing weather (e.g., temperature, precipitation, wind speed, humidity, and visibility), traffic conditions (e.g., traffic flow, occupancy, density, and volume), and road geometry (e.g., elevation, curvature, road surface, and the number of lanes) represent the main external factors that impact the crash likelihood and severity [10,11,115]. Note that these factors should not be considered in isolation since their interactions are complex and can significantly change the crash likelihood. Thus, in this subsection, we highlight three relevant studies that have investigated the combined effect of such factors on crash risk.

Ahmed et al. [116] investigated the effect of the interaction between road geometric features, real-time weather parameters, and traffic data on crash likelihood. Using a Bayesian logistic regression framework, the authors developed two models for snowy and dry seasons. Based on their models

and case study, their results showed that in, both models, the main effects and at least one interaction term were significant. The authors showed that the crash risk during the snowy season was two times that of the dry season. Furthermore, the authors suggested that the crash risk likelihood may also be influenced by the interaction effects between the snowy, icy, or slushy road surface conditions with road segments involving steep grades.

In another study, Yu et al. [117] conducted their study on a 15-mile segment of the I-70 interstate in Colorado. The authors utilized: (a) 30 Remote Traffic Microwave Sensor (RTMS) sensors to extract real-time traffic data; (b) six weather stations for obtaining real-time weather data; and (c) the Roadway Characteristics Inventory (RCI) for obtaining descriptors of road geometry. Different scenarios were considered in the study based on the season and crash type. The results showed that the adverse weather condition combined with critical roadway conditions (e.g., steep slopes) can increase the crash likelihood significantly. Further, single vehicle (SV) and multiple vehicle (MV) models shared some common significant predictors such as precipitation and average speed. Furthermore, in the SV model, the significant variables were more related to weather conditions and vehicle speed. On the other hand, MV crashes were more affected by traffic-related variables.

Wang et al. [118] studied several of the factors that could lead to high risk traffic conditions. They considered traffic, weather, road geometry, and some behavioral aspects, such as trip generation and social demographics. These variables were taken as the characteristics of the region surrounding the crash, not the individuals involved in the crash. They used a case-control design with a 10:1 ratio of non-crashes to crashes. They used support vector machines (SVM) for variable selection and Bayesian logistic regression for inference. They found that the percentage of home-based work production, which includes commuters, was the only behavioral characteristic that had a significant effect on the risk of accident.

Xu et al. [115] developed crash prediction models at different levels of crash severity. Three levels of crash severity were considered: fatal/incapacitating injury crashes (KA), non-incapacitating/possible injury crashes (BC), and property-damage-only crashes (PDO). Results showed that under different crash severity levels, the effect of environmental variables is different. For example, in the all crashes model (KA, BC and PDO), adverse weather conditions would increase crash risk. However, under the injury crashes model (KA and BC), adverse weather conditions had the opposite effect which indicated that it could possibly reduce the likelihood that a crash would result in injuries and fatalities (possibly due to uncaptured changes in driver behaviors). Note also that the significant traffic-related variables are different in these two models which indicates that the interaction of the external variables would result in different level of crash risk and severity.

4.2. Statistical Modeling

Retrospective case-control studies are usually analyzed using logistic regression or other classification models. Since crashes are very rare compared with non-crashes, matching a case (crash) with one or more controls (non-crashes) is considered; this matching is then accounted for in the analysis. In other situations, non-crashes are unmatched; a set of controls is selected to mimic the aggregate of conditions of the crashes. Many studies are unclear about matching and whether (and also how) the matching was taken into account in the analysis. Since non-crashes are much more common than crashes, it is common to take several times as many non-crashes as crashes. Ratios as high as 10:1 are found.

Theofilatos and Yannis [10] surveyed previous research on the relationships between these factors and traffic crashes. Some of the commonalities in the conclusions were that safety was a nonlinear function of traffic flow, speed limits were a factor, and precipitation was related to accident frequency, although the effect on severity is unclear. Roshandel et al. [11] conducted a review and meta analysis of previous traffic safety studies and found that four variables are likely contributors to accident likelihood. These include speed variation around the crash site (odds ratio = 1.226), speed difference (odds ratio = 1.032), average traffic volume (odds ratio = 1.001) and average speed (odds ratio = 0.952).

Shi and Abdel-Aty [65] used a matched control design to study rear end crashes. They matched 243 crashes with 962 non-crashes, a ratio of about 4:1. They used a random forest for variable selection, and then used a Bayesian approach for logistic regression. They found that peak hour, high volume upstream (from the accident), low speed downstream, and high congestion index downstream were significant factors for rear end crashes. Pande and Abdel-Aty [119] also studied exclusively rear end crashes in an unmatched case-control study. They found 2179 rear-end crashes, but only 1620 with full data, in a period of five years and selected a random sample of 150,000 of the roughly 363 million possibilities for the controls. They used classification and regression trees (CART) to discriminate low and high risk situations. Their approach could classify a situation as high-risk about 75% of cases where there was an accident, with approximately a 33% positive rate. Since crashes were rare events, one can conclude that their false positive rate was $\approx 33\%$.

In a later study, Pande et al. [120] studied rear end crashes in a case-control study. They used a 5:1 ratio of non-crashes to crashes and used a random forest for variable selection and a multilevel perception neural network for inference. They found that occupancy downstream and average speed upstream were significant.

Theofilatos et al. [121] studied traffic safety on a multi-lane belt-line highway in Athens, Greece, where there were 17 crashes and 91,118 non-crashes. In one model they use all data, and in another model they use a random sample of the non-crashes. They assume a logistic regression model and in one model they use a penalized maximum likelihood approach, called the Firth method, which uses all of the data. In another approach, they use a bias correction method to estimate parameters in the logistic regression, and for this they use a subset of the data. They find that average speed has a negative effect on crashes. The proportion of trucks on the road was considered but not found to be significant.

Lin et al. [122] studied traffic safety on a corridor of Interstate 64 in Virginia, USA. Their study used a matched case-control design. They propose a frequent pattern (FP) tree which they use for variable selection. For inference on which variables are significant they use a k nearest neighbors algorithm and a Bayesian network. They conclude that the "accident risk prediction models based on FP tree variable selection outperform the models based on all variables ..." They also suggest using 10-minute intervals is more efficient than 5-minute intervals. Finally, they conclude that the Bayesian network model works well, yielding a false alarm rate of 0.38 and a sensitivity of 0.61.

Sun and Sun [123] used a matched case-control design with a ratio of 5:1 to implement a Markov model involving the traffic states upstream and downstream. For example, if one upstream and one downstream segment is considered, then an expressway segment may be in the state FF (free flow upstream and free flow downstream); this leads to a four-state Markov chain. They also consider two upstream and two downstream conditions, leading to a nine-state Markov chain. The transition probabilities were estimated using a dynamic Bayesian network model. Their model with nine states had a crash accuracy of 0.764 with a false alarm rate of 0.237. In addition to their work on the Bayesian network, they found an interesting nonlinear relationship between speed and risk, which they show in the second figure of their paper.

The effect of weaving, that is, traffic entering the expressway and merging while other traffic is exiting, was studied by Wang et al. [124] in a case-control study of 125 crashes and 1250 non-crashes, a 10:1 ratio. They applied a multilevel Bayesian logistic regression model with weaving segments (that is, sections of the expressway where entering and exiting traffic had to merge) as random effects. These random effects were incorporated into the model as random intercepts. They found that the speed at the beginning of the weaving segment, difference in speed between the beginning and end, and the log of traffic volume were significant effects in these weaving segments. Wang et al. [125] approached the traffic safety problem from two perspectives. One involved the crash frequency. This took as the sampling unit a section of the expressway and the number Y_i of accidents as the response. The other approach applied the usual logistic regression, taking the sampling unit as an expressway/time period slice and the indicator variable y_{ij} which is 1 for crash and 0 for a non-crash. The first approach leads

to Poisson regression and the second approach leads to the usual logistic regression. The innovative contribution of their method is to combine, or integrate, these two models. This effectively uses two sources of data. Their integrated model includes the Poisson rate in the logistic regression model, yielding a multi-level model. They find that the integrated model performs better yielding a higher receiver operating characteristic (ROC) curve.

There are a lot of aspects of crash prediction models that can be studied, including model setting, specification, and validation, but those are beyond the scope of this review. Details of statistical models can be found in previously published reviews by Lord and Mannering [126], Mannering and Bhat [127], Abdulhafedh et al. [128], Ambros et al. [129], Yannis et al. [130].

5. Conclusions

Given the tremendous loss of life and property directly attributed to motor vehicle incidents on one hand, and significant advances in relevant data availability on the other, it is natural that data analytics is viewed as having great potential for contributing to solving these problems. A successful effort in this direction necessarily has to rely on a combination of data collection, descriptive analytics, predictive/explanatory modeling, and optimization. At the same time, each piece separately can be a significantly nontrivial problem on its own. Hence, development of a mature data-driven decision support tool incorporating all of these stages “from scratch” is probably beyond the scope or ability of any single researcher. This is especially true since there is not a conscious effort in pulling all of these areas together with the goal of informing practical decision-making. The most significant gap that we have identified, is in the translation of outcomes/insights from predictive/explanatory models (which aim to help us better understand and quantify crash risk) into prescriptive optimization models (which aim to inform route/path selection, driver assignment, etc.). Perhaps, a partial underlying reason is the absence of readily available convenient data sources and/or data processing tools.

In this review, we highlighted a promising opportunity to develop advanced analytical methods for safety-enabled transportation. The following areas represent the main avenues for progress (ordered according to the sections in this review):

- (A) The availability of historical, real-time and forecasted weather and traffic data, as well as the potential to collect driver performance data, means that the accessibility of data is no longer a major factor preventing progress in this area. However, a lack of a unified repository and the reluctance of sharing code/models by our research community leads to a fairly high overhead cost of developing such models (since every researcher has to develop many data collection techniques from scratch);
- (B) Descriptive analytics tools are widely used in the preprocessing of driving-related data. Since the applicability of a particular preprocessing technique (e.g., visualization and clustering) often depends on the specific problem, the challenge here is to determine which method is the most suitable. Sharing best practices by creating reproducible documents (e.g., R Markdown and Jupyter notebook) represents one avenue for making the process more efficient for researchers and practitioners alike.
- (C) Statistical methods for risk evaluation are well-researched and consider a wide range of factors. At the same time, it must be noted that (in some cases) these studies follow a similar pattern of a case-controlled study based on a single road segment data. In our view, there is an opportunity for a statistical analysis of a larger scale since:
 - (i) real-time or near-real-time data are more widely available now;
 - (ii) the computational advancements in the recent years can allow for parallelizing/computing risk across the entire road network or at the very least for all major highways and interstates;
 - (iii) the insights from these relatively small road segments may not be generalizable to the entire road network; and

- (iv) it is unclear how drivers (regular commuters or commercial) can utilize these insights to make more informed decisions about their time-of-travel, path and/or route selection.

In our estimation, a more comprehensive/interdisciplinary approach to crash risk modeling is needed. The research questions should not be limited to only better understand the factors contributing to crash risk, but to also consider how the output from the research can be utilized by commuters and commercial drivers. This is especially important since, despite the technological advancements in sensing technologies and development of public policies that tackle distracted driving/cell phone usage, the rate and counts of motor vehicle injuries and fatalities have remained alarmingly high.

Supplementary Materials: In an effort to bridge the gap between the crash prediction literature and the hazmat/optimization literature, we have made all our source code used for (a) scraping crash-related data, (b) preprocessing of such-data; (c) descriptive analytics (i.e., visualizing traffic/weather/crash data and/or clustering); and (d) explanatory modeling available on a GitHub repository <https://github.com/caimiao0714/TrafficSafetyReviewRmarkdown>. To facilitate the consumption of this code, we host a website showcasing how the code can be used and depicting some of its results. The website was constructed using an R Markdown file [131], which is stored on the following GitHub Page <https://caimiao0714.github.io/TrafficSafetyReviewRmarkdown/>. We hope that the Supplementary Materials provided in this manuscript help promote “open data science” practices in our research community.

Funding: This work was supported in part by: the National Science Foundation (CMMI-1635927 and CMMI-1634992); the Ohio Supercomputer Center (PMIU0138 and PMIU0162); the American Society of Safety Professionals (ASSP) Foundation; the University of Cincinnati Education and Research Center Pilot Research Project Training Program; the Transportation Informatics Tier I University Transportation Center (TransInfo); a Google Cloud Platform research grant for data management; and a Dark Sky grant for extended API access (i.e., they increased the number of possible queries per day). Dr. Megahed’s research was also partially supported by the Neil R. Anderson Endowed Assistant Professorship at Miami University.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|-------|--|
| AADT | Annual Average Daily Traffic |
| DoT | Department of Transportation |
| FHWA | Federal Highway Administration |
| FMCSA | Federal Motor Carrier Safety Administration |
| NDS | Naturalistic Driving Study |
| NHTSA | National Highway Traffic Safety Administration |
| NOAA | National Oceanic & Atmospheric Administration |
| VT | Virginia Tech |

References

1. World Health Organization. WHO | The Top 10 Causes of Death. Available online: <http://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death> (accessed on 24 February 2019).
2. National Highway Traffic Safety Administration, NHTSA. U.S. DOT Announces 2017 Roadway Fatalities Down. Available online: <https://www.nhtsa.gov/press-releases/us-dot-announces-2017-roadway-fatalities-down> (accessed on 23 February 2019).
3. Insurance Institute for Highway Safety. Fatality Facts—IIHS. The Insurance Institute for Highway Safety and the Highway Loss Data Institute. Available online: <http://www.iihs.org/iihs/topics/t/general-statistics/fatalityfacts/overview-of-fatality-facts> (accessed on 24 February 2019).
4. World Health Organization. WHO | Road Traffic Injuries. Available online: <http://www.who.int/mediacentre/factsheets/fs358/en/> (accessed on 22 April 2018).
5. Blincoe, L.; Miller, T.R.; Zaloshnja, E.; Lawrence, B.A. The Economic and Societal Impact of Motor Vehicle Crashes, 2010 (Revised). U.S. Department of Transportation, National Highway Safety Administration, Report No.: DOT HS 812 013. Available online: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812013> (accessed on 28 April 2018).

6. GDP (Current US\$) | Data: United States. World Bank National Accounts Data, and OECD National Accounts Data Files. 2018. Available online: <https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?locations=US> (accessed on 28 April 2018).
7. Erkut, E.; Tjandra, S.A.; Verter, V. Hazardous materials transportation. *Handb. Oper. Res. Manag. Sci.* **2007**, *14*, 539–621.
8. Androutsopoulos, K.N.; Zografos, K.G. A bi-objective time-dependent vehicle routing and scheduling problem for hazardous materials distribution. *EURO J. Trans. Logist.* **2012**, *1*, 157–183. [[CrossRef](#)]
9. Abkowitz, M.; Cheng, P.D.M. Developing a risk/cost framework for routing truck movements of hazardous materials. *Accid. Anal. Prev.* **1988**, *20*, 39–51. [[CrossRef](#)]
10. Theofilatos, A.; Yannis, G. A review of the effect of traffic and weather characteristics on road safety. *Accid. Anal. Prev.* **2014**, *72*, 244–256. [[CrossRef](#)]
11. Roshandel, S.; Zheng, Z.; Washington, S. Impact of real-time traffic characteristics on freeway crash occurrence: Systematic review and meta-analysis. *Accid. Anal. Prev.* **2015**, *79*, 198–211. [[CrossRef](#)] [[PubMed](#)]
12. Aria, M.; Cuccurullo, C. bibliometrix: An R-tool for comprehensive science mapping analysis. *J. Informet.* **2017**, *11*, 959–975. [[CrossRef](#)]
13. Garfield, E.; Sher, I.H. KeyWords Plus™—Algorithmic derivative indexing. *J. Am. Soc. Inf. Sci.* **1993**, *44*, 298–299. [[CrossRef](#)]
14. Thiese, M.S.; Hanowski, R.J.; Kales, S.N.; Porter, R.J.; Moffitt, G.; Hu, N.; Hegmann, K.T. Multiple conditions increase preventable crash risks among truck drivers in a cohort study. *J. Occup. Environ. Med.* **2017**, *59*, 205. [[CrossRef](#)]
15. Newnam, S.; Xia, T.; Koppel, S.; Collie, A. Work-related injury and illness among older truck drivers in Australia: A population based, retrospective cohort study. *Saf. Sci.* **2019**, *112*, 189–195. [[CrossRef](#)]
16. Dingus, T.A.; Hanowski, R.J.; Klauer, S.G. Estimating crash risk. *Ergon. Des.* **2011**, *19*, 8–12. [[CrossRef](#)]
17. Guo, F. Statistical methods for naturalistic driving studies. *Annu. Rev. Stat. Appl.* **2019**, *6*, 309–328. [[CrossRef](#)]
18. Federal Highway Administration. Real-Time System Management. Department of Transportation. 2016. Available online: <https://ops.fhwa.dot.gov/511/index.htm> (accessed on 3 August 2018).
19. Guo, F.; Klauer, S.G.; Hankey, J.M.; Dingus, T.A. Near crashes as crash surrogate for naturalistic driving studies. *Transp. Res. Rec.* **2010**, *2147*, 66–74. [[CrossRef](#)]
20. Jansen, R.J.; Simone Wesseling, S. Harsh Braking by Truck Drivers: A Comparison of Thresholds and Driving Contexts Using Naturalistic Driving Data. In Proceedings of the 6th Humanist Conference, The Hague, The Netherlands, 13–14 June 2018.
21. Mollicone, D.; Kan, K.; Mott, C.; Bartels, R.; Bruneau, S.; van Wollen, M.; Sparrow, A.R.; Van Dongen, H.P. Predicting performance and safety based on driver fatigue. *Accid. Anal. Prev.* **2019**, *126*, 142–145. [[CrossRef](#)] [[PubMed](#)]
22. Zheng, L.; Ismail, K.; Meng, X. Traffic conflict techniques for road safety analysis: Open questions and some insights. *Can. J. Civ. Eng.* **2014**, *41*, 633–641. [[CrossRef](#)]
23. Johnsson, C.; Laureshyn, A.; De Ceunynck, T. In search of surrogate safety indicators for vulnerable road users: a review of surrogate safety indicators. *Transp. Rev.* **2018**, *38*, 765–785. [[CrossRef](#)]
24. Mahmud, S.S.; Ferreira, L.; Hoque, M.S.; Tavassoli, A. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS Res.* **2017**, *41*, 153–163. [[CrossRef](#)]
25. Knipling, R.R. Naturalistic Driving Events: No Harm, No Foul, No Validity. In Proceedings of the Eighth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Salt Lake City, UT, USA, 22–25 June 2015.
26. Knipling, R.R. Threats to Scientific Validity in Truck Driver Hours-of-Service Studies. In Proceedings of the Ninth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Manchester Village, VT, USA, 26–29 June 2017.
27. Guerrero-Ibáñez, J.; Zeadally, S.; Contreras-Castillo, J. Sensor technologies for intelligent transportation systems. *Sensors* **2018**, *18*, 1212. [[CrossRef](#)]
28. Abdelhamid, S.; Hassanein, H.S.; Takahara, G. Vehicle as a mobile sensor. *Proc. Comput. Sci.* **2014**, *34*, 286–295. [[CrossRef](#)]
29. Wikipedia Contributors. OpenStreetMap—Wikipedia, The Free Encyclopedia. 2019. Available online: <https://en.wikipedia.org/w/index.php?title=OpenStreetMap&oldid=900226891> (accessed on 5 June 2019).
30. Eugster, M.J.; Schlesinger, T. osmar: OpenStreetMap and R. *R J.* **2013**, *5*, 53–63. [[CrossRef](#)]

31. Washington, S.P.; Karlaftis, M.G.; Mannering, F. *Statistical and Econometric Methods for Transportation Data Analysis*; Chapman and Hall/CRC: London, UK, 2010.
32. Chen, W.; Guo, F.; Wang, F.Y. A survey of traffic data visualization. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2970–2984. [[CrossRef](#)]
33. Han, W.; Wang, J.; Shaw, S.L. Visual Exploratory Data Analysis of Traffic Volume. In *MICAI 2006: Advances in Artificial Intelligence*; Gelbukh, A., Reyes-Garcia, C.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 695–703.
34. Alam, I.; Ahmed, M.F.; Alam, M.; Ulisses, J.; Farid, D.M.; Shatabda, S.; Rossetti, R.J. Pattern mining from historical traffic big data. In Proceedings of the IEEE Region 10 Symposium (TENSYP), Cochin, India, 14–16 July 2017; pp. 1–5.
35. Nookala, L.S. Weather Impact on Traffic Conditions and Travel Time Prediction. Master’s Thesis, Department of Computer Science, University of Minnesota Duluth, Duluth, MN, USA, 2006. Available online: <https://www.semanticscholar.org/paper/Weather-Impact-on-Traffic-Conditions-and-Travel-Nookala/cc2d6345ee24c5383d5b25560f19856d862edd5e> (accessed on 4 September 2018).
36. Ferreira, N.; Poco, J.; Vo, H.T.; Freire, J.; Silva, C.T. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2149–2158. [[CrossRef](#)] [[PubMed](#)]
37. Guo, H.; Wang, Z.; Yu, B.; Zhao, H.; Yuan, X. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In Proceedings of the 2011 IEEE Pacific Visualization Symposium (PacificVis), Hong Kong, China, 1–4 March 2011; pp. 163–170.
38. Tsai, Y.T.; Alhwiti, T.; Swartz, S.M.; Megahed, F.M. Using visual data mining in highway traffic safety analysis and decision making. *J. Trans. Manag.* **2015**, *26*, 43–60. [[CrossRef](#)]
39. Pu, J.; Liu, S.; Ding, Y.; Qu, H.; Ni, L. T-Watcher: A new visual analytic system for effective traffic surveillance. In Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM), Milan, Italy, 3–6 June 2013; Volume 1, pp. 127–136.
40. NHTSA. FARS Speeding Data Visualization. United States Department of Transportation. Available online: <https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data> (accessed on 7 September 2018).
41. Xie, Z.; Yan, J. Kernel density estimation of traffic accidents in a network space. *Comput. Environ. Urban Syst.* **2008**, *32*, 396–406. [[CrossRef](#)]
42. Lovelace, R.; Nowosad, J.; Muenchow, J. *Geocomputation with R*; CRC Press: Boca Raton, FL, USA, 2019.
43. Kraak, M.J. Visualising spatial distributions. *Chapter 1999*, *11*, 157–173.
44. Erdogan, S. Explorative spatial analysis of traffic accident statistics and road mortality among the provinces of Turkey. *J. Saf. Res.* **2009**, *40*, 341–351. [[CrossRef](#)]
45. Wongsuphasawat, K.; Pack, M.; Filippova, D.; VanDaniker, M.; Olea, A. Visual analytics for transportation incident data sets. *Transp. Res. Rec. J. Transp. Res. Board* **2009**, *2138*, 135–145. [[CrossRef](#)]
46. Liu, S.; Pu, J.; Luo, Q.; Qu, H.; Ni, L.M.; Krishnan, R. VAIT: A visual analytics system for metropolitan transportation. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1586–1596. [[CrossRef](#)]
47. Zeng, W.; Fu, C.W.; Arisona, S.M.; Qu, H. Visualizing Interchange Patterns in Massive Movement Data. In *Proceedings of the 15th Eurographics Conference on Visualization (EuroVis '13)*; The Eurographs Association & #38; John Wiley & #38; Sons, Ltd.: Chichester, UK, 2013; pp. 271–280. doi:10.1111/cgf.12114. [[CrossRef](#)]
48. Kraak, M.J. The space-time cube revisited from a geovisualization perspective. In Proceedings of the 21st International Cartographic Conference. Durban, South Africa, 10–16 August 2003; pp. 1988–1996.
49. Kapler, T.; Wright, W. GeoTime information visualization. *Inf. Vis.* **2005**, *4*, 136–146. [[CrossRef](#)]
50. Romero, B. Traffic Accidents. 2015. Available online: <http://bretromero.com/traffic-accidents-cyclists/> (accessed on 9 September 2018).
51. Galka, M. Traffic Accidents. 2016. Available online: <http://metrocosm.com/map-us-traffic/> (accessed on 9 September 2018).
52. Tominski, C.; Schumann, H.; Andrienko, G.; Andrienko, N. Stacking-based visualization of trajectory attribute data. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 2565–2574. [[CrossRef](#)]
53. Pack, M.L.; Wongsuphasawat, K.; VanDaniker, M.; Filippova, D. ICE-visual analytics for transportation incident datasets. In Proceedings of the IEEE International Conference on Information Reuse & Integration (IRI’09), Las Vegas, NV, USA, 10–12 August 2009; pp. 200–205.

54. Cottrill, C.D.; Thakuria, P.V. Evaluating pedestrian crashes in areas with high low-income or minority populations. *Accid. Anal. Prev.* **2010**, *42*, 1718–1728. [[CrossRef](#)]
55. Pack, M.L. Visualization in transportation: Challenges and opportunities for everyone. *IEEE Comput. Graph. Appl.* **2010**, *30*, 90–96. [[CrossRef](#)] [[PubMed](#)]
56. Chu, D.; Sheets, D.A.; Zhao, Y.; Wu, Y.; Yang, J.; Zheng, M.; Chen, G. Visualizing hidden themes of taxi movement with semantic transformation. In Proceedings of the 2014 IEEE Pacific Visualization Symposium (PacificVis), Yokohama, Japa, 4–7 March 2014; pp. 137–144.
57. van Huysduynen, H.H.; Terken, J.; Martens, J.B.; Eggen, B. Measuring driving styles: A validation of the multidimensional driving style inventory. In Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Nottingham, UK, 22 September 2015; pp. 257–264.
58. Liu, H.; Taniguchi, T.; Tanaka, Y.; Takenaka, K.; Bando, T. Visualization of driving behavior based on hidden feature extraction by using deep learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2477–2489. [[CrossRef](#)]
59. Das, S.; Avelar, R.; Dixon, K.; Sun, X. Investigation on the wrong way driving crash patterns using multiple correspondence analysis. *Accid. Anal. Prev.* **2018**, *111*, 43–55. [[CrossRef](#)] [[PubMed](#)]
60. Croxford, B.; Penn, A.; Hillier, B. Spatial distribution of urban pollution: Civilizing urban traffic. *Sci. Total Environ.* **1996**, *189*, 3–9. [[CrossRef](#)]
61. Havre, S.; Hetzler, B.; Nowell, L. ThemeRiver: Visualizing theme changes over time. In Proceedings of the IEEE Symposium on Information Visualization 2000, Salt Lake City, UT, USA, USA, 9–10 October 2000; pp. 115–123.
62. Van Wijk, J.J.; Van Selow, E.R. Cluster and calendar based visualization of time series data. In Proceedings of the 1999 IEEE Symposium on Information Visualization (InfoVis'99), San Francisco, CA, USA, 24–29 October 1999; pp. 4–9.
63. Gudes, O.; Varhol, R.; Sun, Q.C.; Meuleners, L. Investigating articulated heavy-vehicle crashes in western Australia using a spatial approach. *Accid. Anal. Prev.* **2017**, *106*, 243–253. [[CrossRef](#)] [[PubMed](#)]
64. Sawalha, Z.; Sayed, T. Traffic accident modeling: Some statistical issues. *Can. J. Civ. Eng.* **2006**, *33*, 1115–1124. [[CrossRef](#)]
65. Shi, Q.; Abdel-Aty, M. Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 380–394. [[CrossRef](#)]
66. Hassan, H.M.; Abdel-Aty, M.A. Predicting reduced visibility related crashes on freeways using real-time traffic flow data. *J. Saf. Res.* **2013**, *45*, 29–36. [[CrossRef](#)]
67. Hossain, M.; Muromachi, Y. A real-time crash prediction model for the ramp vicinities of urban expressways. *IATSS Res.* **2013**, *37*, 68–79. [[CrossRef](#)]
68. Yu, R.; Abdel-Aty, M. Utilizing support vector machine in real-time crash risk evaluation. *Accid. Anal. Prev.* **2013**, *51*, 252–259. [[CrossRef](#)]
69. You, J.; Wang, J.; Guo, J. Real-time crash prediction on freeways using data mining and emerging techniques. *J. Mod. Transp.* **2017**, *25*, 116–123. [[CrossRef](#)]
70. Basso, F.; Basso, L.J.; Bravo, F.; Pezoa, R. Real-time crash prediction in an urban expressway using disaggregated data. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 202–219. [[CrossRef](#)]
71. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [[CrossRef](#)]
72. Saeys, Y.; Inza, I.; Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **2007**, *23*, 2507–2517. [[CrossRef](#)]
73. Goldberg, D.E.; Holland, J.H. Genetic algorithms and Machion Learning. *Mach. Learn.* **1988**, *3*, 95–99. [[CrossRef](#)]
74. Kennedy, R.; Eberhart, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks IV, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
75. Xu, C.; Wang, W.; Liu, P. A Genetic Programming Model for Real-Time Crash Prediction on Freeways. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 574–586. [[CrossRef](#)]
76. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
77. Jović, A.; Brkić, K.; Bogunović, N. A review of feature selection methods with applications. In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205.

78. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in Machine Learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
79. Nagendra, S.S.; Khare, M. Principal component analysis of urban traffic characteristics and meteorological data. *Transp. Res. Part D Transp. Environ.* **2003**, *8*, 285–297. [[CrossRef](#)]
80. Lee, H.C.; Cameron, D.; Lee, A.H. Assessing the driving performance of older adult drivers: On-road versus simulated driving. *Accid. Anal. Prev.* **2003**, *35*, 797–803. [[CrossRef](#)]
81. Li, Q.; Jianming, H.; Yi, Z. A flow volumes data compression approach for traffic network based on principal component analysis. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Seattle, WA, USA, 30 September–3 October 2007.
82. Caliendo, C.; Guida, M.; Parisi, A. A crash-prediction model for multilane roads. *Accid. Anal. Prev.* **2007**, *39*, 657–670. [[CrossRef](#)]
83. Guo, F.; Fang, Y. Individual driver risk assessment using naturalistic driving data. *Accid. Anal. Prev.* **2013**, *61*, 3–9. [[CrossRef](#)] [[PubMed](#)]
84. Lee, J.; Abdel-Aty, M.; Shah, I. Evaluation of surrogate measures for pedestrian trips at intersections and crash modeling. *Accid. Anal. Prev.* **2018**. [[CrossRef](#)] [[PubMed](#)]
85. Cook, R.D. Principal components, sufficient dimension reduction, and envelopes. *Annu. Rev. Stat. Appl.* **2018**, *5*, 533–559. [[CrossRef](#)]
86. Tipping, M.E.; Bishop, C.M. Probabilistic principal component analysis. *J. Royal Stat. Soc. Ser. B Stat. Methods* **1999**, *61*, 611–622. [[CrossRef](#)]
87. Schölkopf, B.; Smola, A.; Müller, K.R. *Kernel Principal Component Analysis*; Springer: Berlin/Heidelberg, Germany, 1997; pp. 583–588.
88. Berkhin, P. A survey of clustering data mining techniques. In *Grouping Multidimensional Data*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71.
89. Rai, P.; Singh, S. A survey of clustering techniques. *Int. J. Comput. Appl.* **2010**, *7*, 1–5. [[CrossRef](#)]
90. Fahad, A.; Alshatri, N.; Tari, Z.; Alamri, A.; Khalil, I.; Zomaya, A.Y.; Fofou, S.; Bouras, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 267–279. [[CrossRef](#)]
91. Hall, F.L.; Hurdle, V.; Banks, J.H. Synthesis of Recent Work on the Nature of Speed-Flow and Flow-Occupancy (Or Density) Relationships on Freeways. 1993. Available online: <https://trid.trb.org/view/1172887> (accessed on 10 August 2019).
92. Kerner, B.S.; Rehborn, H. Experimental properties of complexity in traffic flow. *Phys. Rev. E* **1996**, *53*, R4275. [[CrossRef](#)]
93. Wu, N. A new approach for modeling of Fundamental Diagrams. *Transp. Res. Part A Pol. Pract.* **2002**, *36*, 867–884. [[CrossRef](#)]
94. Golob, T.F.; Recker, W.W. A method for relating type of crash to traffic flow characteristics on urban freeways. *Transp. Res. Part A Pol. Pract.* **2004**, *38*, 53–80. [[CrossRef](#)]
95. Xu, C.; Liu, P.; Wang, W.; Li, Z. Evaluation of the impacts of traffic states on crash risks on freeways. *Accid. Anal. Prev.* **2012**, *47*, 162–171. [[CrossRef](#)]
96. Steenberghen, T.; Dufays, T.; Thomas, I.; Flahaut, B. Intra-urban location and clustering of road accidents using GIS: a Belgian example. *Inter. J. Geog. Inf. Sci.* **2004**, *18*, 169–181. [[CrossRef](#)]
97. Xie, Z.; Yan, J. Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: an integrated approach. *J. Transp. Geog.* **2013**, *31*, 64–71. [[CrossRef](#)]
98. Shen, L.; Lu, J.; Long, M.; Chen, T. Identification of Accident Blackspots on Rural Roads Using Grid Clustering and Principal Component Clustering. *Math. Prob. Eng.* **2019**, *2019*. [[CrossRef](#)]
99. Kwon, O.H.; Park, S.H. Identification of Influential Weather Factors on Traffic Safety Using K-means Clustering and Random Forest. In *Advanced Multimedia and Ubiquitous Engineering*; Springer: Singapore, 2016; pp. 593–599.
100. Crum, M.R.; Morrow, P.C.; Olsgard, P.; Roke, P.J. Truck driving environments and their influence on driver fatigue and crash rates. *Transp. Res. Rec.* **2001**, *1779*, 125–133. [[CrossRef](#)]
101. Crum, M.R.; Morrow, P.C. The influence of carrier scheduling practices on truck driver fatigue. *Transp. J.* **2002**, 20–41.

102. Garbarino, S.; Durando, P.; Guglielmi, O.; Dini, G.; Bersi, F.; Fornarino, S.; Toletone, A.; Chiorri, C.; Magnavita, N. Sleep apnea, sleep debt and daytime sleepiness are independently associated with road accidents. A cross-sectional study on truck drivers. *PLoS ONE* **2016**, *11*, e0166262. [[CrossRef](#)]
103. Dingus, T.A.; Klauer, S.G.; Neale, V.L.; Petersen, A.; Lee, S.E.; Sudweeks, J.; Perez, M.A.; Hankey, J.; Ramsey, D.; Gupta, S.; et al. *The 100-Car Naturalistic Driving Study. Phase 2: Results of the 100-Car Field Experiment*; Technical Report; United States Department of Transportation, National Highway Traffic Safety: Washington, DC, USA, 2006.
104. McCauley, P.; Kalachev, L.V.; Smith, A.D.; Belenky, G.; Dinges, D.F.; Van Dongen, H.P. A new mathematical model for the homeostatic effects of sleep loss on neurobehavioral performance. *J. Theor. Biol.* **2009**, *256*, 227–239. [[CrossRef](#)]
105. McCauley, P.; Kalachev, L.V.; Mollicone, D.J.; Banks, S.; Dinges, D.F.; Van Dongen, H.P. Dynamic circadian modulation in a biomathematical model for the effects of sleep and sleep loss on waking neurobehavioral performance. *Sleep* **2013**, *36*, 1987–1997. [[CrossRef](#)]
106. Stern, H.S.; Blower, D.; Cohen, M.L.; Czeisler, C.A.; Dinges, D.F.; Greenhouse, J.B.; Guo, F.; Hanowski, R.J.; Hartenbaum, N.P.; Krueger, G.P.; et al. Data and methods for studying commercial motor vehicle driver fatigue, highway safety and long-term driver health. *Accid. Anal. Prev.* **2019**, *126*, 37–42. [[CrossRef](#)]
107. Bowden, Z.E.; Ragsdale, C.T. The truck driver scheduling problem with fatigue monitoring. *Decis. Sup. Syst.* **2018**, *110*, 20–31. [[CrossRef](#)]
108. Åkerstedt, T.; Folkard, S. Validation of the S and C components of the three-process model of alertness regulation. *Sleep* **1995**, *18*, 1–6. [[CrossRef](#)] [[PubMed](#)]
109. Åkerstedt, T.; Folkard, S.; Portin, C. Predictions from the three-process model of alertness. *Aviat. Space Environ. Med.* **2004**, *75*, A75–A83. [[PubMed](#)]
110. World Health Organization. *Mobile Phone Use: A Growing Problem of Driver Distraction*; World Health Organization: Geneva, Switzerland, 2011.
111. Young, K.; Regan, M.; Hammer, M. Driver distraction: A review of the literature. *Dist. Driv.* **2007**, *2007*, 379–405.
112. Wilson, F.A.; Stimpson, J.P. Trends in fatalities from distracted driving in the United States, 1999 to 2008. *Am. J. Publ. Health* **2010**, *100*, 2213–2219. [[CrossRef](#)] [[PubMed](#)]
113. Olson, R.L.; Hanowski, R.J.; Hickman, J.S.; Bocanegra, J. *Driver Distraction in Commercial Vehicle Operations*; Technical Report; United States Federal Motor Carrier Safety Administration: Washington, DC, USA, 2009.
114. Klauer, S.G.; Guo, F.; Simons-Morton, B.G.; Ouimet, M.C.; Lee, S.E.; Dingus, T.A. Distracted driving and risk of road crashes among novice and experienced drivers. *N. Engl. J. Med.* **2014**, *370*, 54–59. [[CrossRef](#)]
115. Xu, C.; Tarko, A.P.; Wang, W.; Liu, P. Predicting crash likelihood and severity on freeways with real-time loop detector data. *Accid. Anal. Prev.* **2013**, *57*, 30–39. [[CrossRef](#)]
116. Ahmed, M.; Abdel-Aty, M.; Yu, R. Assessment of interaction of crash occurrence, mountainous freeway geometry, real-time weather, and traffic data. *Transp. Res. Rec. J. Transp. Res. Board* **2012**, *2280*, 51–59. [[CrossRef](#)]
117. Yu, R.; Abdel-Aty, M.; Ahmed, M. Bayesian random effect models incorporating real-time weather and traffic data to investigate mountainous freeway hazardous factors. *Accid. Anal. Prev.* **2013**, *50*, 371–376. [[CrossRef](#)]
118. Wang, L.; Abdel-Aty, M.; Lee, J.; Shi, Q. Analysis of real-time crash risk for expressway ramps using traffic, geometric, trip generation, and socio-demographic predictors. *Accid. Anal. Prev.* **2019**, *122*, 378–384. [[CrossRef](#)]
119. Pande, A.; Abdel-Aty, M. Comprehensive analysis of the relationship between real-time traffic surveillance data and rear-end crashes on freeways. *Transp. Res. Rec.* **2006**, *1953*, 31–40. [[CrossRef](#)]
120. Pande, A.; Das, A.; Abdel-Aty, M.; Hassan, H. Estimation of real-time crash risk: Are all freeways created equal? *Transp. Res. Rec. Transp. Res. Rec. J. Transp. Res. Board* **2011**, *2237*, 60–66. [[CrossRef](#)]
121. Theofilatos, A.; Yannis, G.; Kopelias, P.; Papadimitriou, F. Impact of real-time traffic characteristics on crash occurrence: Preliminary results of the case of rare events. *Accid. Anal. Prev.* **2018**, *130*, 151–159. [[CrossRef](#)] [[PubMed](#)]
122. Lin, L.; Wang, Q.; Sadek, A.W. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transp. Res. Part C Emerg. Technol.* **2015**, *55*, 444–459. [[CrossRef](#)]

123. Sun, J.; Sun, J. A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 176–186. [[CrossRef](#)]
124. Wang, L.; Abdel-Aty, M.; Shi, Q.; Park, J. Real-time crash prediction for expressway weaving segments. *Transp. Res. Part C Emerg. Technol.* **2015**, *61*, 1–10. [[CrossRef](#)]
125. Wang, L.; Abdel-Aty, M.; Lee, J. Safety analytics for integrating crash frequency and real-time risk modeling for expressways. *Accid. Anal. Prev.* **2017**, *104*, 58–64. [[CrossRef](#)]
126. Lord, D.; Mannering, F. The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. *Transp. Res. Part A Pol. Pract.* **2010**, *44*, 291–305. [[CrossRef](#)]
127. Mannering, F.L.; Bhat, C.R. Analytic methods in accident research: Methodological frontier and future directions. *Anal. Meth. Accid. Res.* **2014**, *1*, 1–22. [[CrossRef](#)]
128. Abdulhafedh, A. Road crash prediction models: Different statistical modeling approaches. *J. Transp. Technol.* **2017**, *7*, 190–205. [[CrossRef](#)]
129. Ambros, J.; Jurewicz, C.; Turner, S.; Kieć, M. An international review of challenges and opportunities in development and use of crash prediction models. *Eur. Transp. Res. Rev.* **2018**, *10*, 35. [[CrossRef](#)]
130. Yannis, G.; Dragomanovits, A.; Laiou, A.; Richter, T.; Ruhl, S.; La Torre, F.; Domenichini, L.; Graham, D.; Karathodorou, N.; Li, H. Use of accident prediction models in road safety management—an international inquiry. *Transp. Res. Procedia* **2016**, *14*, 4257–4266. [[CrossRef](#)]
131. Xie, Y.; Allaire, J.J.; Grolemond, G. *R Markdown: The Definitive Guide*; CRC Press: Boca Raton, FL, USA, 2018.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

Sensors Editorial Office
E-mail: sensors@mdpi.com
www.mdpi.com/journal/sensors



MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34
Fax: +41 61 302 89 18

www.mdpi.com



ISBN 978-3-0365-0849-8