algorithms

Special Issue Reprint

Algorithms for Smart Cities

Edited by
Gloria Cerasela Crişan and Elena Nechita

mdpi.com/journal/algorithms

MDPI

Algorithms for Smart Cities

Algorithms for Smart Cities

Editors

Gloria Cerasela Crişan
Elena Nechita

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Editors

Gloria Cerasela Crişan
Vasile Alecsandri University
of Bacău
Bacău
Romania

Elena Nechita
Vasile Alecsandri University
of Bacău
Bacău
Romania

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Algorithms* (ISSN 1999-4893) (available at: https://www.mdpi.com/journal/algorithms/special_issues/Algorithms_Smart_Cities).

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

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-2009-2 (Hbk)
ISBN 978-3-7258-2010-8 (PDF)
doi.org/10.3390/books978-3-7258-2010-8

Contents

About the Editors

Gloria Cerasela Crişan

Gloria Cerasela Crişan graduated with a degree in Informatics from the University of Bucharest, Romania, in 1986. She holds a M.Sc. degree in Computer Science from "Dunărea de Jos" University of Galaţi, Romania, in 2003, and a PhD in Informatics from "Alexandru Ioan Cuza" University of Iaşi, Romania, in 2008. She is an Associate Professor and currently the Dean of the Faculty of Sciences at "Vasile Alecsandri" University of Bacău, Romania. Additionally, she is a PhD coordinator at the Faculty of Informatics at "Alexandru Ioan Cuza" University of Iaşi, Romania. To date, she has contributed to over 60 scientific publications, 48 of which are indexed on the Web of Science platform. Her primary research interests involve solving Combinatorial Optimization Problems, particularly with applications in Transportation and Logistics.

Elena Nechita

Elena Nechita graduated from "Alexandru Ioan Cuza" University of Iaşi, Romania, in 1987 and holds a PhD in Informatics from the same institution, in 2000. She currently works at "Vasile Alecsandri" University of Bacău, Romania, in the Department of Mathematics and Informatics, and she also handles internationalization duties. She teaches courses on AI and risk management in ICT projects. Her research interests include combinatorial optimization, ICT for sustainable development, and Computer Science Education, and she has led several initiatives related to the latter subject. With over 20 years of research experience in European and national projects, she has published papers in conferences and journals and has served as a reviewer for international conferences and journals in Computer Science.

algorithms

MDPI

Article

Optimal Model for Carsharing Station Location Based on Multi-Factor Constraints

Qiuyue Sai [1,*], Jun Bi [2] and Jinxian Chai [3]

1 School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
2 Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing 100044, China; jbi@bjtu.edu.cn
3 The Experimental High School Attached to Beijing Normal University, Beijing 100032, China; jasonchaijinxian@163.com
* Correspondence: 19114026@bjtu.edu.cn; Tel.: +86-15727306200

Received: 10 January 2020; Accepted: 13 February 2020; Published: 18 February 2020

Abstract: The development of the sharing economy has made carsharing the main future development model of car rental. Carsharing network investment is enormous, but the resource allocation is limited. Therefore, the reasonable location of the carsharing station is important to the development of carsharing companies. On the basis of the current status of carsharing development, this research considers multiple influencing factors of carsharing to meet the maximum user demand. Meanwhile, the constraint of the limited cost of the company is considered to establish a nonlinear integer programming model for station location of carsharing. A genetic algorithm is designed to solve the problem by analyzing the location model of the carsharing network. Finally, the results of a case study of Lanzhou, China show the effectiveness of the establishment and solution of the station location model.

Keywords: carsharing; station location modeling; genetic algorithm; fix stations and free stations

1. Introduction

Carsharing is gaining popularity due to its green advantages. Carsharing helps reduce the number of private cars, which alleviates traffic congestion in turn [1]. Carsharing refers to the leasing business in which cars are owned by a carsharing company and used by different users at varying times with measurement by duration and mileage. Unlike traditional offline car rental service and day-by-day billing units, carsharing users register and authenticate on their mobile phones, use applications to find surrounding stations, and rent cars by time [2]. This research analyzes electric carsharing, which is a carsharing that is powered by electricity and has less pollution impact than traditional cars.

This research focuses on the types of electric carsharing [3,4] in which the cars can be fetched and parked at fix or free stations. Fix stations require a certain area to store cars, and the costs include land rent, charger and station construction costs [5,6]. If the area does not have fix stations, then users can park their cars on the public parking space in a certain area. Different fees of users are charged for parking at varying stations. However, the parking fee is paid by the company. Therefore, for the carsharing company, the free station does not require the cost of station construction, but parking costs will be incurred. The construction of fixed stations is conducive to the management of cars, and free stations are convenient for users to use cars and for reducing the cost of station construction. The combination of two operation forms is conducive to the sustainable development of carsharing.

Carsharing is advantageous because it provides great convenience for users and improves the flexibility of cars [7]. Zhu [8] analyzed the objective conditions of the new energy carsharing in China and revealed the popularization value and prospects of carsharing in China. Feng [9] analyzed the

current development status and future scale of the major car long-term rental, short-term rental, and online car rental markets. These studies are biased toward policy and market analysis. Notably, carsharing has a great future in the transportation field. However, the development of the carsharing industry is still in its early stages. The problems of few stations, few available cars, and few chargers have become key restricting factors of the development of carsharing companies [10]. Some studies have found that carsharing network settings can affect user willingness and company development. Ciari [11] used a binary logistic model and showed that the location of stations actually affects potential membership. This research used elastic analysis to find the relationship between distance and number of users but did not introduce specific methods for station location optimization. Correia [12] found that financial losses can be reduced through appropriate choices with respect to the number, location, and size of the depots. This research provided the foundation for the necessity of the station location optimization model.

Scholars have conducted the following research to address the problems of carsharing station location optimization. Jiang [13] used analytic hierarchy process to calculate the best scheme for carsharing stations. The study identified variables with a significant effect on station location selection but did not build an optimization model for carsharing station location determination. Lu [14] used the interval fuzzy soft set method of risk preference for each carsharing station in Wang Cheng County. The evaluation of the plan has certain reference significance for the location and future planning of carsharing. This method is more suitable for evaluating existing carsharing stations than for planning for new cities. The above-mentioned research methods are relatively subjective. Other methods using mathematical models are presented as follows. Çalık [15] illustrated a carsharing locating recharging station model that operates under demand uncertainty. The research developed a demand forecasting method that allows the generation of many demand scenarios. Hu [16] formulated a mixed queuing network model for the joint design of fleet size and station capacities. The optimization problem was solved by the genetic algorithm. They proved that the profit is maximized when the existing road congestion is moderate. This research mainly considered the carsharing model of the fixed station but ignored the setting of free stations. Wielinski, G [17] researched the travel area and behavior of free-floating carsharing users. This study mainly considered the operation model of free-floating carsharing but ignored the setting of fixed stations. Zheng [18] proposed a method for optimizing the location of charging stations for one-way electric carsharing systems. The objective function was to maximize the profit of carsharing service. Simulations were performed to prove the effectiveness of the research method. The final station was optimized by 0–1 variable station selection among existing candidate sites. However, the study failed to provide scientific theoretical basis for the selection of the alternative station. Lee [19] analyzed eight spatial elements related to carsharing location. A model was established to determine the optimal carsharing locations with the minimum total distance between carsharing users. This model took the minimum travel cost of the users as the optimization goal, which optimized the station location based on fixed station operations. Chen [20] designed a genetic algorithm to solve the problem with the least total change and the smallest gap between supply and demand. The model results obtained the adjustment of the number of cars at each station after optimization. This model is more suitable for the scale expansion of a city that already has carsharing than for the location optimization of a new city. In this research, the electric carsharing location and car allocation model of fixed and free stations are considered first. This way improves not only the efficiency of electric carsharing management but also the convenience of user demands. Second, this study uses the grid division method for area discretization, which improves the shortcomings of random optimization of alternative stations based on alternative station optimization. To the best of the authors' knowledge, this study is the first to integrate the above-mentioned method to carsharing location method.

Electric carsharing is an asset-heavy and high-risk industry. Networking is the development trend. The electric carsharing station network is not only crucial to the company's profitability but also significant for the development of urban transportation systems [21]. First, electric carsharing network is difficult to alter after completion due to its large investment. The geographical location of

the network directly determines the network location and economic benefits of the company. Second, the current charging, parking space, and car license plate resources in the industry are limited. The location of the stations determines whether these resources can be fully utilized for reducing costs [22].

This research analyzes the optimization model of the location of electric carsharing networks. The optimization model and application scenarios of electric carsharing are combined to determine the affecting factors of electric carsharing network station location. The objective function is to meet the maximum user demand with the limited cost of the company as the constraint while considering factors, such as land rent, parking prices, and the number of chargers. An optimization model for the location of electric carsharing is established. Then, the location plan of the network in the research area is obtained using genetic algorithm by analyzing the established nonlinear integer programming model. Finally, a case study is conducted. The results indicate that this research can improve the coverage ratio of user demand and the convenience of users. Accordingly, the competitiveness of the electric carsharing industry development can be enhanced. On the one hand, this research is important in promoting healthy development of the industry. On the other hand, this research can provide theoretical guidance for the station location of electric carsharing companies.

The remainder of the paper is organized as follows. In Section 2, the data process is introduced. Section 3 presents the electric carsharing location model and the solution of genetic algorithm in detail. Section 4 demonstrates the case study and solution result. Conclusions and prospects are elaborated in Section 5.

2. Data Preprocessing

2.1. Region Discretization

In this research, the area is discretized [23]. As shown in Figure 1, the area is divided into a grid with m rows by n columns. Each small grid is a demand point and a candidate station. The number and size of small grids can be determined on the basis of the development status and development plan of the study area.

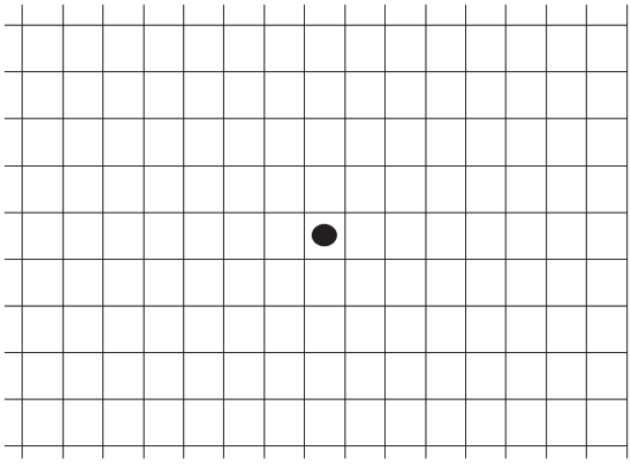


Figure 1. Schematic of the study area meshing.

2.2. Definitions of Variables

2.2.1. Regional Demand Forecast

The amount of user demand in the area as needed is determined. First, the total demand for electric carsharing in the region can be obtained through the macro demand forecast. Then, according

to the population density, land use, and other parameters of each small grid, the total demand for regional electric carsharing is allocated to each small area, and the demand matrix of the entire area can be obtained [24]. In accordance with Equation (1), the total demand of electric carsharing in the area is obtained on the basis of the urban resident population in consideration of the average daily trip volume, travel mode rate, and electric carsharing travel demand. We define the total number of carsharing in a region that is proportional to the user's total carsharing distance divided by the daily carsharing travelable distance.

$$M = \frac{\omega \sum_{i=0}^{r} apd}{gv}, \tag{1}$$

where M is the total demand number of electric carsharing. r is the total number of residents. i is the resident. a is the daily number of trips per capita, (trips/day). p is the share of electric carsharing that can be acquired by survey data. d is the average distance traveled by the electric carsharing, (meter). g is the daily travelable duration of the electric carsharing,(minutes). v is the average travel speed of the electric carsharing, (meter/minutes). ω is the adjustment coefficient, which is related to car redistribution efficiency and car utilization efficiency.

After the total demand number of electric carsharing is obtained, the demand for electric carsharing in each small region is calculated. Combined with factors, such as electric carsharing application scenarios and pedestrian flow, this research considers seven indicators, namely, residential population density, population of the campus, job numbers, leisure entertainment venues, tourist attractions, transportation hubs, and ordinary interchange point demand for electric carsharing, to measure the effect of each small area. The demand for electric carsharing C_{uv} in each small grid is calculated as follows.

$$C_{uv} = c^1_{uv} + 0.5c^2_{uv} + c^3_{uv} + c^4_{uv} + c^5_{uv} + c^6_{uv} + c^7_{uv}, \tag{2}$$

where v is the v-th column and u is the u-th row. We consider several factors that may affect the demand for carsharing in an area. These factors are chosen due to the following reasons. c^1_{uv} is the daily human traffic of residential users and represents the number of potential users and the amount of trips to a certain extent (people volume/day). c^2_{uv} is the daily human traffic of college students who have a high percentage of driving licenses, limited purchasing power, and strong ability to accept new things. Meanwhile, students need to go to the city center for shopping and entertainment because most college campuses are in suburbs. Thus, college students are the important users of electric carsharing (people volume/day). c^3_{uv} is the daily human traffic for jobs. Job places are not only the starting and ending points of commute but also an important node for people traveling during the day. High-density jobs will generate high travel demands (people volume/day). c^4_{uv} is the daily human traffic of the leisure entertainment places. Electric carsharing is often suitable for the purposes of going to the places of leisure entertainment places at night, weekends, and holidays (people volume/day). c^5_{uv} is the daily human traffic of tourist attractions that often attract a large number of tourists from the city. The convenience and privacy of electric carsharing meet their travel needs to stations, hotels, and attractions (people volume/day). c^6_{uv} is the daily human traffic of large-scale transportation hub, which mainly refers to passenger stations, high-speed rail stations, ordinary railway stations, and airports in the city. These places are distribution centers for a large number of people and accommodate a certain number of customer groups (people volume/day). c^7_{uv} is the daily human traffic of ordinary interchange point, which mainly refers to the subway and bus interchange stations in the city. Although the short-term flow of people at these interchange points is not as large as that at a transportation hub, these interchange points are more numerous in cities and function as important connection points among different modes of transportation (people volume/day). The weight of each factor is set according to the average of survey data. Although college students have a high intention to use carsharing, economic conditions restrict the use frequency for carsharing. Therefore, the weight of c^2_{uv} is set to 0.5. Other factors are set to 1, which means other kind of human has more influence than c^2_{uv}.

According to the ratio of the demand of each small grid to the total demand of the area, the total cars in the area is distributed to each small grid, and the demand number of electric carsharing R_{uv} in grids of row u and column v is obtained, as follows. C_{uv} is the demand for electric carsharing in grids of row u and column v. $\sum_{u=1}^{m}\sum_{v=1}^{n} C_{uv}$ is the total demand for electric carsharing in this area. We allocate total cars according to the proportion of demand in each grid.

$$R_{uv} = \frac{C_{uv}}{\sum_{u=1}^{m}\sum_{v=1}^{n} C_{uv}} * M, \tag{3}$$

where R is the number of electric carsharing demand matrix. R_{uv} is the demand number of electric carsharing in grids of row u and column v.

2.2.2. Regional Service Matrix

The service capacity and scope of a single electric carsharing station are limited. If the user arrives at an electric carsharing station at a long distance, then the user's willingness to rent will be greatly decreased. This phenomenon can be quantified using business circle theory. According to the business circle theory in economics, the mall or business district within a certain economic area is the center to expand in a certain direction and distance, thus forms a certain range or area that attracts customers. Commercial districts are generally divided into core, secondary, and marginal commercial districts. In this study, the willingness to use the carsharing decreases with distance between the station and the user, which also means the capacity of each station's service decreases with distance. Placing carsharing in a grid will attract users in nearby grids. Therefore, not 100% of the cars in this grid will serve users in this grid. Some cars put in this grid will be used by users in nearby grids. So, carsharing in a locale will also provide services to users in nearby grids. The grid with electric carsharing network is assumed as the core district. The number of cars that can serve this point accounts for β_1 of the total number of users. The secondary district accounts for β_2. The marginal district accounts for β_3. If the station set is at row u and column v, then its service matrix is A_{uv}.

$$A_{uv} = \begin{bmatrix} \beta_3 & \beta_2 & \beta_3 \\ \beta_2 & \beta_1 & \beta_2 \\ \beta_3 & \beta_2 & \beta_3 \end{bmatrix}$$

where β_1 is the service capacity of the core districts. β_2 is the service capacity of the secondary districts. β_3 is the service capacity of the marginal districts.

$L(u, v)$ is a matrix that indicates the distribution of the network stations to be created, where l_{uv}= 1 indicates that a network point is to be created at that grid, and l_{uv}= 0 indicates that no station is to be created at this grid. $L(u, v)$ is a 0–1 matrix with m rows and n columns. The u-th row and v-th column of this matrix are equal to 1, and others are equal to 0. Then, the carsharing station of the distribution matrix in the entire area $\boldsymbol{L}$ is calculated as follows.

$$L = \sum L(u, v), \tag{4}$$

where $L(u, v)$ is the distribution of the network stations to be created in row u and column v. $\boldsymbol{L}$ is the distribution matrix of carsharing station.

The service provided by a single station in row u and column v, P'_{uv} can be calculated as follows.

$$P'_{uv} = L(u, v) \otimes (\mathrm{A}_{uv} B_{uv}), \tag{5}$$

where B_{uv} is the number of cars to be invested at the stations in row u and column v. $\otimes$ represents two-dimensional discrete volume integrals [25]. The service matrix provided by a single station in

the entire area can be directly obtained through the calculation of two-dimensional discrete volume integrals. Therefore, the sum of the services provided by the entire station is calculated as follows:

$$P = \sum_{u=1}^{m}\sum_{v=1}^{n} P_{uv} = \sum_{u=1}^{m}\sum_{v=1}^{n}\sum P'_{uv}, \tag{6}$$

3. Electric Carsharing Location Optimization Model

3.1. Assumptions

We make several assumptions to construct the electric carsharing location optimization model. The specific content is listed as follows:

1. The influence of electric carsharing stations on user demand in the region is ignored.
2. We mainly focus on one-way carsharing system. The number of cars in each station remains stable.
3. The maintenance costs of cars in free parking areas are ignored.

3.2. Problem Setting

This study aims to determine the location and scale of electric carsharing under the condition of limited resources for maximizing the number of users. First, the research area is gridded. Second, the demands within each grid are calculated. Finally, the optimal network scheme is obtained using the mathematical model.

3.3. Mathematical Model

The model is mainly divided into two parts: objective function and cost constraint. The objective function is obtained by comparing the user demand matrix with the service matrix. The demand and service matrixes are obtained by discretizing the regions. Constraints mainly consider cost restrictions. Costs include land rent, car purchase fees, charging construction fees, and parking fees. This section discusses the electric carsharing station location model in detail.

3.3.1. Objective Function

Many factors should be considered in the station location of electric carsharing. The most important factor is user demand. The more user demands that are satisfied, the larger the economic benefits. This factor is the foundation and value of existence of the electric carsharing company. Therefore, the objective function is to maximize the user demands that are met in accordance with Equation (7).

$$\max S = \sum_{u=1}^{m}\sum_{v=1}^{n} \min(P_{uv}, R_{uv}), \tag{7}$$

where P_{uv} is the service provided by the stations in row u and column v. R_{uv} is the demand of the grid in row u and column v. S is the total number of user demands that can be met.

3.3.2. Constraint Condition

(1) Car purchase cost

We assume that the purchase cost of cars is linearly related to their number. Then, the car purchase cost can be calculated as follows:

$$F_{car} = \sum_{u=1}^{m}\sum_{v=1}^{n} b * B_{uv}, \tag{8}$$

where B_{uv} is the car purchase plan and denotes the number of cars to be invested at the stations in row u and column v. b is the cost of each car.

(2) Charger cost

At present, the chargers located in cities cannot meet the charging demands of all electric vehicles, and electric carsharing will have demands for charging during the operation process. Thus, electric carsharing operators need to build a certain number of chargers. The construction costs for chargers can be calculated as follows:

$$F_{charging} = \lambda * e \sum_{u=1}^{m} \sum_{v=1}^{n} B_{uv}, \tag{9}$$

where e is the construction cost of each charger and λ is the corresponding coefficient between the required number of stations and charging. λ is generally less than 1.

(3) Parking fee cost

At free stations, parking fees are incurred after cars are used. The parking fee is generally related to the parking price, parking time, and the number of cars in the station. The parking costs for one year can be calculated as follows:

$$F_{parking} = \sum_{u=1}^{m} \sum_{v=1}^{n} J_{uv} * t * K_{uv}, \tag{10}$$

where K_{uv} is the locations of free stations. J_{uv} is the average parking price for 1 h in row u and column v. t is the average parking time of each car in a year.

(4) Land rent cost

The rent of each fixed station is related to the average local land price and the area of the station. The car number in the station determines the size of the station. The land rent can be calculated as follows:

$$F_{rent} = \sum_{u=1}^{m} \sum_{v=1}^{n} \mu * V_{uv} * (B_{uv} - K_{uv}) \tag{11}$$

where V_{uv} is the average rent per square meter and per year in row u and column v. μ is the correspondence coefficient between size and location of the station. B_{uv} is the delivery plan for cars in the entire area. K_{uv} is the delivery plan for cars at free stations.

The total cost F is equal to the accumulation of various costs, as shown in Equation (12).

$$\begin{aligned} F &= F_{car} + F_{rent} + F_{charging} + F_{parking} \\ &= \sum_{u=1}^{m} \sum_{v=1}^{n} b * B_{uv} + \sum_{u=1}^{m} \sum_{v=1}^{n} V_{uv} * \mu * (B_{uv} - K_{uv}) + \sum_{u=1}^{m} \sum_{v=1}^{n} \lambda * \mathrm{e} * B_{uv} + \sum_{u=1}^{m} \sum_{v=1}^{n} J_{uv} * t * K_{uv} \\ &= \sum_{u=1}^{m} \sum_{v=1}^{n} B_{uv} l_{uv} (b + \mu * V_{uv} + \lambda * e) + \sum_{u=1}^{m} \sum_{v=1}^{n} K_{uv} * (J_{uv} * l_{uv} * t - V_{uv} * \mu) \end{aligned} \tag{12}$$

(5) Budget constraint

We suppose that the maximum investment of these companies in these costs is W. Thus, the cost constraint is Equation (13).

$$F \leq W. \tag{13}$$

(6) Other constraints

Constraints on the size of stations are given by Equation (14). From practical considerations, the scale of stations should not be too large or too small. Thus, the number of cars put in is limited.

$$B_{uv} \in [num1, num2], u \in [1, m], v \in [1, n] \tag{14}$$

Constraints on the total number of stations are shown below. The total number of stations to be laid out in the solved layout plan should not exceed the total number of alternative stations. According to Equation (15), N is the total number of proposed network points.

$$1 < N < mn \tag{15}$$

3.4. Optimization Algorithm

Genetic algorithm was proposed by John Holland in 1975 as an intelligent bionic algorithm that mimics the evolutionary process of living things [26]. This algorithm is a heuristic algorithm. This algorithm borrows natural phenomena, such as the initial population generation, natural selection, gene crossing, and mutation during the evolution of organisms, and iteratively generates new offspring. [27] The fitness function is set by simulating Darwin's "survival of the fittest" rule. The genetic algorithm is highly scalable and can thus judge the pros and cons of generating offspring, continuously optimize the problem, and obtain the optimal solution of the problem.

The model for station optimization and layout of electric carsharing network established in this study has the following characteristics. First, all variables are integers. The variables to be solved include the network deployment and launch matrixes. The network deployment status is represented by a value of 0–1, and the number of cars launched by the network is also an integer. Second, the calculation of two-dimensional discrete volume integrals in the model cannot be solved by Lingo software. The objective function is also nonlinear. Finally, the model involves a large number of 0–1 variables. At present, finding the optimal solution is difficult using a traditional enumeration method on a computer. However, the genetic algorithm can only find a satisfactory solution for the model. The calculation and iteration process of the genetic algorithm is complicated. A problem often needs to be iterated thousands of times to find its optimal solution, but this process is regular and can be repeatedly executed according to a certain program [28]. The genetics are considered in this study. The basic cyclic process and steps of the algorithm are shown in Figure 2 [29,30].

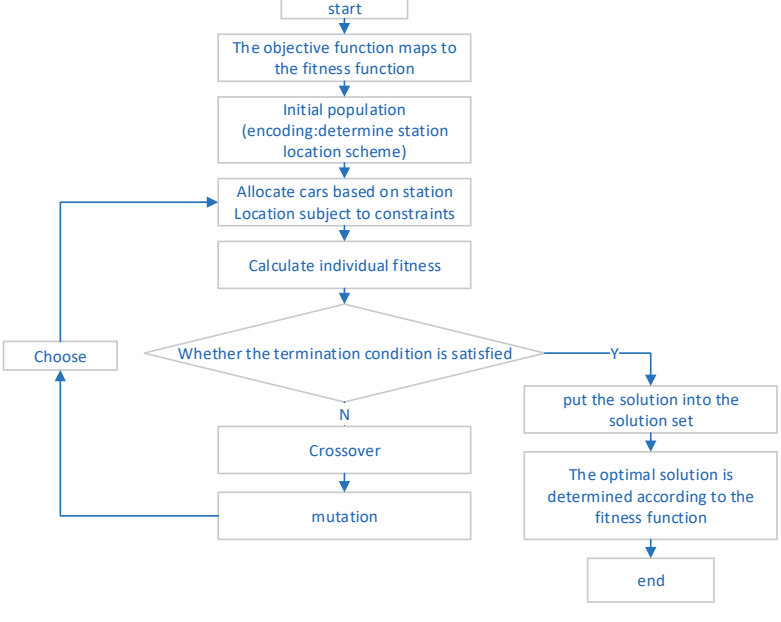


Figure 2. Flowchart of genetic algorithm processing.

In this model, the decision variables include whether a station is established in the grid, the number of cars allocated to a fixed station, and the number of cars allocated to a free station. The demands of each area are known. Therefore, a multivariate mixed integer programming model solution is designed.

3.4.1. Encoding

The encoding steps are as follows. Each grid has a size of 15×12. Each grid is initialized to $l_{uv} = 0$, that is, no station is established in the grid. Grids are then selected to establish a fixed station, and let $l_{uv} = 1$. Accordingly, a chromosome can be formed from all the l_{uv} values. Chromosome matrix is designed as $LC = \left\{L^1, L^2, \ldots, L^z\right\}$, which represents the various attributes of a scheme as a chromosome. In this matrix, each gene fragment L in the chromosome corresponds to a station-selected scheme for any gene fragment $l_{uv} = \{0, 1\}$. Depending on the demands of each grid, the numbers of fixed and free stations allocated to optimize the objective function are allocated.

3.4.2. Fitness Function

The objective function is selected as the fitness function. For simplicity of calculation, the fitness function takes the following form. $\max S = \sum_{u=1}^{m} \sum_{v=1}^{n} \frac{P_{uv} + R_{uv} - |P_{uv} - R_{uv}|}{2}$. The solution with the maximum fitness function is the optimal solution of the station location optimization model.

3.4.3. Selection

This study designs a selection strategy for alternative solution sets. According to different station selection and car allocation schemes, the strategy chooses to be liberated into alternative solution sets that meet the budget constraints. Then, this strategy compares the alternative solution sets to make the objective function optimal into the offspring. Finally, the strategy performs random traversal sampling on the current population until the offspring population is the same size as the parent.

3.4.4. Variation

Crossover of chromosomes is made to redefine new fixed station combinations. The crossover operator is double-point crossover.

For example, we have two chromosomes $L^1 = \begin{bmatrix} l^1_{11}, l^1_{12}, l^1_{13}, \ldots, l^1_{1v}, \ldots, l^1_{1n} \\ l^1_{21}, l^1_{22}, l^1_{23}, \ldots, l^1_{2v}, \ldots, l^1_{2n} \\ \cdots \\ l^1_{m1}, l^1_{m2}, l^1_{m3}, \ldots, l^1_{mv}, \ldots, l^1_{mn} \end{bmatrix}$ and $L^2 = \begin{bmatrix} l^2_{11}, l^2_{12}, l^2_{13}, \ldots, l^2_{1v}, \ldots, l^2_{1n} \\ l^2_{21}, l^2_{22}, l^2_{23}, \ldots, l^2_{2v}, \ldots, l^2_{2n} \\ \cdots \\ l^2_{m1}, l^2_{m2}, l^2_{m3}, \ldots, l^2_{mv}, \ldots, l^2_{mn} \end{bmatrix}$.

Let $\left[\begin{array}{cccc|c|c} l^1_{11}, l^1_{12}, l^1_{13}, \ldots, & l^1_{1v}, & \ldots, & l^1_{1n} \\ l^1_{21}, l^1_{22}, l^1_{23}, \ldots, & l^1_{2v}, & \ldots, & l^1_{2n} \\ \multicolumn{4}{c|}{\cdots} & & \\ l^1_{m1}, l^1_{m2}, l^1_{m3}, \ldots, & l^1_{mv}, & \ldots, & l^1_{mn} \end{array}\right]$ crossover with $\left[\begin{array}{cccc|c|c} l^2_{11}, l^2_{12}, l^2_{13}, \ldots, & l^2_{1v}, & \ldots, & l^2_{1n} \\ l^2_{21}, l^2_{22}, l^2_{23}, \ldots, & l^2_{2v}, & \ldots, & l^2_{2n} \\ \multicolumn{4}{c|}{\cdots} & & \\ l^2_{m1}, l^2_{m2}, l^2_{m3}, \ldots, & l^2_{mv}, & \ldots, & l^2_{mn} \end{array}\right]$. Then, we obtain $\left[\begin{array}{cccc|c|c} l^1_{11}, l^1_{12}, l^1_{13}, \ldots, & l^2_{1v}, & \ldots, & l^1_{1n} \\ l^1_{21}, l^1_{22}, l^1_{23}, \ldots, & l^2_{2v}, & \ldots, & l^1_{2n} \\ \multicolumn{4}{c|}{\cdots} & & \\ l^1_{m1}, l^1_{m2}, l^1_{m3}, \ldots, & l^2_{mv}, & \ldots, & l^1_{mn} \end{array}\right]$ and $\left[\begin{array}{cccc|c|c} l^2_{11}, l^2_{12}, l^2_{13}, \ldots, & l^1_{1v}, & \ldots, & l^2_{1n} \\ l^2_{21}, l^2_{22}, l^2_{23}, \ldots, & l^1_{2v}, & \ldots, & l^2_{2n} \\ \multicolumn{4}{c|}{\cdots} & & \\ l^2_{m1}, l^2_{m2}, l^2_{m3}, \ldots, & l^1_{mv}, & \ldots, & l^2_{mn} \end{array}\right]$. Any row to row and column to column can crossover.

Chromosomal mutations are used to redefine new fixed station locations. The mutation operator is multipoint mutation. A chromosome is randomly selected to mutate to ensure l_{uv} change.

3.4.5. Termination Condition

We determine whether the termination condition is reached. If yes, then end the iteration; if not, then rerun the genetic iteration. We set the maximum number of iterations reached and a plateau reached as the termination condition. When the maximum number of iterations is reached, the fitness function value converges, which indicates that the optimal solution is reached.

The costs are calculated under this allocation plan. If the cost meets the budget constraint and the number of vehicle constraint, then the allocation scheme is added to the set of alternative solutions. Otherwise, we transfer and readjust the number of allocated cars and repeat the steps until the optimal number of allocated cars at this station is obtained.

4. Case Study

4.1. Setting up the Case Study

Lanzhou is an important tourist city in China. The city is an important comprehensive transportation hub. At present, the resident population in Lanzhou City is more than 3.6 million, and the number of cars is more than 1.04 million. Urban transportation problems are prominent in the city because of the large population and cars. The spatial layout of Lanzhou is shown in Figure 3.

Nowadays, the construction of electric carsharing stations is based on experience. Thus, optimization of the entire region is difficult. As a result, the company spends a lot of time and resources in operating. Therefore, the rational construction of an electric carsharing location model helps companies reduce blind investment and improve operating efficiency in station optimization. This research applies the location optimization model to optimize the electric carsharing station in Lanzhou.

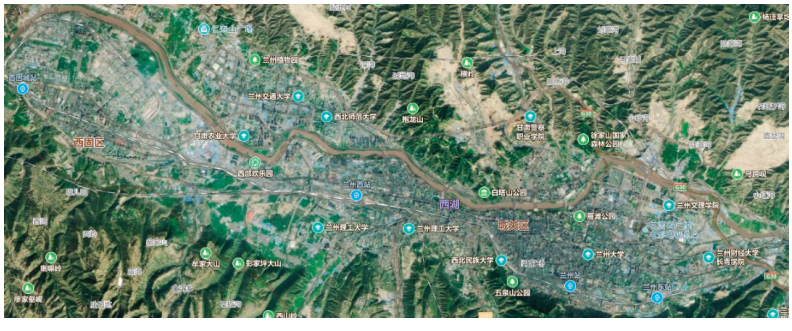


Figure 3. Spatial layout of the study area.

4.2. Parameter Calibration

Lanzhou is a strip-shaped city. This research selects a rectangular research area of 30 km × 12 km in Lanzhou and divides the research area into small areas of 2 km × 1 km. A total of 15 × 12 area is obtained. The specific division is shown in Figure 4.

Figure 4. Diagram of area division.

The demand for electric carsharing D is obtained according to the survey done by the carsharing company. Seven types of land use, namely, transportation hubs, ordinary interchange point, business districts, college campuses, jobs, residential areas, and tourist attractions, have significant effects on the demand for electric carsharing. Using Equation (2), the final carsharing demand matrix is obtained.

We divide the area by a grid of 12 rows and 15 columns and set the fixed station placement plan B as a matrix of 12 rows and 15 columns. K is the free station placement plan. In this research, the location of the free station is preset. Therefore, the capacity of the free station needs to be optimized.

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{2,4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{3,0} & k_{3,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_{4,0} & k_{4,1} & k_{4,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{7,7} & 0 & k_{7,9} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{8,8} & k_{8,9} & k_{8,10} & 0 & k_{8,12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{9,8} & k_{9,9} & 0 & k_{9,11} & 0 & k_{9,13} & k_{9,14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{10,9} & k_{10,10} & k_{10,11} & k_{10,12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_{11,12} & 0 & 0 \end{bmatrix}$$

The service capacity R_{uv} and demand P_{uv} of a single car rental station are generally closely related to the location and capacity of the electric carsharing at the station. It is given by the electric carsharing company based on its own investment level and development strategy. Service intensity usually decreases as the distance to the center of the carsharing station increases. Thus, this research divides the district into core, secondary, and marginal districts. Then, the service matrix of each station can be constructed. This study supposes that a single station has same service capabilities A.

$$A = \begin{bmatrix} 2\% & 10\% & 2\% \\ 10\% & 45\% & 10\% \\ 2\% & 10\% & 2\% \end{bmatrix}$$

V is the average monthly rent matrix per square meter in each small grid.

$$V = \begin{bmatrix} 2 & 3 & 1 & 2 & 2 & 2 & 2 & 2 & 4 & 3 & 3 & 3 & 3 & 2 & 2 \\ 2 & 2 & 3 & 7 & 5 & 5 & 5 & 2 & 8 & 3 & 4 & 5 & 7 & 4 & 2 \\ 15 & 10 & 8 & 7 & 5 & 5 & 5 & 2 & 8 & 3 & 4 & 5 & 7 & 4 & 2 \\ 25 & 26 & 15 & 13 & 8 & 6 & 14 & 12 & 11 & 11 & 11 & 8 & 8 & 5 & 10 \\ 24 & 24 & 18 & 12 & 12 & 10 & 10 & 10 & 10 & 7 & 14 & 12 & 11 & 10 & 8 \\ 11 & 12 & 10 & 10 & 10 & 10 & 8 & 8 & 7 & 7 & 4 & 5 & 7 & 10 & 10 \\ 4 & 4 & 3 & 6 & 5 & 5 & 8 & 8 & 8 & 9 & 16 & 7 & 7 & 9 & 9 \\ 4 & 4 & 3 & 6 & 5 & 9 & 10 & 10 & 10 & 30 & 20 & 20 & 15 & 15 & 15 \\ 8 & 5 & 5 & 8 & 6 & 9 & 10 & 10 & 10 & 30 & 30 & 20 & 20 & 20 & 20 \\ 2 & 3 & 3 & 5 & 8 & 8 & 8 & 8 & 10 & 25 & 25 & 30 & 20 & 19 & 17 \\ 3 & 2 & 2 & 3 & 4 & 4 & 6 & 8 & 8 & 15 & 20 & 24 & 20 & 15 & 15 \\ 2 & 2 & 3 & 3 & 3 & 3 & 5 & 7 & 5 & 3 & 8 & 10 & 25 & 20 & 10 \end{bmatrix}$$

J is the parking price per hour in the free parking area.

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 7 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 8 & 6 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 0 & 5 & 0 & 6 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \end{bmatrix}$$

The company intends to invest $W = 50$ million in these costs.

The cost of buying each car is $b = 100{,}000$ yuan.

The construction cost of each charger is $e = 2000$ yuan.

The average parking time for each car in the free parking area is $t = 1440$ h a year.

The corresponding coefficient $\mu = 20$ of the area of the stations and the size of the stations.

The corresponding coefficient $\lambda = 0.5$ between the scale of the stations and the required number of chargers.

The maximum number of cars dropped at each station is 100.

We design a genetic optimization algorithm to solve the model. The calculation process of the algorithm is shown in Figure 2. The corresponding parameters are set as follows:

The selection operator is tournament selection;

The crossover operator is single-point crossover;

The crossover probability is 0.8;

The mutation operator is multipoint mutation;

The mutation probability is 0.05;

The population size is set to 500;

The stop rule is set to a maximum of 600 iterations.

4.3. Results

The generated genetic algorithm iterative curve is shown in Figure 5. The figure shows that the genetic algorithm iterative curve shows a rising trend, which is consistent with the optimization model

established in this study. Specifically, before the 400th generation the speed of rising is evident, and the degree of dispersion is large. After the 400th generation, the curve is stable, and the fluctuations are reduced. The latter trend continues to the end of the iteration. Therefore, the genetic algorithm converges, and the optimal value of 426 of the stationary period is taken as the optimal solution of the algorithm.

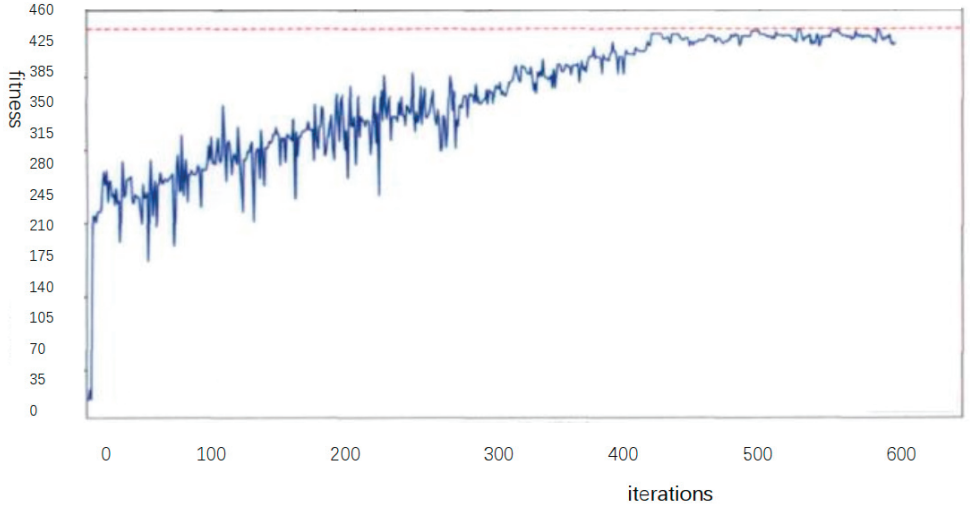


Figure 5. Graph of the iterative process of genetic algorithm.

Figure 6 shows the result of superimposing the service range of the electric carsharing network calculated by the genetic algorithm and the demand range of the demand matrix. As observed, the scope of requirements can be surrounded by the service matrix, which indicates that the requirements can be met. Furthermore, the demands are mainly distributed diagonally, which is similar to the demand matrix and the actual situation.

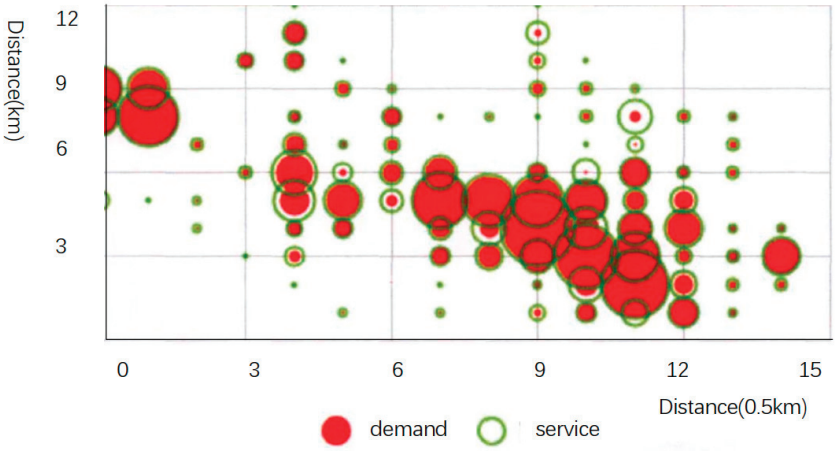
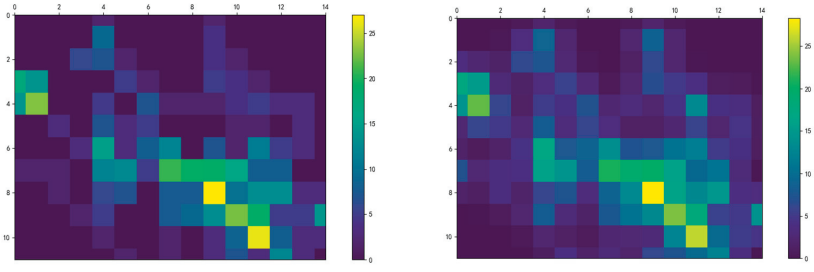


Figure 6. Bubble chart of service and demand ranges.

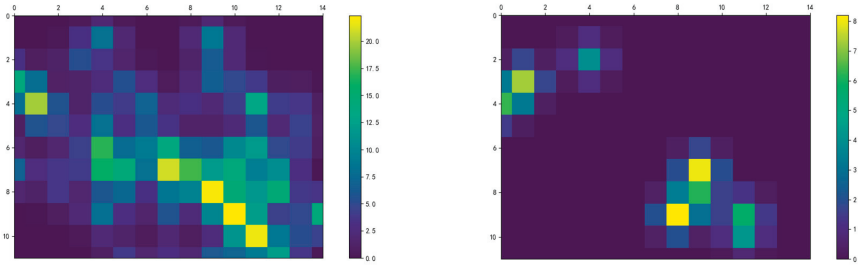
This research meets the overall demands of the city by placing cars in free stations or fixed stations, as shown in Figures 7 and 8. Figure 7 shows the demand range (left) and overall service ability (right)

of free and fixed stations. The lighter the color, the larger the quantity, and vice versa. Figure 8 shows the service capabilities and scope of fixed and free sites. The lighter the color, the greater the service capability, and vice versa.



(**a**) (**b**)

Figure 7. (**a**) Thermal map of demand range and (**b**) overall service range.



(**a**) (**b**)

Figure 8. (**a**) Thermal map of the fixed station and (**b**) free station service range.

Figure 9 presents the heat map for the final network of the stations. Color shades represent the number of services available minus the number of requirements. The figure shows that user demands can be met in most areas. Some cases in the red grid have demands that are not met.

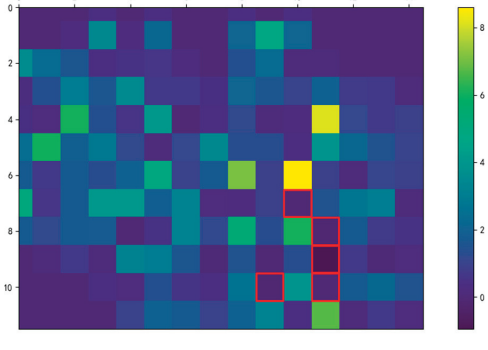


Figure 9. Thermal map of the service excess matrix.

The specific statistics of the station planning are shown in Table 1. The entire area must be resettled at 56 regional points, of which two points are required to set stations at free and fixed stations. A maximum of 24 cars can be installed in a single area. A total of 425 vehicles are settled.

Table 1. Statistical table of the arrangement plan.

Station Type	Station Number	Maximum Number of Cars	Car Number Installed
Free station	8	8	44
Fix station	50	20	381
All station	56	24	425

5. Conclusions and Prospects

The rapid development of the electric carsharing industry has made the way to locate and design electric carsharing stations a critical issue. This study analyzes the development status and existing problems of electric carsharing. An electric carsharing station location model is established in consideration of multiple influencing factors. The main factors for the demand of electric carsharing are summarized by studying the application scenarios and business models of electric carsharing. On the basis of these factors, a quantitative method that allocates cars according to travel demand is proposed. A model of station location is established on the basis of a two-dimensional spatial analysis. A genetic algorithm is designed to solve the problem given the characteristics of the model. The feasibility of the model and solution method is verified using a case study of Lanzhou.

This research can be used for carsharing station location optimization in a new city. A comprehensive station location optimization model is constructed by considering factors, such as the total number of people in the city, the proportion of travel modes, station construction costs, and construction budget. This method can also be used to study the locations of shared bicycles, electric vehicle charging piles, and logistics distribution points. Therefore, this method can be extended.

The station location problem of electric carsharing is a multi-disciplinary problem and has many influencing factors. On the basis of the analysis and summary of the development status of electric carsharing, this study establishes a network location model and the solution methods. The proposed methods can be improved from the following aspects. First, the forecast of the overall demand for electric carsharing in the region is insufficiently accurate. This study only quantitatively predicts the demand from several macro data indicators. Relatively few influencing factors are considered, and the correlation between the factors has not been studied and verified with examples. Further research is needed. Second, some parameters use an average estimation method that may be inconsistent with actual operations. Finally, the model also ignores the effect of other operators on the regional demand. These issues will be complemented in the future research.

Author Contributions: Conceptualization, J.B. and J.C.; methodology, Q.S.; resources, J.B.; writing—original draft preparation, Q.S.; writing—review and editing, J.C. 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 (91846202, 71961137008).

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

References

1. Sai, Q.; Bi, J.; Xie, D.; Guan, W. Identifying and Predicting the Expenditure Level Characteristics of Car-Sharing Users Based on the Empirical Data. *Sustainability* **2019**, *11*, 6689. [CrossRef]
2. Chen, W.; Yang, R.; Yang, H. Research on the Status Quo, Problems and Countermeasures of New Energy Vehicle Time-sharing Development in China—Taking the "EVCARD" Model of Shanghai as an Example. *Compr. Transp.* **2017**, *8*, 89–92.
3. Jorge, D.; Correia, G.H.; Barnhart, C. Comparing Optimal Relocation Operations with Simulated Relocation Policies in One-Way Carsharing Systems. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1667–1675. [CrossRef]

4. Du, Z.; Huang, X.; Ding, T.; Li, Z. Discussion on the Development Dilemma of China's Automobile Time-sharing Leasing. *Automob. Pract. Technol.* **2017**, *19*, 1–3.
5. Zhang, Z. Some Suggestions on Promoting the Healthy Development of Car Time-sharing Leasing. *Transp. Transp.* **2017**, *33*, 27–28.
6. Zhao, Z. Problems and Solutions of Time-sharing Rental Service of Electric Vehicles in China. *J. Longdong Univ.* **2017**, *28*, 132–134.
7. Lu, T.; Huang, J.; Jiang, W. Construction of carsharing network site optimization model: Taking Hongshan District of Wuhan as an example. *J. Transp. Res.* **2017**, *3*, 8–15.
8. Zhu, R. Finance Leasing Paves Way for New Energy Vehicles in China. *Chinas Foreign Trade* **2015**, *1*, 34.
9. Chu, F. Explosive growth! The car sharing market is developing rapidly. *Transp. Manag. World* **2016**, *13*, 44–45.
10. Sawik, B.; Faulin, J.; Perez-Bernabeu, E. *Selected Multi-Criteria Green Vehicle Routing Problems*; Applications of Management Science Book Series: Applications of Management Science; Emerald: Bingley, UK, 2017; Volume 18, pp. 57–83.
11. Ciari, F.; Weis, C.; Balac, M. Evaluating the influence of carsharing stations' location on potential membership: A Swiss case study. *EURO J. Transp. Logist.* **2016**, *5*, 345–369. [CrossRef]
12. De Almeida Correia, G.H.; Antunes, A.P. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transp. Res. Part E Logist. Transp. Rev.* **2012**, *48*, 233–247. [CrossRef]
13. Jiang, B. *Research on Vehicle Supply-Demand Matching under Shared Leasing Model*; Hefei University of Technology: Hefei, China, 2017.
14. Lu, L. Research on Location Optimization of "carsharing" Based on Interval Intuitionistic Fuzzy Soft Set Method Based on Risk Preference. *Business* **2017**, *15*, 214–216.
15. Çalık, H.; Fortz, B. A Benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty. *Transp. Res. Part B Methodol.* **2019**, *125*, 121–150. [CrossRef]
16. Hu, L.; Liu, Y. Joint design of parking capacities and fleet size for one-way station-based carsharing systems with road congestion constraints. *Transp. Res. Part B Methodol.* **2016**, *93*, 268–299. [CrossRef]
17. Wielinski, G.; Trepanier, M.; Morency, C. Exploring Service Usage and Activity Space Evolution in a Free-Floating Carsharing Service. *Transp. Res. Rec.* **2019**, *2673*, 36–49. [CrossRef]
18. Zheng, J.; Qi, G. Location of Charging Stations for Unidirectional Electric carsharing Systems under Fund Constraints. *Appl. Comput. Syst.* **2019**, *28*, 208–214.
19. Lee, J.B.; Byun, W.; Lee, S.H.; Do, M. Correlation between optimal carsharing locations and carbon dioxide emissions in urban areas. *Int. J. Environ. Sci. Technol.* **2014**, *11*, 2319–2328. [CrossRef]
20. Chen, Z.; Qin, K.; Liu, K. Genetic Algorithm-Based Location Optimization and Optimization Design of Carsharing Stations. *Jiangsu Sci. Technol. Inf.* **2019**, *36*, 39–41.
21. Zhu, X.; Li, J.; Liu, Z.; Yang, F. Optimization Approach to Depot Location in Car Sharing Systems with Big Data. In Proceedings of the IEEE International Congress on Big Data, New York, NY, USA, 27 June–2 July 2015.
22. Shoufeng, L.; Ximin, L. Based on Hybrid Genetic Algorithm and Cellular Automata Combined Traffic Signal Control and Route Guidance. In Proceedings of the 2007 Chinese Control Conference, Changsha, China, 26–31 July 2007.
23. Cheng, Y. *Game Study on Urban Transportation Trip Modes under CarSharing*; Harbin Institute of Technology: Harbin, China, 2015.
24. Deng, Y.; Cardin, M.A. Integrating operational decisions into the planning of one-way vehicle-sharing systems under uncertainty. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 407–424. [CrossRef]
25. Thomas, D.; Kovoor, B.C. A Genetic Algorithm Approach to Autonomous Smart Vehicle Parking system. *Procedia Comput. Sci.* **2018**, *125*, 68–76. [CrossRef]
26. Wu, Y.; Yang, X.; Zheng, Z. Research on Optimized Layout of Carsharing Parking Places Based on Improved Genetic Algorithm. *Logist. Sci. Technol.* **2017**, *40*, 78–82.
27. He, L.; Tang, C.; Zhou, R.; Hou, X. Research on SDN enhanced path boxing problem based on genetic algorithm. *Comput. Technol. Dev.* **2019**, *7*, 1–4.
28. Clarke, D.D.; Forsyth, R.; Wright, R. Behavioural factors in accidents at road junctions: The use of a genetic algorithm to extract descriptive rules from police case files. *Accid. Anal. Prev.* **1998**, *30*, 223–234. [CrossRef]

29. Ceylan, H.; Bell, M.G.H. Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transp. Res. Part B Methodol.* **2004**, *38*, 329–342. [CrossRef]
30. Zhang, Q. Design and implementation of optimization algorithm for traffic line selection based on genetic algorithm. *Comput. Eng. Appl.* **2009**, *45*, 226–228.

algorithms

MDPI

Article

Multi-Level Joint Feature Learning for Person Re-Identification

Shaojun Wu [1,2] and Ling Gao [1,2,*]

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250014, Shandong Province, China; wusj98287@gmail.com
[2] Institute of Data Science and Technology, Shandong Normal University, Jinan 250014, Shandong Province, China
* Correspondence: gaoling@sdnu.edu.cn

Received: 1 April 2020; Accepted: 27 April 2020; Published: 29 April 2020

Abstract: In person re-identification, extracting image features is an important step when retrieving pedestrian images. Most of the current methods only extract global features or local features of pedestrian images. Some inconspicuous details are easily ignored when learning image features, which is not efficient or robust to for scenarios with large differences. In this paper, we propose a Multi-level Feature Fusion model that combines both global features and local features of images through deep learning networks to generate more discriminative pedestrian descriptors. Specifically, we extract local features from different depths of network by the Part-based Multi-level Net to fuse low-to-high level local features of pedestrian images. Global-Local Branches are used to extract the local features and global features at the highest level. The experiments have proved that our deep learning model based on multi-level feature fusion works well in person re-identification. The overall results outperform the state of the art with considerable margins on three widely-used datasets. For instance, we achieve 96% Rank-1 accuracy on the Market-1501 dataset and 76.1% mAP on the DukeMTMC-reID dataset, outperforming the existing works by a large margin (more than 6%).

Keywords: deep learning; intelligent monitoring; person re-identification

1. Introduction

Public safety incidents often occur in dense crowds. Therefore, a large number of surveillance cameras are installed and applied in various areas of the city. Person re-identification is a key component technology in the field of urban remote sensor monitoring. For a certain target person appearing in a remote sensing surveillance video or remote sensing pedestrian image, the method of person re-identification can accurately and quickly identify this target person in other remote sensing monitoring fields. The goal of person re-identification is to find the same person from the videos or images captured from different cameras [1], as in Figure 1. Recently, deep learning methods achieve great success by designing feature representations [2–6] or learning robust distance metrics [7–10].

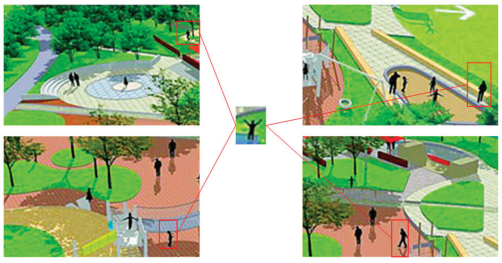


Figure 1. Retrieving the same pedestrian image under different cameras.

The pedestrian features extracted by deep learning can be divided into two types: global features and local features. The global features are extracted from the whole picture, which is easy to calculate and intuitive. These features contain the most significant information of a person (such as the color of pedestrian clothes), which is helpful to indicate the identity of different pedestrians [6]. However, some inconspicuous details (such as hats, belts, etc.) are easily ignored by the global features. For example, if two persons are wearing clothes of the same color, and one of them is wearing a hat, it is hard to discriminate the two persons from only the overall appearance. Moreover, when the background is complex, it is difficult for the global features to associate the images of the same person with different backgrounds into one identity, as shown in Figure 2.

In order to solve the problem of person re-identification, some recent work mainly uses deep learning models to extract local features, using salient local details to match the local features of a queried pedestrian. Local feature information of each body part is extracted by neural network. The similarity between local features is very low, which is more conducive to person re-identification. However, the method of extracting local features may ignore the overall pedestrian information, as shown in Figure 2.

Figure 2. If the network only extracts local features of (**a**), it cannot be determined that those local features belong to the same person. If the network only extracts global features, complex background content can be detrimental to identifying pedestrians, as shown in (**b**). Horizontal arrows mean we horizontally divided the feature map into six parts and extracted the local features of each part.

Representing local information of individuals by locating notable body parts from pedestrian images is also an effective method of person re-identification in recent years [11–13]. Local features are

extracted from different body parts. Each body part contains a small portion of local information from the whole body [14,15]. In this way, we can learn detailed local features from the divided parts which make the features focus on the local details. The learned local features supplement important details, which can be taken as the complementary of the global features.

Therefore, in this paper, the local features and global features are jointly learned for person re-identification. In this paper, we propose a Multi-level Feature Fusion (MFF) model that fuses global features and local features. Moreover, the local features are extracted from different network depths. An MFF model consists of two components: Part-based Multi-level Net (PMN) and Global-Local Branch (GLB). PMN is used to extract local features from different layers of the network. GLB extracts local features and global features at the highest level. The global features and local features are used to perform identity predictions in MFF. We train the MFF model on three classic datasets. Performance of experiments show that the MFF model which fuses global and local features is particularly effective and our model results outweigh many state-of-the-art methods.

The main contributions of our work are as follows:

- We add Part-based Multi-level Net (PMN) to extract local features more comprehensively from lower to higher layers of the network. Compared with other traditional feature extraction methods, PMN can learn more local detailed features from different network layers.
- We join to learn global features and local features. The robustness of the MFF model can be improved by joint learning features. We use multi-class loss functions to classify the features extracted from different network branches separately, which enhances the accuracy of MFF.
- We implement extensive experiments on three challenging person re-identification datasets. Experiments show that our method is superior to existing person re-identification methods.

The remainder of our paper is organized as follows: some related works are reviewed in Section 2. The structure of our proposed model and implementation details are presented in Section 3. Extensive comparative experiment results on three benchmark datasets are shown in Section 4. The conclusions of our work are described in Section 5.

2. Related Work

Person re-identification aims to find matching pedestrian images from different camera views. With the rapid development of deep learning, feature learning by deep networks has become a common practice in person re-identification. Li et al. [16] combined deep siamese network architecture with pedestrian human body feature learning for the first time and achieved higher performance. Zheng et al. [11] proposed a baseline that combined ID-discriminative embedding (IDE) with a ResNet-50 backbone network for modern deep person re-identification systems. Proposed methods also improved the performance of deep person re-identification. Varior et al. [17] described the interrelationship of local parts by computing mid-level features of image pairs. Xiao et al. [18] improved the generalization of different pedestrian scenes by using the Domain Guided Dropout method. Yang et al. [19] used deep learning networks to integrate multiple feature representations together for person re-identification. Some recent works use features of different views or top-view for person re-identification [20,21]. Paolanti et al. [21] extracted neighborhood component features and used multiple nearest neighbor classifiers to identify pedestrians.

For local features extraction, Li et al. [12] proposed a deep learning method called STN. Local features can be easily localized from image patches by learning deep contextual awareness of body and potential parts. Zhao et al. [13] applied deep learning method to align same parts of different images after splitting pedestrian image. Liu et al. [22] utilized attention module to extract part features emphasized of the model. Bai et al. [23] combined some feature slices which are vertically divided into multiple pieces with the LSTM network. Some recent works strengthen the representation of the body part by embedding attention information [22,24,25]. In our proposed method, we extract local features

from several horizontal stripes. At the same time, local features are extracted from different network depths which achieves good performance.

For global feature extraction, a kernel feature map is used to obtain similar information of all patches from different images [26]. Liao et al. [6] proposed a method called Local Maximal Occurrence (LOMO) to represent a local feature which has a positive effect on person re-identification. In our paper, we combine global features and low-to-high level local features together for person re-identification.

In the feature learning phase, classification loss is a commonly used loss function. Some loss functions based on softmax loss achieve state-of-the-art performance in face recognition. Liu et al. [27] proposed L-Softmax to improve the discrimination of pedestrian image features by adding angular constraints to each identity. A-Softmax [28] improves L-Softmax by normalizing weights to recognize by learning angularly discriminative features. Since softmax loss is robust to various multi-class classification tasks and can be used individually [25,29] or in combination with other losses [10,16,23,30–32], softmax loss is often used as a classification for loss function in person re-identification. In our proposed method, we also use softmax loss to solve multi-class tasks.

3. Materials and Methods

Details of our method are described in this section. Proposed Multi-level Feature Fusion (MFF) is introduced in detail which contains two main components: Part-based Multi-level Net (PMN) and Global- Local Branch (GLB), as shown in Figure 3. PMN is used to extract local features from different layers of the network. GLB extracts local features and global features from the final layer. More details of the MFF are introduced in Sections 3.1 and 3.2. The loss function is introduced in Section 3.3.

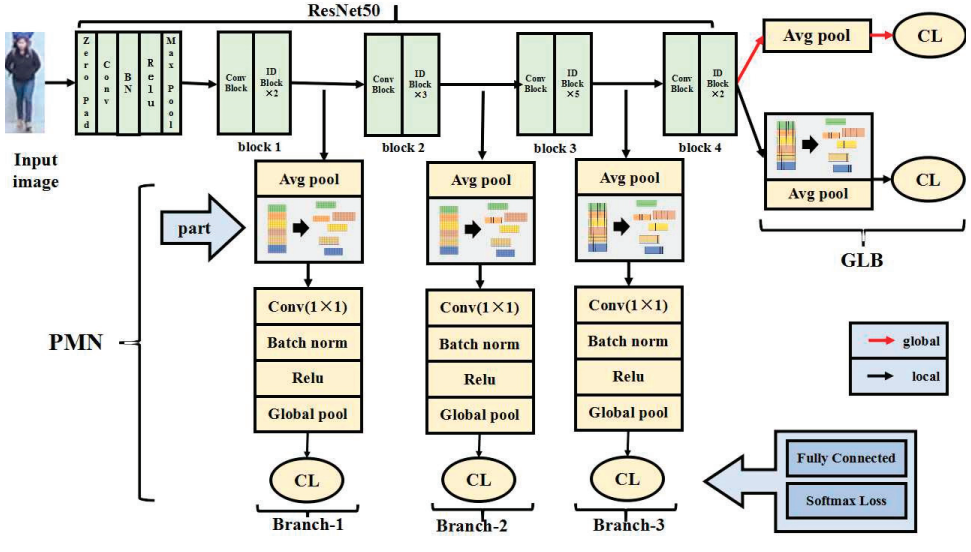


Figure 3. Multi-level Feature Fusion (MFF) architecture is split into two components: Multi-level Net (PMN) and Global-Local Branch (GLB). An input image is fed to the modified ResNet50 to obtain feature maps. The extracted global features are sent to a classifier. Meanwhile, the feature maps are divided into several parts and each part after dimension reduction is sent to a classifier.

3.1. Change of Backbone Network

By considering the relatively effective performance and concise architecture, this paper uses ResNet50 as the backbone network. In order to extract features more accurately, the ResNet50 structure is divided into block1, block2, block3 and block4, as in Figure 3. Then we can extract feature maps between each block and use classifiers to predict identity. Each block consists of conv block (includes

multiple convolution blocks) and identity blocks. The upper layer network of block1 is max pooling layer. The backbone structure of ResNet50 remains unchanged until block4. In this paper, we remove the entire network after block4 (including the global average pooling layer). In this way, feature maps will be retained with more feature information by removing the global average pooling layer.

3.2. Structure of Multi-Level Feature Fusion (MFF)

The combination of global features and local features can learn more information which leads to more accurate pedestrian retrieval results. In this paper, we propose an MFF model which fuses local features and global features together. In MFF, local features and global features are learned identity predictions. As shown in Figure 3, the MFF model is composed of Part-based Multi-level Net (PMN) and Global-Local Branch (GLB). The structure of PMN and GLB are introduced separately. And Table 1 shows the dimensions of the features extracted from each branch.

Table 1. Comparison of the settings for five branches.

Branch	Dimension
Branch-1	256×6
Branch-2	256×6
Branch-3	256×6
GLB-1	256
GLB-2	256×6

The structure of Global-Local Branch (GLB) consists of two parts to extract local and global features from the deepest layer of the network, respectively. Given an input image, we can obtain the feature maps through the backbone network. Then an average pooling layer and a classifier are employed after the ResNet50 network to get the 256-dimension global features. The classifier is composed of a fully connected layer and a softmax layer to get the prediction of pedestrian identity from the global feature. The second branch of GLB is used to extract local features from the deepest layer of the network. In order to extract the local features, we divide the feature maps horizontally into six parts as shown in Figure 3. We add an average pooling layer and a classifier after the divided feature map to get the prediction of pedestrian identity.

The structure of Part-based Multi-level Net (PMN) consists of three parts (Branch-1, Branch-2 and Branch-3) which is used to extract local features from lower to higher layers of the network, as shown in Figure 4. ResNet50 consists of four blocks, and we add Branch-1, Branch-2 and Branch-3 between each pair of continuous blocks. In each branch, firstly, we apply an average pooling on the corresponding output feature map. Then the feature map is divided into six parts horizontally as introduced in previous subsection. We add a 1×1 kernel-sized convolutional layer, a batch normalization layer, a relu layer and a global pooling layer to obtain 6×256-dimension local features. Then each local feature is input into a classifier, where each classifier is implemented with a fully-connected (FC) layer and a softmax layer. The classifier is used for the identity prediction. Note that, Branch-1, Branch-2 and Branch-3 run in parallel.

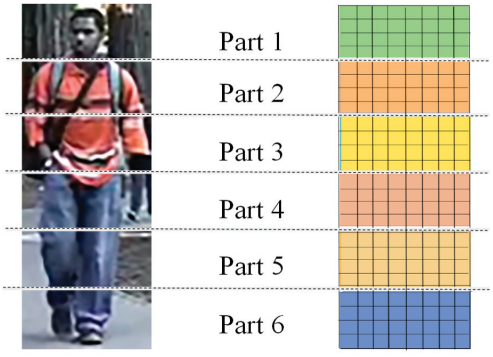


Figure 4. Method of body part partitions is shown above. The feature maps extracted from the deep learning model is horizontally divided into six parts.

3.3. Loss Function

In our paper, we regard the person identification task as a multi-class classification problem. Considering that softmax loss is widely used in various deep person re-identification methods, we employ softmax loss as the loss function for classification in training stage.

In MFF, we regard person re-identification task as a multi-class classification problem. For *i*-th learned class vector h_i, the softmax loss function is described as follows:

$$L_{softmax} = -\sum_{i=1}^{M} \log \frac{\exp\left(K_{yi}^{T} h_i\right)}{\sum_{c=1}^{D} \exp\left(K_c^{T} h_i\right)} \tag{1}$$

where K_c is the weight of class c, D is the number of classes in training dataset, K_{yi} is the weight of $yi - th$ in fully connected layer, $yi - th$ is the $i - th$ value of output vector y. M is the size of mini-batch in training process. In MFF, the softmax loss is employed into the features extracted by GLB and PMN.

The final loss function is formulated as follows:

$$L = L_{softmax}^{G} + L_{softmax}^{L} + L_{softmax}^{L_1} + L_{softmax}^{L_2} + L_{softmax}^{L_3} \tag{2}$$

where $L_{softmax}^{G}$ and $L_{softmax}^{L}$ represent the identity classification tasks in global and local branches of GLB, $L_{softmax}^{L_1}$, $L_{softmax}^{L_2}$ and $L_{softmax}^{L_3}$ represent the identity classification tasks in Branch-1, Branch-2 and Branch-3 of PMN, respectively.

Each classifier predicts the most similar pedestrian images when using a single classifier to make decisions. A pedestrian image with the same identity as a query image is usually determined as the most similar pedestrian image in the process of classification. We vote on the prediction results obtained by five classifiers to get the final classification prediction results.

4. Results

4.1. Datasets

In order to evaluate the performance of the MFF model, here we evaluate three datasets in the experiments, i.e., Market-1501 [33], DukeMTMC-reID [6] and CUHK03 [34]. The dataset of person re-identification is divided into Training_set, Verification_set, Query and Gallery. In our experiment, the network model is trained on the training set. Then we calculate the similarity of features extracted from Query and Gallery which is used to find similar pedestrian images of Query in Gallery. Pedestrian images of the Gallery are sorted according to the similarity of image features, as shown in Figure 5.

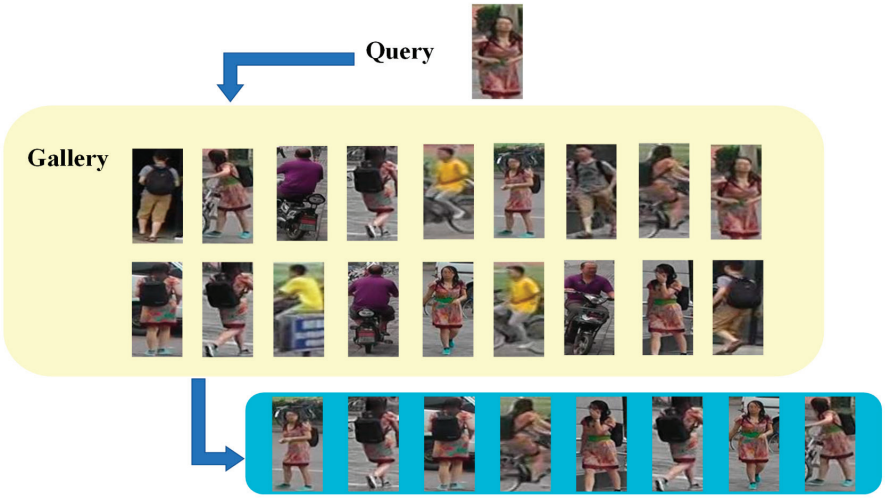


Figure 5. Examples of person re-identification. The similar pedestrian images of Query are shown in blue box.

The Market-1501 [33] dataset includes 1501 identities captured by six cameras and 32,668 detected pedestrian rectangles under six camera viewpoints. In this dataset, each pedestrian contains at least two camera viewpoints. The training set is consisted of 751 identities and each identity includes 17.2 training data on average. The test set is composed of 19,732 images of 750 identities. The pedestrian detection rectangle in the gallery is detected by DPM [35]. Here, we use mean Average Precision (mAP) to evaluate person re-identification algorithms.

The DukeMTMC-reID [6] dataset consists 36,411 images of 1404 identities. With those images collected by eight cameras and each image sampled every 120 frames from the video. This dataset is composed of 16,552 training images, 2228 query images and 17,661 gallery images. Half of the identities are randomly sampled as training sets while the others as test sets. DukeMTMC-reID offers human labeled bounding boxes.

The CUHK03 [34] dataset is composed of 13,614 images and 1467 identities. Each identity automatically captured by two cameras. In this dataset, bounding boxes are provided by two different ways: automatically detected which is the same as Market-1501 dataset and hand-labeled bounding boxes. Here we use two kinds of bounding boxes in this paper. In the whole experiment, we evaluate the single-query setting and adopt new test protocol proposed in [36] which is similar to Market-1501. CUHK03 is divided into a training set consisting of 756 pedestrians and a test set of 700 pedestrians in the new protocol. A randomly selected image is used as query image while the rest is used as gallery. In this way, each pedestrian gets multiple ground truths in gallery.

The detailed information about these datasets is summarized in Table 2. Three widely-used person re-identification datasets contain many challenges, such as misalignment, low resolutions, viewpoints and background clusters. In addition, Figure 6 shows some image samples of the four datasets.

Table 2. The details of person re-identification dataset.

Dataset	Release Tine	Identities	Cameras	Crop Size	Label Method
Market-1501	2015	1501	6	Vary	Hand/DPM
DukeMTMC-reID	2017	1812	8	128×64	Hand
CUHK03	2014	1467	10	Vary	Hand/DPM

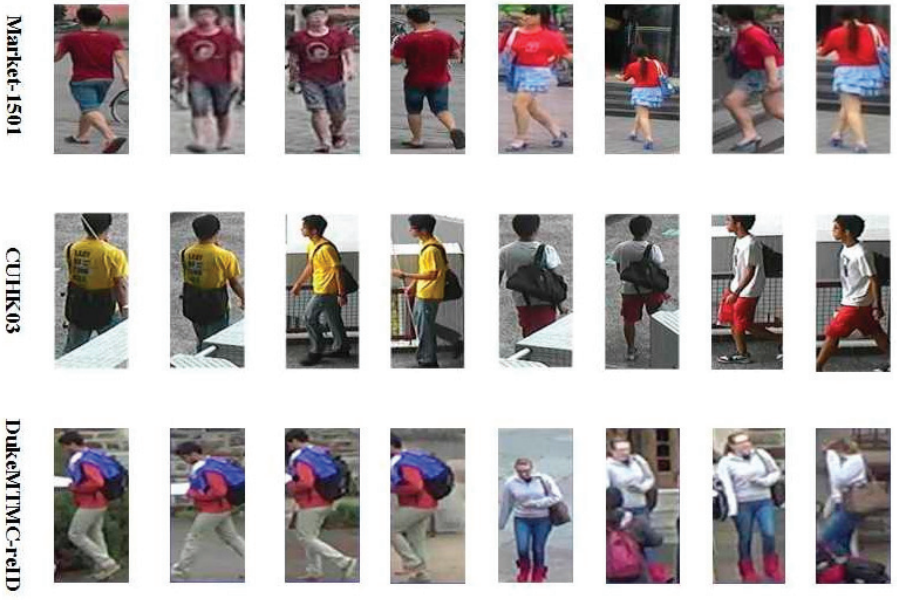


Figure 6. Some samples from Market-1501, CUHK03 and DukeMTMC-reID. Here each row includes two different identities captured under different cameras.

For each query image, we merge the five feature vectors into one and calculate the Euclidean distance between query image and pedestrian image in gallery. We use the Euclidean distance value to rank the images. The higher the ranking, the more similar the image is to the query image. Then we arrange them in descending order according to the Euclidean distance, and use the Cumulative Match Characteristic (CMC) curve to show the performance. In terms of performance measurement, we use the Rank-1 accuracy and the mean Average Precision (mAp).

Mean Average Precision (mAP) is an important evaluation indicator for person re-identification. Precision and recall are important components of mean Average Precision. Precision is the ability of a model to identify only the relevant objects. Recall is the ability of a model to find all the relevant cases. The precision and recall are expressed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

where TP means the number of true positive, FP means the number of false positive, FN means the number of false negative.

Average precision (AP) means the mean of the highest precision under different recalls, which is expressed as follows:

$$AP = \frac{1}{R}\sum_{r \in R} \text{Precision}(r) \tag{5}$$

MAP is the average value of the AP, which is expressed as follows:

$$mAP = \frac{1}{M}\sum_{m \in M} AP(m) \tag{6}$$

4.2. Implementation Details

We pre-trained ResNet50 on ImageNet [37] and used the weight of ResNet50 in MFF. Our training environment is Pytorch and code is edited using python. The computer configuration system is 64-bit ubuntu 16.04LTS. Single-GPU training is used in MFF and the type of GPU is NVIDA GEFORCE GTX 1080. Considering the configuration of the graphics card, we set batch size to 32. Due to differences between three datasets, the learning rate of each dataset is different. Learning rate of Market-1501 is 0.05. Learning rate is set to 0.045 when training on DukeMTMC-reID. The learning rate of CUHK03 is 0.08. The entire training process is terminated in 60 epochs. We randomly select one image as the query image which means we conduct all the experiments under single-query settings, and the input pedestrian images are resized to 384×192.

4.3. Comparison with Market1501

Comparison with the proposed method on Market-1501 is detailed in Table 3. The MFF model is compared with several state-of-the-art person re-identification methods on Market-1501 in recent years, for example, the bag of words model BoW+KISSME [33] with a hand-crafted method, the SVDNet [34] using global features extracted by deep learning model, and the part-aligned representation PAR [17] using part features extracted by a deep learning model. We can observe from Table 3 that the proposed MFF model gets best results in Rank-1 accuracy, Rank-5 accuracy and Rank-10 accuracy. In the experiment, we use mean average precision (mAP) as an evaluation index of person re-identification. The MFF model achieves 87.9% mAP on the Market-1501, which is 18.8% higher than the best proposed method. In addition, the MFF model achieves Rank-1 accuracy of 96.0%, which is 11.1% higher than the best method. Rank-5 accuracy of our model achieves 98.7%, 4.5% better than the best compared method. This is because the MFF model fuses the global features and local features together. Moreover, adding PMN when extracting local features is also helpful to obtain better results.

Table 3. Comparison with existing methods on Market1501.

Method	**Market1501**			
	Rank-1	**Rank-5**	**Rank-10**	**mAP**
BoW + KISSME [33]	44.4	63.9	72.2	20.8
WARCA [36]	45.2	68.1	76.0	-
SVDNet [34]	82.3	92.3	95.2	62.1
PDC [38]	84.4	92.7	94.9	63.4
Triplet Loss [39]	84.9	94.2	-	69.1
DML [40]	87.7	-	-	68.8
PAR [13]	81.0	92.0	94.7	63.4
MFF (Ours)	96.0	98.7	99.3	87.9

The above shows Rank-1 to Rank-5 accuracy (%) and mean Average Precision (mAP) (%).

4.4. Comparison with CUHK03

Comparison between the proposed method and CUHK03 is detailed in Tables 4 and 5. We conduct experiments on a CUHK03-detected dataset and a CUHK03-labeled dataset, respectively. We only use the single-query method for person re-identification on CUHK03-detected and CUHK03-labeled datasets. In this paper, our model is compared with many methods, such as LOMO+KISSME [6] using a horizontal occurrence model, pedestrian alignment network [41] and HA-CNN [25] using harmonious attention network. In this experiment, we use Rank-1 accuracy and mAP as evaluation indicators. As shown in Table 4, the MFF model achieves Rank-1 accuracy of 67.4% which is 0.6% higher than the best experimental result on CUHK03-detected data. Additionally, the mAP reaches 66.7%, which is 0.7% better than the best experimental result. Comparison results obtained on CUHK03-labeled are as follows: we surpass MGN by 1.6% in Rank-1 accuracy for the single-query setting. The MFF model reaches mAP of 68.8%. Compared with other deep learning methods, our

model is even more discriminative, which is attributed to our global feature extraction and each-part feature extraction. We believe that local feature extraction benefits from PMN, which is because PMN can extract low-to-high level features more comprehensively.

Table 4. Comparison with existing methods on CUHK03-detected data.

Method	CUHK03-Detected	
	Rank-1	mAP
BoW + KISSME [33]	6.4	6.4
LOMO + KISSME [6]	12.8	11.5
IDE [42]	21.3	19.7
PAN [41]	36.3	34.0
DPFL [43]	40.7	37.0
SVDNet [34]	41.5	37.3
HA-CNN [25]	41.7	38.6
MLFN [44]	52.8	47.8
PCB+RPP [11]	63.7	57.5
MGN [45]	66.8	66.0
MFF (Ours)	67.4	66.7

Rank-1 accuracy (%) and mAP (%) are compared.

Table 5. Comparison with existing methods on CUHK03-labeled data.

Method	CUHK03-Labeled	
	Rank-1	mAP
BoW + KISSME [31]	7.9	6.4
LOMO + KISSME [6]	14.8	11.5
IDE [42]	22.2	19.7
PAN [41]	36.9	34.0
DPFL [43]	43.0	37.0
SVDNet [34]	40.9	37.3
HA-CNN [25]	44.4	38.6
MLFN [44]	54.7	49.2
MGN [45]	68.0	67.4
MFF (Ours)	69.6	68.8

Rank-1 accuracy (%) and mAP (%) are compared.

4.5. Comparison with DukeMTMC-reID

We compare the MFF model with a state-of-the-art model on DukeMTMC-reID. Comparative details are shown in Table 6. Methods of extracting features are different in Table 6, for example, LOMO+KISSME [6] extract local features with a horizontal occurrence model, whereas PAN [41] and SVDNet [34] use a deep learning method to extract global features.

Table 6. Comparison with existing methods on DukeMTMC-reID.

Method	DukeMTMC-reID	
	Rank-1	mAP
BoW + KISSME [33]	25.1	12.2
LOMO + KISSME [6]	30.8	17.0
Verif + Identif [46]	68.9	49.3
ACRN [47]	72.6	52.0
PAN [41]	71.6	51.5
SVDNet [34]	76.7	56.8
DPFL [43]	79.2	60.6
HA-CNN [25]	80.5	63.8
Deep-Person [48]	80.9	64.8
PCB+RPP [11]	83.3	69.2
MFF (Ours)	86.0	76.1

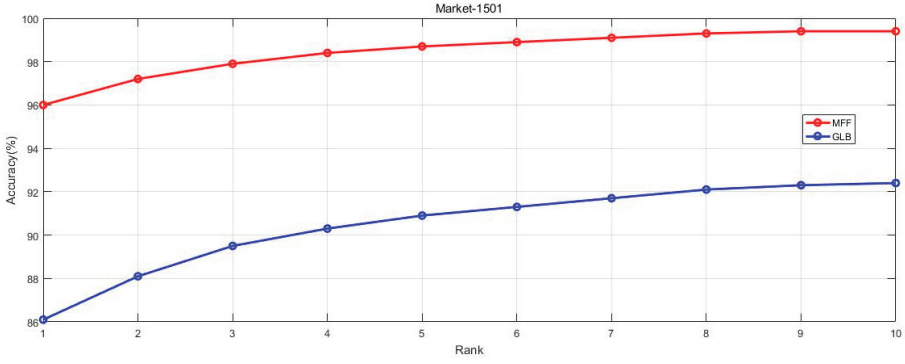
Rank-1 accuracy (%) and mAP (%) are compared above.

We evaluate the MFF model on DukeMTMC-reID with single-query-setting and the significant advantage can be observed in Table 6. Rank-1 accuracy reaches 86.0% which achieves the highest accuracy in comparison methods. We also use mAP as an evaluation indicator. MFF model reaches 76.1% in mAP. Extracting local features and global features enrich the available features when searching for target pedestrians. Adding a classifier in different levels of ResNet50, which is good for extracting part features, can also increase the accuracy of our model. In addition, we visualize the top-10 ranking results on DukeMTMC-reID for some randomly-selected query pedestrian images in Figure 7.

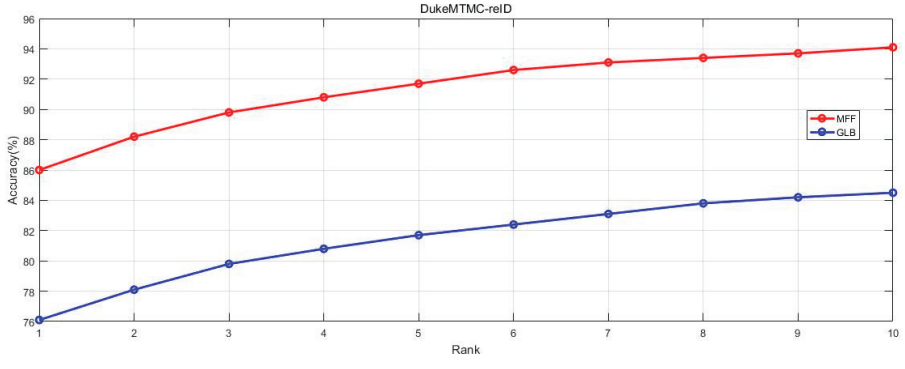
Figure 7. One example query image in DukeMTMC-reID dataset and ranking list results from Rank-1 to Rank-10 using MFF model. The blue boundary means true positive and red means false positive.

4.6. Effectiveness of PMN

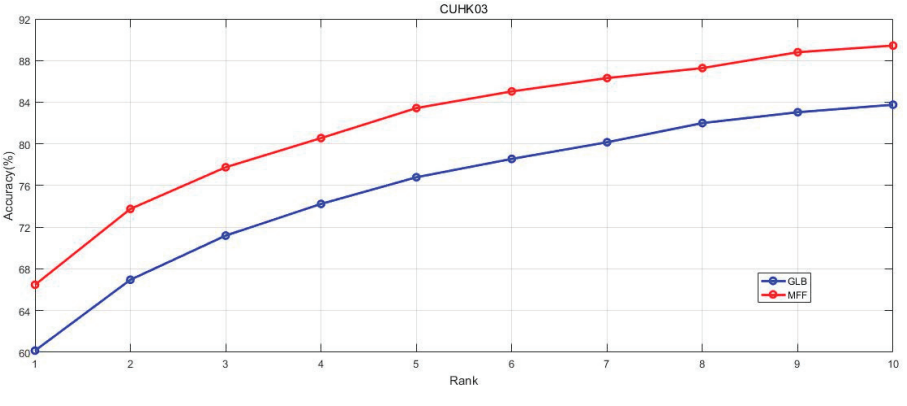
We evaluate the MFF model compared to three classic datasets: Market1501, CUHK03 and DukeMTMC-reID. PMN is proposed to extract local features from the low-to-high level layers. In order to further explore the influence of the PMN model, we conduct two experiments on each dataset. Firstly, we remove the structure of the PMN model. We fuse local features and global features extracted from entire backbone network. GLB is the structure without the PMN model, as in Figure 8. Experiments on GLB can clearly test the performance of our model without adding the PMN structure. Then we train the MFF model on three datasets and report their performance in Figure 8. Difference between MFF and GLB is that MFF fuses low-to-high level local features.



(**a**)

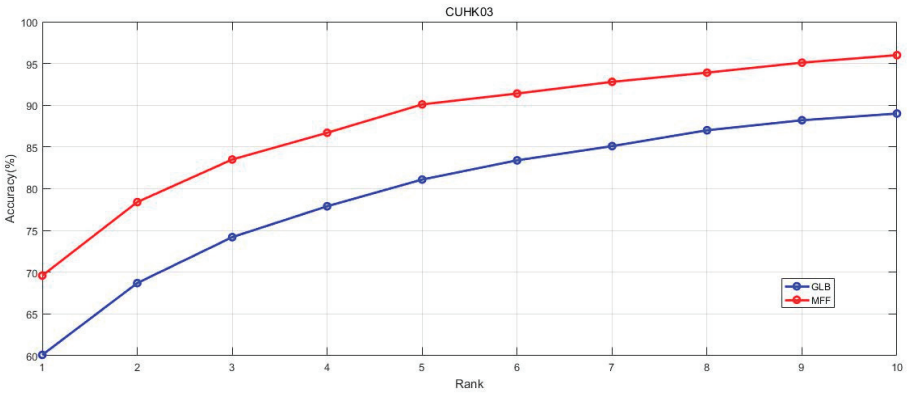


(**b**)



(**c**)

Figure 8. *Cont.*



(**d**)

Figure 8. Evaluation of GLB and MFF are shown above. Rank-1 to Rank-10 accuracy is compared on three datasets. From (**a**–**d**) is the evaluation performance on Market-1501, DukeMTMC-reID, CUHK03-detected and CUHK03-labeled.

We exhaustively train MFF and GLB on three datasets separately and use Rank-1 accuracy to Rank-10 accuracy as the evaluation standard. In Figure 8, a comparison of experimental results of two models not only shows the effect of model enhancement after fusing low-to-high level local features, but also shows that the improvement effect of PMN on each dataset is different. PMN structure has the most significant effect on CUHK03 especially on CUHK03-labeled data. But the effect on Market-1501 is less significant. Figure 8 shows that rank accuracy of MFF is higher than GLB on three datasets, which proves that low-to-high local features extracted by PMN structure have a positive impact on person re-identification.

4.7. Influence of the Number of Parts

In this paper, we use the method of dividing a pedestrian image into several parts to extract local features. The visualization of the delicate parts is shown in Figure 9. Intuitively, the granularity of the part feature affects the results. When the number of parts is one, the learned feature is a global feature. As the number of divided parts increases, the retrieval accuracy increases. However, accuracy does not always increase with the number parts, as shown in Figure 10. Rank-1 accuracy of three datasets shows that when the number of parts increases to eight, the performance drops dramatically. The over-increased parts actually compromise the extraction of local features. Therefore, we use six parts in our experiments.

Figure 9. Visualization of the parts under six values.

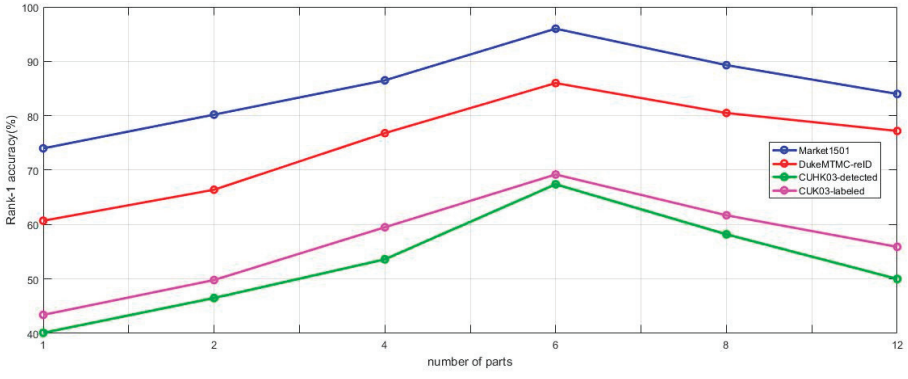


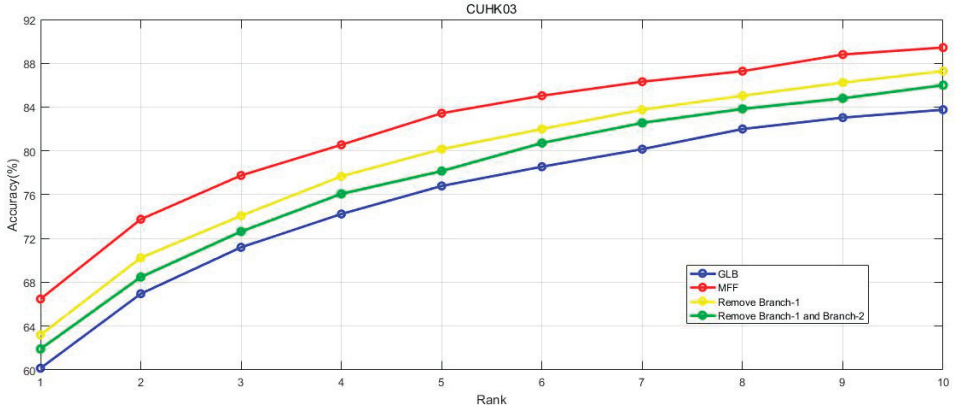
Figure 10. Visualization of the parts under six values.

Discussion: we divide the pedestrian image into six parts to get the best results. We consider different proportions and attributes of body parts. We divide the image into six parts according to the position of the elbow joint, crotch, knee joint, etc., as shown in Figure 2. Due to the limitation of joints, the grate range of human motion is limited to these six parts. The image is divided into six parts to ensure that the local features of each part have a high degree of recognition when a pedestrian is engaged in a wide range of activities. In addition, we also consider the effect of attributes on the results. The relevant attributes in pedestrian images include clothing categories (dresses, shorts etc.), clothing color, hat, hair, etc. The recognition of the attribute features of each part is also strengthened after dividing the image into six parts.

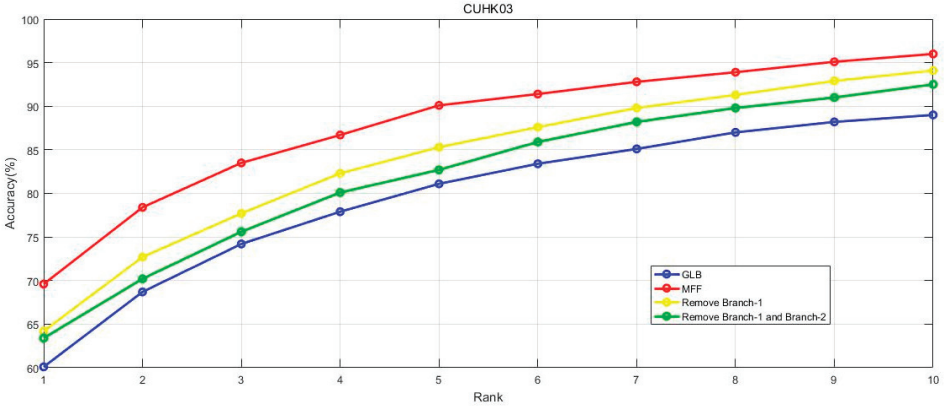
4.8. Influence of the PMN Branches

Low-to-high level local features are extracted by Branch-1 to Branch-3 as in Figure 3. To verify the effectiveness of different branches in PMN, we remove the branches of PMN in different ways and the experimental results are compared in Figure 11. The way of removing branches is as follows. (1) Only Branch-1 is removed. (2) Branch-1 and Branch 2 are both removed. (3) Structure of PMN (Branch-1 to Branch-2) is removed (GLB). (4) No branches are removed (MFF). In Figure 11, we can observe that MFF model achieves the highest rank precision. Removing Branch-1 means not extracting low-level local features which reduces the rank accuracy. In the same way, the more branches in PMN are removed, the lower rank accuracy of the model. This experiment proves that sampling local features from different depths is effective for MFF.

We can try to use PMN networks with different network structures to extract features in the future. In addition, the PMN branches can be used for face recognition to extract facial features from different network depths and learn higher discriminative features. PMN has a wide range of applications and can also be used in other image recognition networks.



(**a**)



(**b**)

Figure 11. Impact of low-to-high level local features. Rank-1 to Rank-10 accuracy is compared on datasets CUHK03-detected (**a**) and CUHK03-labeled (**b**).

5. Conclusions

This paper mainly verified the important role of our model in solving person re-identification problems. A deep learning network called Multi-level Feature Fusion (MFF) is proposed to extract local features and global features. The proposed Part-based Multi-level Net (PMN) structure not only extracts local features more comprehensively from low to high levels, but also can be flexibly applied into different deep learning models. PMN greatly improves the performance of Multi-level Feature Fusion (MFF) by extracting different levels of local features. A more comprehensive feature fusion effectively improves the accuracy of searching for the target person in person re-identification and outperforms the current state-of-the-art methods with considerable margins.

Author Contributions: Conceptualization, S.W.; Methodology, S.W.; Resources, L.G.; Validation, S.W. and L.G.; Writing – original draft, S.W.; Writing – review and editing, S.W. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61672329).

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Ristani, E.; Tomasi, C. Features for Multi-Target Multi-Camera Tracking and Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 28 June 2018; pp. 6036–6046.
2. Wang, C.; Zhang, Q.; Huang, C.; Liu, W.; Wang, X. Mancs: A Multi-task Attentional Network with Curriculum Sampling for Person Re-Identification. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 6 September 2018; pp. 365–381.
3. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 24 July 2017; pp. 7291–7299.
4. Gong, K.; Liang, X.; Zhang, D.; Shen, X.; Lin, L. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 16 July 2017; pp. 932–940.
5. Wei, L.; Zhang, S.; Yao, H.W.; Gao, W.; Tian, Q. GLAD: Global-local-alignment descriptor for pedestrian retrieval. In Proceedings of the ACM International Conference on Multimedia, Mountain View, CA, USA, 13 October 2017; pp. 420–428.
6. Liao, S.; Hu, Y.; Zhu, X.; Li, S.Z. Person re-identification by local maximal occurrence representation and metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 20 June 2015; pp. 2197–2206.
7. Xiao, Q.; Luo, H.; Zhang, C. Margin Sample Mining Loss: A Deep Learning Based Method for Person Re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 7 July 2017.
8. Li, W.; Zhu, X.; Gong, S. Person re-identification by deep joint learning of multi-loss classification. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 1 May 2017; pp. 2194–2200.
9. Yao, H.; Zhang, S.; Zhang, Y.; Li, J.; Tian, Q. Deep representation learning with part loss for person re-identification. *IEEE Trans. Image Process.* **2019**, *28*, 2860–2871. [CrossRef] [PubMed]
10. Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; Zheng, N. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 14 June 2016; pp. 1335–1344.
11. Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; Wang, S. Beyond Part Models: Person Retrieval with Refined Part Pooling. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 6 September 2018; pp. 480–496.
12. Li, D.; Chen, X.; Zhang, Z. Learning deep context-aware features over body and latent parts for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 25 July 2017; pp. 384–393.
13. Zhao, L.; Li, X.; Wang, J.; Zhuang, Y. Deeply-learned part-aligned representations for person re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 23 October 2017; pp. 3219–3228.
14. Kalayeh, M.M.; Basaran, E.; Gökmen, M.; Kamasak, M.E.; Shah, M. Human Semantic Parsing for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 20 June 2018; pp. 1062–1071.
15. Suh, Y.; Wang, J.; Tang, S.; Mei, T.; Lee, K.M. Part-Aligned Bilinear Representations for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 9 June 2018; pp. 402–419.
16. Li, W.; Zhao, R.; Xiao, T.; Wang, X. Deepreid: Deep filter pairing neural network for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 18 June 2014; pp. 152–159.
17. Varior, R.R.; Haloi, M.; Wang, G. Gated siamese convolutional neural network architecture for human re-identification. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 24 October 2016; pp. 791–808.

18. Xiao, T.; Li, H.; Ouyang, W.; Wang, X. Learning deep feature representations with domain guided dropout for person reidentification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, Las Vegas, NV, USA, 26 June 2016; pp. 1249–1258.
19. Yang, Y.; Liu, X.; Ye, Q.; Tao, D. Ensemble Learning-Based Person Re-identification with Multiple Feature Representations. *Complexity* **2018**, *2018*. [CrossRef]
20. Wang, H.; Zhu, X.; Gong, S.; Xiang, T. Person re-identification in identity regression space. *Int. J. Comput. Vis.* **2018**, *126*, 1288–1310. [CrossRef] [PubMed]
21. Paolanti, M.; Romeo, L.; Liciotti, D.; Pietrini, R.; Cenci, A.; Frontoni, E.; Zingaretti, P. Person Re-Identification with RGB-D Camera in Top-View configuration through Multiple Nearest Neighbor Classifiers and Neighborhood Component Features Selection. *Sensors* **2018**, *18*, 3471. [CrossRef] [PubMed]
22. Liu, X.; Zhao, H.; Tian, M.; Sheng, L.; Shao, J.; Yi, S.; Yan, J.; Wang, X. Hydraplus-net: Attentive deep features for pedestrian analysis. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 18 October 2017; pp. 350–359.
23. Bai, X.; Yang, M.; Huang, T.; Dou, Z.; Yu, R.; Xu, Y. Deep-Person: Learning Discriminative Deep Features for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 20 November 2017.
24. Liu, H.; Feng, J.; Qi, M.; Jiang, J.; Yan, S. End-to-End Comparative Attention Networks for Person Re-Identification. *IEEE Trans. Image Process.* **2017**, 3492–3506. [CrossRef] [PubMed]
25. Lei, W.; Zhu, X.; Gong, S. Harmonious Attention Network for Person Re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 6 June 2018; pp. 2285–2294.
26. Chen, D.; Yuan, Z.; Hua, G.; Zheng, N.; Wang, J. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 1 June 2015; pp. 1565–1573.
27. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-Margin Softmax Loss for Convolutional Neural Networks. *arXiv* **2016**, arXiv:1612.02295.
28. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. SphereFace: Deep hypersphere embedding for face recognition. In Proceedings of the Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6738–6746.
29. Ahmed, E.; Jones, M.; Marks, T.K. An improved deep learning architecture for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 10 June 2015; pp. 3908–3916.
30. Zhang, X.; Luo, H.; Fan, X.; Xiang, W.; Sun, Y.; Xiao, Q.; Jiang, W.; Zhang, C.; Sun, J. Alignedreid: Surpassing human-level performance in person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 8 November 2017.
31. Chen, W.; Chen, X.; Zhang, J.; Huang, K. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 6 July 2017; pp. 403–412.
32. Hoffer, E.; Ailon, N. Deep Metric Learning Using Triplet Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 20 December 2015.
33. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable person reidentification: A benchmark. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 9 December 2015; pp. 1116–1124.
34. Sun, Y.; Zheng, L.; Deng, W.; Wang, S. SVDNet for pedestrian retrieval. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 16 October 2017; pp. 3800–3808.
35. Engel, C.; Baumgartner, P.; Holzmann, M.; Nutzel, J.F. Person re-identification by support vector ranking. In Proceedings of the British Machine Vision Conference, Aberystwyth, UK, 1 August 2010.
36. Jose, C.; Fleuret, F. Scalable metric learning via weighted approximate rank component analysis. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 1 October 2016; pp. 875–890.
37. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Miami, FL, USA, 7 June 2009.

38. Su, C.; Li, J.; Zhang, S.; Xing, J.; Gao, W.; Tian, Q. Pose-driven deep convolutional model for person re-identification. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 25 October 2017; pp. 3960–3969.
39. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person reidentification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 17 March 2017.
40. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 1 June 2018; pp. 4320–4328.
41. Zheng, Z.; Zheng, L.; Yang, Y. Person Alignment Network for Large-scale Person Re-identification. *IEEE Trans. Image Process.* **2018**. [CrossRef]
42. Zheng, L.; Yang, Y.; Hauptmann, A.G. Person Re-identification: Past, Present and Feature. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 10 October 2016.
43. Chen, Y.; Zhu, X.; Gong, S. Person re-identification by deep learning multi-scale representations. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 15 September 2017; pp. 2590–2600.
44. Chang, X.; Hospedales, T.M.; Xiang, T. Multi-Level Factorisation Net for Person Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 24 March 2018; pp. 2109–2118.
45. Wang, G.; Yuan, Y.; Chen, X.; Li, J.; Zhou, X. Learning Disriminative Features with Multiple Granularities for Person Re-Identification. *arXiv* **2018**, arXiv:1804.01438.
46. Zheng, Z.; Zheng, L.Y.; Yang, Y. A Discriminatively Learned Cnn Embedding for Person Re-identification. *ACM Trans. Multimed. Comput. Commun. Appl.* **2018**, *14*, 13. [CrossRef]
47. Schumann, A.; Stiefelhagen, R. Person Re-identification by Deep Learning Attribute-Complementary Information. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 8 October 2017; pp. 20–28.
48. Bai, X.; Yang, M.; Huang, T.; Dou, Z.; Yu, R.; Xu, Y. Deep-Person: Learning Discriminative Deep Features for Person Re-Identification. *arXiv* **2017**, arXiv:1711.10658. [CrossRef]

algorithms

MDPI

Article

Automobile Fine-Grained Detection Algorithm Based on Multi-Improved YOLOv3 in Smart Streetlights

Fan Yang [1,2,*], Deming Yang [1,*], Zhiming He [1], Yuanhua Fu [1] and Kui Jiang [1]

[1] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; zmhe@uestc.edu.cn (Z.H.); f_yuanhua@163.com (Y.F.); 18847149074@163.com (K.J.)

[2] College of Physics and Electronic Information, Inner Mongolia Normal University, Hohhot 010022, China

* Correspondence: 201722020930@std.uestc.edu.cn (F.Y.); yalvhe2009@163.com (D.Y.)

Received: 17 March 2020; Accepted: 29 April 2020; Published: 2 May 2020

Abstract: Upgrading ordinary streetlights to smart streetlights to help monitor traffic flow is a low-cost and pragmatic option for cities. Fine-grained classification of vehicles in the sight of smart streetlights is essential for intelligent transportation and smart cities. In order to improve the classification accuracy of distant cars, we propose a reformed YOLOv3 (You Only Look Once, version 3) algorithm to realize the detection of various types of automobiles, such as SUVs, sedans, taxis, commercial vehicles, small commercial vehicles, vans, buses, trucks and pickup trucks. Based on the dataset UA-DETRAC-LITE, manually labeled data is added to improve the data balance. First, data optimization for the vehicle target is performed to improve the generalization ability and position regression loss function of the model. The experimental results show that, within the range of 67 m, and through scale optimization (i.e., by introducing multi-scale training and anchor clustering), the classification accuracies of trucks and pickup trucks are raised by 26.98% and 16.54%, respectively, and the overall accuracy is increased by 8%. Secondly, label smoothing and mixup optimization is also performed to improve the generalization ability of the model. Compared with the original YOLO algorithm, the accuracy of the proposed algorithm is improved by 16.01%. By combining the optimization of the position regression loss function of GIOU (Generalized Intersection Over Union), the overall system accuracy can reach 92.7%, which improves the performance by 21.28% compared with the original YOLOv3 algorithm.

Keywords: smart streetlight; YOLOv3; multi-scale training; anchor clustering; label smoothing; mixup; IOU; GIOU; fine-grained classification of automobile

1. Introduction

With the rapid development of the modern transportation industry, a common scenario in an urban transportation network is that certain sections of the road may experience severe traffic congestion, whereas the traffic flow on nearby sections is relatively smooth. Therefore, by knowing the traffic conditions of each road in real time, the intelligent transportation system can help drivers choose a reasonable driving route, which is also an effective approach to solve urban traffic congestion [1–4]. The image-processing-based traffic length detection system combines image processing with various traffic information technologies and has the advantages of wide application range, high measurement precision, excellent real-time performance and direct upgrade based on the existing monitoring system. Therefore, it is an important technical component for obtaining modern intelligent traffic information.

During the last decade, researchers have conducted extensive studies. In 2014, Ross Girshick et al. [5] proposed a visual inspection method named R-CNN (Regions with Convolutional Neural Networks features). This method first divides the inspected image into thousands of different regions, then extracts

deep features of these regions, and finally adopts Softmax for classification. On dataset VOC2012, the mean average precision (mAP) detected by R-CNN could reach 53.3%, which is higher than the precision of previous methods. However, it has the disadvantages of long detection time and difficult engineering application. In 2016, Liu et al. [6] proposed the SSD (Single-Shot Multi-Box Detector) method, which improves the deep convolutional neural network of VGG-16 (Visual Geometry Group Network) [7] and the target detection method of YOLO (You Only Look Once) [8], extracts multi-scale feature maps, and directly outputs the location and class of the detected target. On dataset VOC2007, this method could achieve a mAP of 74.3% and a speed of 59 FPS. In 2018, Chu et al. [9] reported a deep convolutional neural network (CNN) based on multiple tasks and the vehicle detection method for an automatic pilot based on region-of-interest voting. In this method, the multi-task targets of CNN include area overlapping, subclass, regression of detection frame and region of proposals (ROI) regional training, and the vehicle is detected by fully considering the influence of the neighborhood ROI. According to the validation of vehicle datasets KITTI [10] and VOC [11], it presents a much better performance than most similar algorithms. In 2019, Chang-Yu Cao et al. used the YOLO-UA model to improve the detection precision of vehicles under complicated weather conditions [12].

In the various studies above, all researchers have ignored a problem, which is that, in actual street lighting, we need to extend the effective detection distance of the video as much as possible to reduce the number of intelligent streetlights, and thus reduce the cost of the entire system. However, in the case of long-distance vision, the accuracy of the current algorithm to achieve fine-grained car classification is not ideal. In order to address this issue, in this article, we primarily make the following three contributions:

1. In order to realize high-precision and large-scale traffic situation detection, we propose a distributed system based on smart streetlights.
2. Based on the dataset UA-DETRAC, we add the local manually labeled data images in the sight of smart streetlights, establish the classification dataset for SUVs, sedans, taxis, commercial vehicles, small commercial vehicles, vans, buses, trucks, and pickup trucks, and build the dataset UA-DETRAC-LITE-NEW.
3. We optimize the YOLOv3 algorithm in various respects to improve the detection accuracy for distant cars in the sight of smart lights and then combine it with multi-scale training and anchor clustering methods to improve the accuracy of automobile target detection. In addition, we apply label smoothing and mixup approaches to increase the generalization ability of the model and adopt optimized position regression loss functions of IOU (Intersection Over Union) and GIOU to increase the system accuracy. Each step of the improvement is experimentally verified.

2. Smart Streetlights and Experimental Datasets

2.1. Smart Streetlights

As shown in Figure 1, the camera and computing unit are integrated into the ready-made streetlight to form an intelligent streetlight. After the camera in the intelligent streetlight identifies the type and number of vehicles, the traffic data collected by the computing unit is transmitted to the cloud strategy platform through the wireless communication, power line carrier communication or visible light communication system. According to the information from smart streetlights, the cloud strategy platform could provide a variety of application services, such as traffic guidance and emergency rescue. Because the streetlight system is a complete infrastructure, cable layout and routing have cost advantages. In addition, streetlights can provide sufficient brightness for video and image detection to ensure the accuracy of detection.

Furthermore, the number of nine kinds of vehicles in the field of vision of each streetlight can be summed up, and the ratio of the road area occupied can be accurately calculated according to the area of the road projected by each type of vehicle. This index is very important for illegal parking

assessment, traffic diversion and traffic light strategy formulation, which is also difficult to achieve by other technologies (satellite remote sensing, microwave radar).

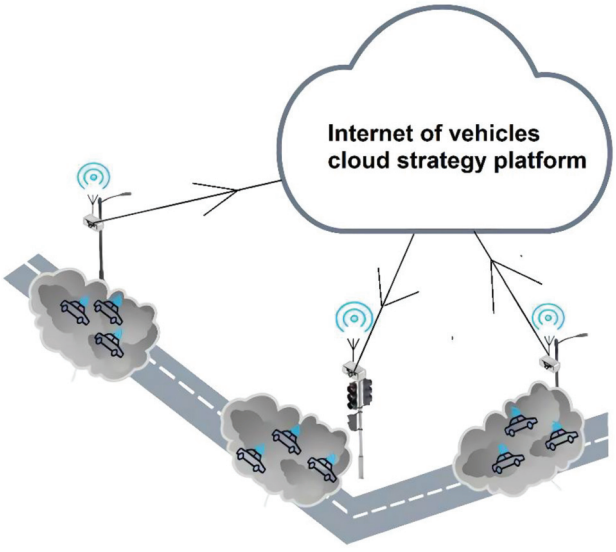


Figure 1. Smart streetlights in the Internet of Vehicles.

2.2. Experimental Dataset

The target detection in this research focuses on the fine-grained detection of vehicles, but there is no public fine-grained vehicle detection dataset at present. Therefore, in this paper, the dataset used in the experiment is obtained by mixing the manually labeled dataset with the public dataset.

Part of the manually labeled dataset was obtained by the camera installed in a pedestrian bridge in Chengdu. This dataset includes the front and back images of vehicles. The vehicle models include sedans, buses, taxis, trucks, SUVs and pick-ups; the involved scenarios include: daytime, night, fine day and cloudy day. In total, 1231 images were obtained, and the resolution was adjusted to 540×960. The other part was obtained through network crawling: 387 images were obtained in total, and the resolution was adjusted to 540×960. Finally, the vehicle location was labeled using LabelImage, and the labeled data were given in the format of VOC2007. Figure 2 presents the frame image decomposed from the vehicle video collected from a smart streetlight in Chengdu. The detection range is 67 m.

The common public vehicle detection datasets include UA-DETRAC [13–15], KITTI [16], etc. Because the images in the dataset UA-DETRAC were all collected from a pedestrian bridge, the image collection angle is closest to the angle of the image collected by intelligent streetlights in the IoV (Internet of Vehicles), so the dataset UA-DETRAC is used in this paper. This dataset consists of ten-hour video continuously collected by a Cannon EOS 550 camera at 24 different locations in Beijing and Tianjin, China. The video was collected at the speed of 25 frames per second (fps); then, each frame of video was decomposed to images using software, and each image had a resolution of 960×540 pixels. In this dataset, 8250 vehicles were manually labeled, and there were 1.21 million label boxes in total. The dataset includes the labels of four major classes and thirteen subclasses, such as SUV, sedan, taxi, commercial vehicle, small commercial vehicle, large van, hatchback, bus, police car, medium van, truck, pick-up and platform truck. The shooting scenarios include cloudy weather, night, sunny days and rainy days, as shown in Figure 3.

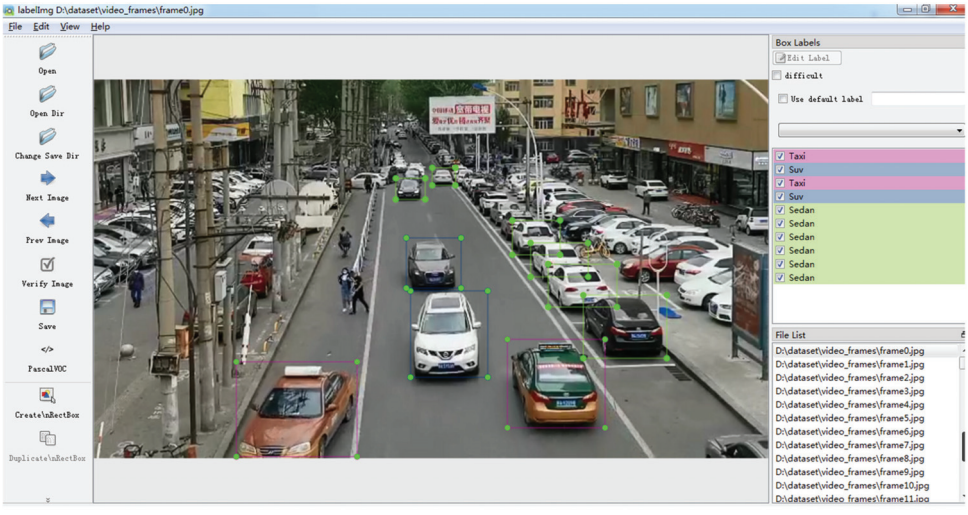


Figure 2. Labelling of vehicles with image label software.

Figure 3. Images of dataset UA-DETRAC.

Because, among the subclasses of vehicles in the dataset UA-DETRAC, the distribution of various types of vehicles is not even in number, the numbers of police cars and platform trucks are particularly low, and there are only 300 labeled police cars in the entire dataset. Furthermore, there are not many police cars on real roads, and manual shooting is inconvenient. Considering the above factors, in this paper, dataset UA-DETRAC only consists of the following nine subclasses of vehicles: SUVs, sedans, taxis, commercial vehicles, small commercial vehicles, vans, buses, trucks and pick-ups, in which the class of vans combines the two subclasses of large vans and medium vans, and trucks combine platform trucks and original trucks. Because the computing resources are not particularly sufficient, in this paper, we select 13,516 images from the 82,118 images in dataset UA-DETRAC, and mix them with 1618 manually labeled images to form the dataset used in this paper: UA-DETRAC-LITE-NEW. Among these, the 2053 manually labeled images are mainly used to supplement the insufficient number of pick-ups and trucks in the original dataset UA-DETRAC. In this paper, we select 12,455 images from

the 15,569 in the dataset UA-DETRAC-LITE-NEW as the training dataset, and we select 3114 images captured in the sight of streetlights as the test dataset.

3. Methodology

3.1. Introduction of YOLOv3

In 2018, Joseph Redmon et al. [17] proposed YOLOv3, and, compared with the last generation of the YOLO algorithm, this method has increased detection prevision and strengthened identification ability for small objects. In YOLOv3, a new backbone network called Darknet-53 is used. By referring to the residual network (ResNet [18]), this network has set shortcut connections between some layers, so that information can be directly transmitted from shallow layers to deep layers, so as to maintain the integrity of information. In addition, the idea, similar to FPN [19], is to conduct multi-scale prediction of bounding boxes, which can prevent missing of small objects.

In YOLOv3, the L2 distance is used to calculate the loss by default. Loss consists of three parts: position loss, confidence loss and class loss. The loss computation is as shown in Formula (1):

$$loss = coorLoss + confidenceLoss + classError \tag{1}$$

where YOLOv3 is used to calculate loss. Pay attention to the following details:

(1) During calculation of position loss, the L2 distance does not have scale invariability relative to the bounding box, so the loss of a small bounding box should be distinguished from that of a big bounding box, which is more reasonable. The author multiplies factor α ($a = 2 - truth_w \times truth_h$) with the L2 distance loss of position to weaken the negative influence of scale on position loss. $truth_w$ and $truth_h$ are the width and height of normalized true-value bounding boxes. Factor α is smaller when encountering large-scale targets and bigger when encountering small-scale targets, and in this way, the loss of small bounding boxes can be improved.

(2) During calculation of loss, no matter whether it is position loss or other loss, the true value should correspond to the predicted values one by one to realize calculation. However, in reality, the number of targets in each image is not fixed. When the output image has a scale of 416, assuming m image targets participate in the training, while the YOLOv3 model will predict 10,647 targets, they cannot correspond to each other one by one in this case. Therefore, YOLOv3 employs a matching mechanism to reconstruct m targets in true-value tag into the data in the format of 10,647 targets. m targets are placed in their proper positions in the 10,647 constructed data, and, at this moment, the calculation and training process is completed.

(3) In the YOLOv3 algorithm, only the lattice with the scale which best matches the anchor is used to predict the target. However, under normal circumstances, it is also highly possible to use the lattices near this lattice to predict this target. This situation is normal, which does not require punishing this lattice. However, when the lattice far away from this lattice is also used to predict this target, this situation is abnormal, which requires punishment. Therefore, in YOLOv3, the predicted value and corresponding position of constructed real-value tag are used to calculate IoU, then the IoU is used to decide when punishment is required and when it is not. In general, a hyper-parameter will be defined in the algorithm: ignore thresh. When the IoU of a lattice is bigger than the ignore thresh, this lattice is regarded as being close to the real lattice, and it is normal that it has high confidence, which does not require punishment; when the IoU of a lattice is smaller than the ignore thresh, this lattice is regarded as being far from this lattice, and it is abnormal when it has high confidence, which requires punishment.

(4) In (2), the m targets are reconstructed into the data in the format of 10,647 targets, and all the parts without targets are 0. In (3), it is mentioned that the IoU of the corresponding position of predicted value and real-value tag need to be calculated, while in the real-value tags without targets, all data are 0, and in this case, the confidence in the predicted value and corresponding

real-value tags cannot be calculated. Therefore, in YOLOv3, the lattices of all real-value tags and current predicted value are used to calculate the IoU, and the highest one is used as the final IoU of this lattice to participate in the subsequent operation.

To sum up, the loss function of YOLOv3 is as shown in the following Formula (2):

$$loss = loss_{scale1} + loss_{scale2} + loss_{scale3} \tag{2}$$

Because the loss functions under various scales are very similar, this paper only provides the loss function of one scale. The loss function of Scale I is as shown in Formula (3):

$$\begin{aligned} loss_{scale1} = & \sum_{i=0}^{S_1^2} \sum_{j=0}^{B} a_{ij} I_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \\ & \sum_{i=0}^{S_1^2} \sum_{j=0}^{B} a_{ij} I_{ij}^{obj} \left[(w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i \right)^2 \right] + \\ & \sum_{i=0}^{S_1^2} \sum_{j=0}^{B} I_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \beta_{ij} I_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 + \\ & \sum_{i=0}^{S_1^2} \sum_{j=0}^{B} I_{ij}^{obj} \sum_{c \in classes} \left(p_{ij}(c) - \hat{p}_{ij}(c) \right)^2 \end{aligned} \tag{3}$$

where S_1^2 refers to the number of lattices in the first scale and B represents the number of anchors under each scale, or the maximum number of lattices predicted by each lattice under each scale. In YOLOv3, B is set at 3; a_{ij} refers to the scale factor corresponding to the jth prediction of the ith bounding box; as mentioned above, each lattice will predict three bounding boxes, I_{ij}^{obj} represents that the object is in the jth prediction box of the ith lattice, while I_{ij}^{noobj} indicates that the object is not in the jth prediction box of the ith lattice; (x, y, w, h, C, p) is the true value, and $(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{C}, \hat{p})$ is the predicted value of the model.

The third line represents the confidence error. The confidence error consists of two parts: one type is the lattice with true value and target, and it only requires calculating the error between the corresponding positions of predicted value and true value in a normal way; the other type is the lattice with the true value but without the target; in this case, the predicted value of the confidence in corresponding position might be too high, and we need to determine whether such a high predicted value of confidence is normal. As mentioned above, current lattices with high predicted values and the IoUs of all real-value bounding boxes need to be calculated, so as to determine whether the lattice with high predicted confidence is close to a certain real-value bounding box. If it is close, it does not need to calculate the loss of the current lattice; otherwise, it requires calculating the loss. In this process, β_{ij} indicates whether it requires calculating the loss of current lattice. If the ignore thresh is T, the computation formula of β_{ij} is as shown in Formula (4).

$$\beta_{\mathrm{ij}} = \begin{cases} 0 & IoU(pred, true) > \mathrm{T} \\ 1 & \text{else} \end{cases} \tag{4}$$

The fourth line shows the class error. Because the calculation of class error is only meaningful when there is a target, this line also uses I_{ij}^{obj} to filter out the lattices without a target, and it only calculates the loss between lattices with targets.

To sum up, YOLOv3 still uses the L2 distance massively during loss calculation. In position calculation, although the scale factor α is used to balance the influence of object loss with different scales, this influence still cannot be completely eliminated, which will cause negative influence on

the optimization of the network. Therefore, it will reduce the regression precision of object detection position to a certain degree.

3.2. Scale Optimization for Vehicle Target Detection Scenario

This paper focuses on studying vehicle target detection under the intelligent streetlight in the IoV, while in the target detection problem, different scenarios involve different bounding boxes. For example, in face detection, the finally predicted bounding box has a shape close to square with a high probability. In pedestrian detection, the predicted bounding box is more like a long and thin rectangular box; therefore, there is prior information for each scenario, and such prior information can be maximally utilized to better optimize the model. In this section, the clustering method will be used to determine the optimal prior box for the vehicle detection scenario. Furthermore, in the vehicle detection problem, the vehicle might be far or near, so the multi-scale training method can be employed to improve the model performance, and we call this method scale optimization.

3.2.1. Multi-Scale Training

In the target detection problem, the size of the input image has a significant influence on the precision of the model. Although a bigger input image may indicate better precision performance of the model, it also means a higher computation cost and slower inference speed. In practical applications, it generally requires achieving a balance between the model precision and inference speed, so the size of input images should not be too big.

After the image is input into the basic network, it will generally generate a feature map ten times smaller than the input image, which makes it difficult for the network to capture the small objects in a small feature map. Therefore, by imputing bigger and more images to train the model, the model's robustness against object size can be improved to a certain degree. As a result, the model can learn more common features of the target rather than specific details of target, and the model's generalization ability can be improved. During the inference stage, small input images are still used for inference, which does not incur any loss in inference speed. However, more optimal model performance can be obtained through multi-scale training, so when the same size of input image is used during inference, the model precision obtained through multi-scale training is higher.

This paper has defined the scale rule as shown in Formula (5) during training, in which n is a constant number. For each batch, the image of a scale will be randomly selected from the scale rule and inputted into the neural network for training and learning.

$$S(n) = 320 + 32n \; 0 \leq n \leq 9 \tag{5}$$

3.2.2. Anchor Clustering

Generally speaking, the anchor-based target detection algorithm needs to set the anchor to predict the bounding box of target. However, if the selection of anchor is not suitable, more iterative trainings of neutral network are needed for the neural network to better predict the bounding box. In general, different anchors should be set for different application scenarios, so the setting of the anchor can also be regarded as a hyper-parameter. In this paper, the k-means clustering algorithm is adopted to find a group of anchors suitable for the scenarios discussed in this paper.

In the k-means algorithm, at the beginning, the user needs to specify k initial centroids. These k initial centroids will scan all data and divide the points with closest "distance" to themselves into clustering classes, and, in this way, all data will be divided into k classes. Next, the centroid of each cluster will be updated, then all data are scanned once again for classification, and the iteration is repeated until the algorithm converges. The k-means algorithm involves the following steps:

(1) Select k initial centroids (as the k initial clusters);
(2) For each sample point, obtain the nearest centroid based on calculation, and label its class as the corresponding cluster of this centroid;

(3) Recalculate the centroids corresponding to k clusters;
(4) The algorithm will end when the centroids of k clusters do not change anymore; otherwise, jump to Step 2.

Under the Euclidean distance, the sum of the squared error (SSE) is generally used as the objective function of clustering, and this index is often used as the index to measure the quality of clustering result. Its expression is as shown in Formula (6).

$$\text{Loss} = \text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} distance(x, c_i) \tag{6}$$

where c_i presents the centroid of the ith cluster; and $distance(x, c_i)$ represents the "distance" from x to c_i. The reason why the word distance mentioned is in quotes is because, generally speaking, the common distance is the Euclidean distance, but under the service scenario required in this paper, if the Euclidean distance is used for clustering of bounding boxes, a big bounding box will generate more errors than a small bounding box. Therefore, we hope that the distance required in clustering is irrelevant to the size of the bounding box, and the distance formula applicable to the scenario of this paper is as shown in Formula (13).

$$distance(x, c_i) = 1 - IoU(x, c_i) \tag{7}$$

In this paper, after running the clustering algorithm in dataset UA-DETRAC-LITE-NEW multiple times, an optimal group of anchors is obtained: (8,9), (10,23), (19,15), (23,33), (40,25), (54,40), (101,80), (139,145), (253,224). The comparison between the anchor set by the original YOLO algorithm and the anchor with optimal performance obtained by our clustering method in the vehicle detection environment is shown in Figure 4. In this paper, the input image sizes for all target detections are set at 416×416, so the comparison in the diagram below is also based on the comparison of original image size. In addition, nine anchors are drawn, with the center of the image (i.e., coordinate position (208,208)) as the center. Through comparison, it can be found that the original YOLO algorithm focuses more on the generality of algorithm, different sizes of anchors have uniform distribution in number, and the algorithm emphasizes universality; the anchors obtained with our clustering algorithm are generally smaller than the anchors obtained with the original YOLO algorithm, and most anchors have small sizes. This also suits the intelligent streetlight environment of IoV. Most vehicles in the image have small sizes.

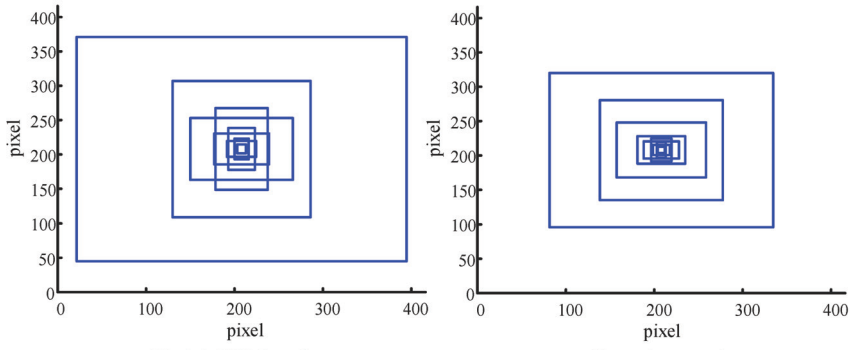


Figure 4. Comparison of anchor before and after clustering.

After completing the anchor data clustering, when there is prior box information closer to the real target, the YOLO algorithm can return to the target position with less iterations, which can reduce the model training times to a certain extent.

3.3. Improvement of the Model's Generalization Ability

The model's generalization ability refers to whether the model can have good performance when facing unknown data. We train the model to make the model extract and learn the characteristics of real object, so that the model can be used for real unknown data in the future. From this perspective, the model's generalization ability is particularly important. On the one hand, mixup is used in target detection in this section to improve model's generalization ability when facing unknown data (test dataset), thus further improving the model's average precision; on the other hand, the label smoothing method is introduced in this section to reduce the model's "confidence" about objects, thus improving the model's generalization ability.

3.3.1. Improvement Based on Label Smoothing

The training dataset used in this paper is generally from dataset UA-DETRAC. Dataset UA-DETRAC was collected from real roads, while the vehicles on the roads are generally sedans and SUVs, and the number of trucks, pick-ups and commercial vehicles is significantly smaller than that of sedans. Although some of these vehicles, in small numbers, were also manually collected and labeled in this paper, which are also added to the training dataset, this still cannot balance the numbers of various classes of vehicles. For vehicle classes with small numbers in the dataset, the deep neutral network might not be able to learn features with strong generalization ability, and, as a result, the final model obtained by training has poor generalization ability. In this paper, the label smoothing technique is adopted to maximally prevent the problem of the weak generalization ability of the model.

When the target detection algorithm proposed in this paper is finally used to predict the target class, the one-hot coding method is adopted. For example, if the sample is observed as belonging to a certain class, the position of the corresponding class of vector has the value of 1, or otherwise, it is 0. Such a coding method may cause over-fitting when the above samples have uneven distribution, because limited samples are used for training in general, which cannot cover all possible situations. In an extreme example, when the training data include 200,000 sedans but only 3000 pick-ups, the occurrence probability of a sedan is around 98.52%, while that of a pick-up is only about 1.48%. In this case, with continuous training, the neutral network tends to predict the class of sedan, and the probability of the predicted class being a sedan is 100% or very close to 100%, while the class of pick-ups is ignored. At this time, the model is "overconfident" in the prediction of sedans.

The Label Smoothing Regularization (LSR) method aims to alleviate the over-fitting problem which tends to occur because the true-value label is not "soft" enough. It is a constraint method which adds some noise to the true value of the class to constrain the model and lower the over-fitting degree of the trained model.

The LSR formula is as shown in Formula (8), in which q_i represents the one-hot coding after smoothing; $\varepsilon \in [0, 1]$, ε is a small hyper-parameter, which represents the smoothing degree or how much noise has been added to the real value, and bigger ε indicates more noise has been added to real value; K refers to the number of classes; the true value of the class is generally stored in the ID of the class before being transformed to the one-hot code, and in the formula, y is the ID of the classes true value.

$$q_i = \begin{cases} 1 - \varepsilon \text{ i} = y \\ \frac{\varepsilon}{\text{K}-1} \text{ else} \end{cases} \tag{8}$$

In essence, LSR takes the value of true-value class in the original one-hot code from ε, and then evenly distributes it among other classes, so that the final distribution of true-value label is no longer distributed on its true-value position. As shown in Figure 5, $\varepsilon = 0$ on the left diagram, which means no LSR; the right diagram shows the case of $\varepsilon = 0.1$.

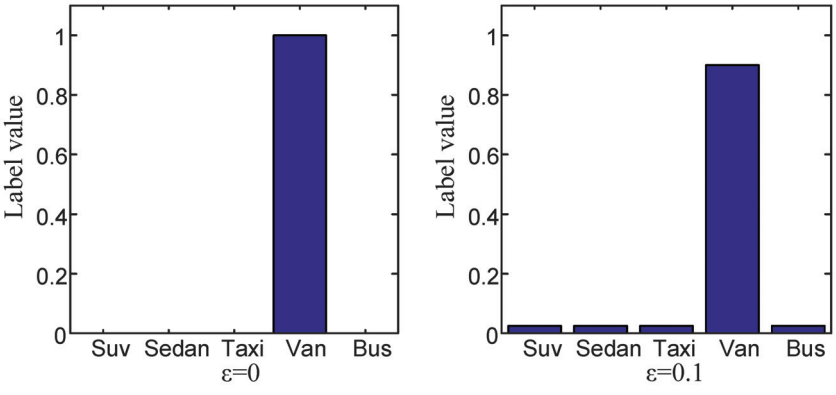


Figure 5. Schematic diagram of Label Smoothing Regularization.

In the above diagram, only five classes are chosen to present the change of label value before and after label smoothing. In the meantime, ε is set as 0.1 to make it easier to see the change. In practical training, ε is set as 0.01. It can be seen that the label value has a broader distribution after label smoothing. As a result, the model is finally not that "confident", and the model's generalization ability is improved.

3.3.2. Improvement Based on Mixup

In the field of target detection, the data enhancement technique will frequently be used. When the dataset does not contain particularly sufficient data, the deep model is unable to learn good features. In addition, in most datasets, various types of samples are not balanced in number, which will result in poor ability of the model in predicting samples with small numbers. The above problem can be alleviated through data enhancement, which can efficiently improve the network's performance and generalization ability. Data enhancement refers to utilizing certain image conversion methods (such as random cropping, random translation, angle rotation, mirror image flipping, etc.) based on existing datasets to generate some "new data" for the neural network.

Hongyi Zhang et al. proposed the mixup method, which can be used to effectively improve the performance and generalization ability of the object classification model [20]. The mixup method is a data enhancement method irrelevant to data, which is also simple. It is used to generate new samples with the method shown in Formulas (9) and (10).

$$x = \lambda x_i + (1 - \lambda) x_j \tag{9}$$

$$y = \lambda y_i + (1 - \lambda) y_j \tag{10}$$

The original author applied mixup in the classification task, in which x_i and x_j are two random image samples, while y_i and y_j are their corresponding class labels. $\lambda \in (0,1)$, and λ follow a Beta distribution: $\lambda \in$ Beta (α,α), $\alpha \in (0,\infty)$. The mixup method linearly integrates existing samples and generates new samples, which makes the model more stable.

In this paper, the data enhancement technique of mixup is introduced into vehicle target detection. In the dataset used in this paper, all images have 540×960 pixels. In image integration, similar to the above Formulas (9) and (10), a new sample x can be obtained through superposition of the corresponding pixels of randomly selected images x_i and x_j according to linear coefficients λ and $(1 - \lambda)$. Tags y_i and y_j of images x_i and x_j are kept, and tag y is obtained corresponding to a new sample x. A schematic diagram of using mixup to realize data enhancement in the vehicle target detection problem is shown in Figure 6.

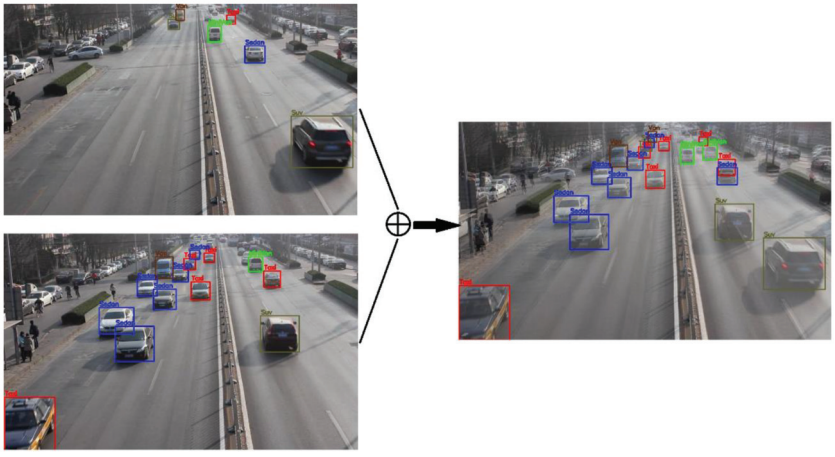


Figure 6. Schematic diagram of Mixup.

3.4. Improvement Based on the Position Regression Loss Function

In the YOLOv3 algorithm, the predicted position (x,y,w,h) of a bounding box is obtained through direct regression, because one bounding box is regressed, while the bounding box may have big size or small size. During calculation of position loss, if the size of the bounding box is not considered, it will cause inaccurate regression of the bounding box, slow training and convergence speed and other problems. YOLOv3 uses the scale factor to reduce the difference in position loss between bounding boxes of different sizes. However, because the L2 distance does not possess scale invariability, this approach will only reduce part of the difference between big and small bounding boxes in loss. Therefore, we believe a distance evaluation index with scale invariability is needed to improve the loss function. In this section, IoU and GIOU will be used as the indices to measure the distance between bounding boxes and improve the loss function of YOLOv3.

3.4.1. Improvement of Loss Function Based on IoU

Because IoU has invariability with the change of scale, it means it is not affected by the sizes of two objects. Therefore, in this section, IoU will be used to improve the loss function of the YOLO algorithm. Moran Ju [21] once proposed using this method to improve the algorithm's performance, and we adopt this method in our system.

IoU is within the range of 0~1. When IoU is 0, it means the two bounding boxes are completely disjoint; when IoU is 1, it indicates the two bounding boxes completely overlap. Therefore, Formula (11) can be used to describe the distance between two bounding boxes, and this distance can be directly used in the position loss function of the YOLO algorithm.

$$d(b_0, b_1) = 1 - IoU(b_0, b_1) \tag{11}$$

IoU satisfies scale invariability, which is not affected by the scales of two objects, and it can also be used to describe the distance between two objects. During regression of coordinates, YOLOv3 can use Formula (11) as the distance to directly describe the difference between the predicted bounding box and the real-value bounding box. The size of this difference can be used in the position regression loss function and participate into the network optimization.

3.4.2. Improvement of Loss Function Based on GIOU

IoU has scale invariability and can be used as the index to measure the distance between objects; however, when IoU is used as distance, if the two objects are completely disjoint, no matter whether

they are far from or close to each other, the distance value is fixed, which is 1. At this time, the gradient cannot be returned, and the model may be unable to learn the position of part of true-value bounding box during training, which has lowered the final precision of the model. In this section, GIOU will be used as the distance to improve the YOLO position loss.

GIOU is a new index used to measure the overlapping degree between two bounding boxes on the basis of IoU, which considers the non-overlapping area not considered by IoU, and it can reflect the overlapping method between A and B. Assume the coordinate of the predicted bounding box is $B^p = \left(x_1^p, y_1^p, x_2^p, y_2^p\right)$, and the coordinate of true-value bounding box is $B^g = \left(x_1^g, y_1^g, x_2^g, y_2^g\right)$. GIOU and the computation process using GIOU as the loss function is as follows:

(1) In order to ensure the coordinates of the predicted box satisfy $x_2^p > x_1^p$ and $y_2^p > y_1^p$, conduct data manipulation as in Formulas (12)–(15):

$$\hat{x}_1^p = \min\left(x_1^p, x_2^p\right) \tag{12}$$

$$\hat{x}_2^p = \max\left(x_1^p, x_2^p\right) \tag{13}$$

$$\hat{y}_1^p = \min\left(y_1^p, y_2^p\right) \tag{14}$$

$$\hat{y}_2^p = \max\left(y_1^p, y_2^p\right) \tag{15}$$

(2) Calculate the area of B^g:

$$A^g = \left(x_2^g - x_1^g\right) \times \left(y_2^g - y_1^g\right) \tag{16}$$

(3) Calculate the area of B^p:

$$A^p = \left(\hat{x}_2^p - \hat{x}_1^p\right) \times \left(\hat{y}_2^p - \hat{y}_1^p\right) \tag{17}$$

(4) Calculate the intersection between B^p and B^g:

$$x_1^I = \max\left(\hat{x}_1^p, x_1^g\right) \tag{18}$$

$$x_2^I = \min\left(\hat{x}_2^p, x_2^g\right) \tag{19}$$

$$y_1^I = \max\left(\hat{y}_1^p, y_1^g\right) \tag{20}$$

$$y_2^I = \min\left(\hat{y}_2^p, y_2^g\right) \tag{21}$$

$$I = \begin{cases} \left(x_2^I - x_1^I\right) \times \left(y_2^I - y_1^I\right) & x_2^I > x_1^I, y_2^I > y_1^I \\ 0 & \text{else} \end{cases} \tag{22}$$

(5) Find the smallest rectangular box B^c enclosing B^p and B^g:

$$x_1^c = \min\left(\hat{x}_1^p, x_1^g\right) \tag{23}$$

$$y_1^c = \min\left(\hat{y}_1^p, y_1^g\right) \tag{24}$$

$$y_2^c = \max\left(\hat{y}_2^p, y_2^g\right) \tag{25}$$

(6) Calculate the area of B^c:

$$A^c = \left(x_2^c - x_1^c\right) \times \left(y_2^c - y_1^c\right) \tag{26}$$

(7) Calculate IoU and GIOU:

$$IoU = \frac{I}{A^p + A^g - I} \tag{27}$$

$$GIoU = IoU - \frac{A^c - (A^p + A^g - I)}{A^c} \tag{28}$$

(8) Obtain the loss function based on GIOU:

$$\mathcal{L}_{GIoU} = 1 - GIoU \tag{29}$$

4. Experiment and Results

4.1. Evaluation Metrics

In this paper, we adopt the broadly used evaluation model indices with concentrated VOC data: average precision (AP) and mean average precision (mAP). In addition, the parameters of confusion matrix, PR curve, model frames per second (FPS), etc. are also used to further evaluate the model.

See Formulas (30) and (31) for the calculation of AP and mAP.

$$AP = \int_0^1 p(r)dr \tag{30}$$

$$mAP = \frac{\sum_{i=1}^{N} AP(i)}{N} \tag{31}$$

where $p(r)$ represents the curve function with precision (P) as the ordinate and recall (R) as the abscissa, hereafter referred to as the PR curve. The enclosed area by the PR curve and abscissa axis is AP; mAP refers to the mean average precision, which equals to the sum of the precisions of all classes divided by the class number N. In this paper, the 11-point computation method in VOC2007 is used for calculation of precision.

The indices of precision and recall were mentioned above, and, in order to introduce these two indices, we need to introduce other common indices. In general, the multi-classification task can be split into multiple binary classification tasks, and, in binary classification, the samples are generally divided into positive samples and negative samples. Therefore, for real value and predicted value, the following four situations may occur through combination: the true value is a positive sample, and the predicted value is also a positive sample, which is called true positive (TP); the true value is a positive sample, while the predicted value is a negative sample, which is called false negative (FN); the true value is a negative sample, while the predicted value is a positive sample, which is called false positive (FP); the true value is a negative sample, and the predicted value is also a negative sample, which is called true negative (TN). However, in a target detection task, the target is regarded as a positive sample, and the background is treated as a negative sample, so the background will not be detected in target detection task. Therefore, the two indices of TN and FN are generally not used. In this paper, indices which can better reflect the model performance will be built based on TP and FP.

See Formula (32) for the definition of precision; in general, see Formula (33) for the definition of recall. However, in the abovementioned target detection problem, FN and TN are generally not discussed. TP+FN refers to the number of positive samples, and the number of positive samples can be obtained from the true-value tag. Therefore, in this paper, Formula (34) is used to redefine recall, and it is not different from the original definition in meaning. In it, N_p refers to the number of positive samples; see Formula (35) for the definition of F_1 measurement.

$$P = \frac{TP}{TP + FP} \tag{32}$$

$$R = \frac{TP}{TP + FN} \tag{33}$$

$$R_{new} = \frac{TP}{N_p} \tag{34}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{35}$$

The confusion matrix is another index used to evaluate the model performance. Its row vector represents the true value, and its column vector represents the predicted value. Therefore, it is easy and intuitive to find which class is misclassified into another class from the confusion matrix. In addition, the frames per second (FPS) can also be used to measure the computation efficiency of algorithm.

4.2. Experimental Results

4.2.1. Improved Experimental Results Based on Results of Scale Optimization

In this section, dataset UA-DETRAC-LITE-NEW is used to conduct an experiment with the original YOLOv3 algorithm and the algorithm proposed in this section, respectively. In the experiment, the following data are preprocessed: randomly regulate the brightness, contrast, hue and saturation; by ensuring that there is bounding box, randomly crop the image; randomly turn the image; conduct other operations. During training, set the batch size at 4, and in order to realize faster convergence of the network, conduct continuous training based on the trained pre-training weight by the author of YOLOv3 on ImageNet. Use the gradient descent method with momentum to optimize the model. Set the momentum as 0.9 and the learning rate as 0.001.

The average precision (AP) of each class and the mean average precision (mAP) of all classes are obtained in the experiment. See Table 1 for specific data; a more intuitive diagram comparison is shown in Figure 7. The comparison diagram of the omission ratio is presented in Figure 8; the confusion matrix is shown in Table 2.

Table 1. Comparison of accuracy between the original YOLO algorithm and its improved algorithm.

Type	Original YOLO (%)	YOLO with Scale Optimization (%)
Suv	71.80	76.42
Sedan	88.55	89.41
Taxi	89.55	91.45
Van	82.15	84.67
Truck-Box	76.94	84.16
Bus	82.94	90.94
MiniVan	74.48	78.08
Truck-Util	40.82	67.80
Pickup	35.54	52.08
mAP	71.42	79.45

Table 2. The confusion matrix of the improved YOLO algorithm (unit %).

Type	Suv	Sedan	Taxi	Van	Truck-Box	Bus	MiniVan	Truck-Util	Pickup
Suv	88.35	9.36	0.25	0.17	0	0	1.87	0	0
Sedan	0.21	99.48	0.22	0.05	0	0	0.05	0	0
Taxi	0	0.42	99.52	0	0	0	0.05	0	0
Van	0.45	0.57	0.11	95.69	0	0	3.17	0	0
Truck-Box	0	1.55	0	1.55	96.37	0	0	0.52	0
Bus	0	0.17	0	0	0	99.83	0	0	0
MiniVan	1.94	2.53	0.17	1.6	0	0	93.76	0	0
Truck-Util	0	0	0	1.27	0	0	0	98.73	0
Pickup	0	8.33	0	2.08	0	0	0	2.08	87.5

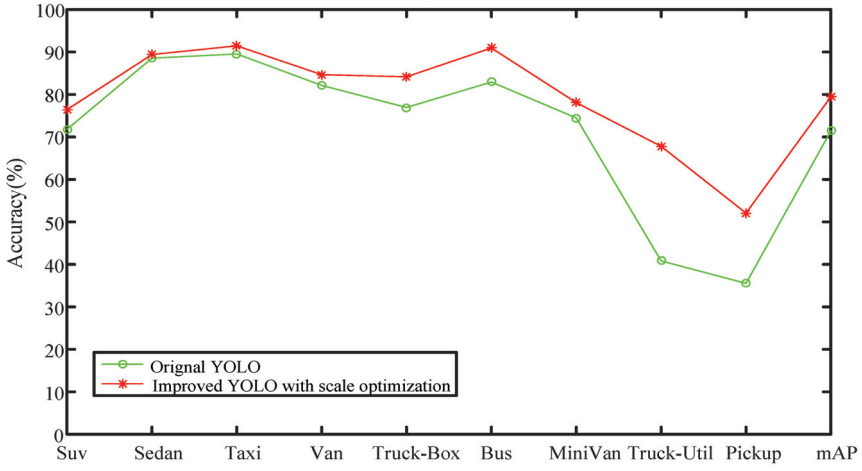


Figure 7. Comparison of accuracy between the original YOLO algorithm and its improved algorithm.

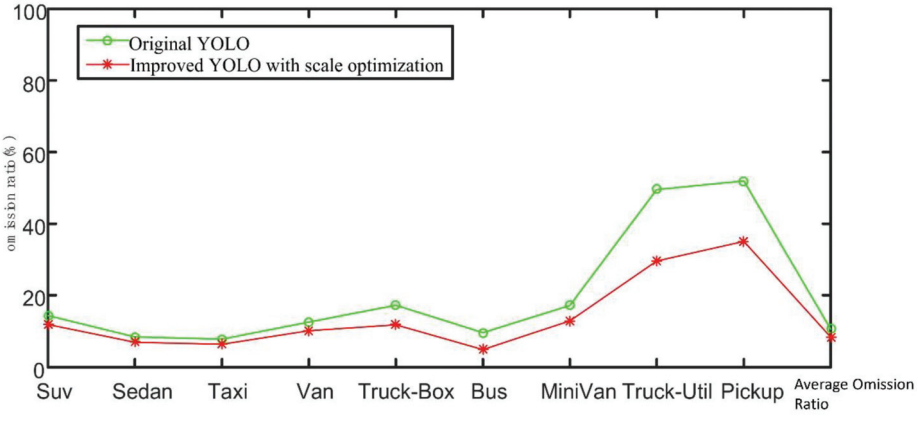


Figure 8. Comparison of omission ratio between the original YOLO algorithm and its improved algorithm.

According to the experimental results, it can be seen that the improved YOLO algorithm proposed in this section can significantly improve the detection performance for the two classes of trucks and pick-ups. In precision, the detection precisions for the classes of trucks and pick-ups are increased by 26.98% and 16.54%, respectively; through further analysis of the omission ratio, it can be seen that the main reason for the low precisions of the two classes of trucks and pick-ups is that they have low omission ratios, which further lowers the index of precision.

Although the detection precisions for trucks and pick-ups are significantly increased after improvement, their precisions are still lower than the average level. Next, the improved model proposed in this section will be continuously analyzed through confusion matrix. The row vector of confusion matrix represents the true value, while the column vector represents the predicted value. Therefore, the confusion matrix can only be used to analyze the distribution and accuracy of detected targets by the model. The diagonal from upper left to lower right represents the prediction accuracy for detected targets, and other positions indicate false class detections of detected targets. Through the confusion matrix, it can be seen that the detection accuracies for trucks and pick-ups by the improved model in this section are not low, so we can conclude that the main cause for the low precisions for these two classes is that they involve high omission ratios. Through the confusion matrix, we can also see that the model predicts the class of SUVs with true value of 9.36% as sedans and predicts pick-ups

with a value of 8.33% as sedans. This indicates that the model has not learned more specific features of SUVs and pick-ups yet.

By comparing the omission ratios of different algorithms, we can find that, after clustering, the model has a lower omission ratio. This can also be summed up as follows: after anchor clustering, the prior information of size closer to the size of real target is provided to the model, so that the model can regress more target positions faster, and the detection rate is increased accordingly.

4.2.2. Improved Experimental Results Based on Improving Model's Generalization Ability

In the experiments of this section, the classic two-stage target detection algorithm Faster-RCNN is used as the algorithm for comparison with the original YOLO algorithm and the improved YOLO algorithm proposed in this section. Among them, the Faster-RCNN algorithm adopts ResNet50 with stronger feature extraction ability as its backbone network. The improved algorithm 1 in this section requires the following experimental conditions: the anchor of the original YOLO algorithm, multi-scale training, using label smoothing, and using mixup for the first 10 epochs of training. The improved algorithm 2 in this section requires the following experimental conditions: the anchor for clustering of vehicle detection scenario in this paper, multi-scale training, using label smoothing, and using mixup for the first 10 epochs of training.

According to Table 3 and Figure 9, it can be seen that the various precisions of the two-stage Faster-RCNN algorithm are higher than the precisions of the YOLO algorithm. This is because Faster-RCNN adopts the RPN network to generate candidate area that may contain the target, so that more candidate areas can be obtained, and the reliability is also higher. Therefore, the Faster-RCNN algorithm has low omission ratio, and its precision is improved as a result. In addition, because dataset UA-DETRAC-LITE-NEW has labeled many pick-ups captured by camera from a long distance, where it is almost impossible to see the back of pick-up with human eyes, it is difficult to detect the class of the pick-up. By comparing the performances of the Faster-RCNN algorithm, the original YOLO algorithm, the improved YOLO algorithm 1 and the improved YOLO algorithm 2 in this section, it can be seen that the mixup and label smoothing methods can be used to effectively improve the model's generalization ability, almost without increasing the network computation overhead, so as to further improve the model's accuracy.

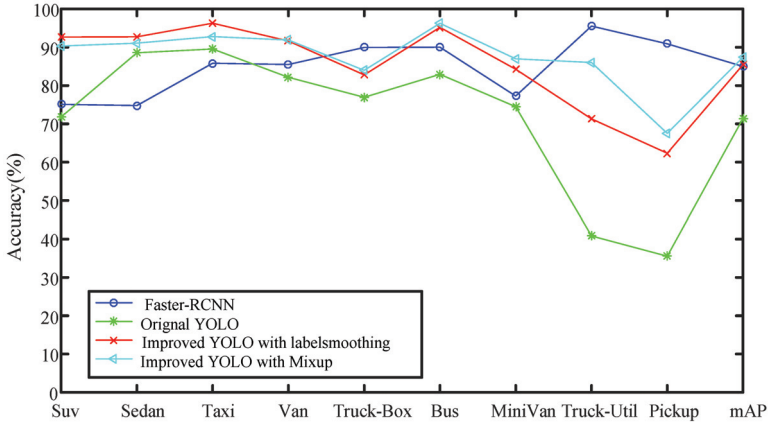


Figure 9. Comparison of Faster-RCNN, YOLO and improved YOLO in terms of precision.

Table 3. Comparison of Faster-RCNN, YOLO and improved YOLO in terms of precision.

Type	Faster-RCNN (%)	Original YOLO (%)	Label Smoothing (%)	Mixup (%)
Suv	75.10	71.80	92.66	90.33
Sedan	74.78	88.55	92.72	91.08
Taxi	85.83	89.55	96.27	92.79
Van	85.50	82.15	91.63	91.88
Truck-Box	89.95	76.94	82.78	83.98
Bus	90.01	82.94	95.19	96.25
MiniVan	77.27	74.48	84.25	86.98
Truck-Util	95.53	40.82	71.30	86.02
Pickup	90.89	35.54	62.34	67.53
mAP	84.99	71.42	85.46	87.43

According to Figure 10, it can be seen that the improved YOLO algorithms have lower omission ratios. Although the improved algorithm 2 has a higher omission ratio than the improved algorithm 1 in general, similar to the situation in the previous section, because the improved algorithm involves a higher FP ratio, the improved algorithm 2 has a higher average precision than the improved algorithm 1. Furthermore, the average omission ratio of the two improved algorithms is lower than that of the Faster-RCNN algorithm for comparison, and their omission ratio for most classes is lower than that of Faster-RCNN. Their omission ratio is higher than that of Faster-RCNN only for the two classes of pick-ups and trucks, which once more proves the effectiveness of the improved algorithm.

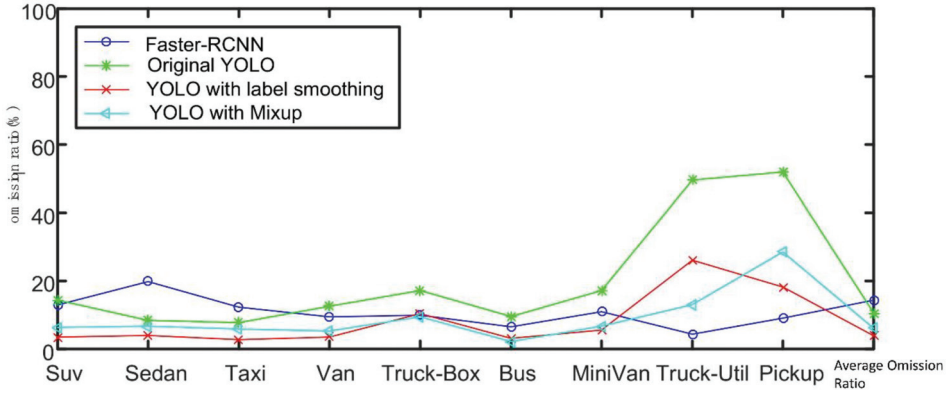


Figure 10. Comparison of Faster-RCNN, YOLO and improved YOLO in terms of omission ratio.

Table 4 shows the improved YOLO confusion matrix in this section. According to analysis of the confusion matrix, the probability of the model misclassifying the detected target into another class is very small. For the class of small commercial vehicles, the model falsely predicts 3.58% of small commercial vehicles as SUVs, and this is probably because the small commercial vehicles are close to SUV in height and size. In contrast, the model also falsely predicts 0.8% of SUVs as small commercial vehicles.

The main difference between improved algorithm 1 and improved algorithm 2 in this section is whether the anchor for clustering of the vehicle detection scenario proposed in this paper is used. It can be seen that when the anchor obtained through clustering is used to train the model, the performance can be improved by 1.98% compared to the performance of the improved algorithm using the original anchor. Once again, it proves that the model performance can be effectively improved for scenario clustering.

Table 4. Improved YOLO confusion matrix in this section (unit %).

Type	Suv	Sedan	Taxi	Van	Truck-Box	Bus	MiniVan	Truck-Util	Pickup
Suv	98.35	1.57	0	0	0	0	0.8	0	0
Sedan	0.23	99.7	0.2	0.2	0.3	0		0	0
Taxi	0.5	0.21	99.68	0	0	00	0.5	0	00
Van	0.43	1.18	0	97.33	0	0	1.7	0	0
Truck-Box	0	0.53	0	0	98.41	0	0	1.6	0
Bus	0	0	0	0	0	1	0	0	0
MiniVan	3.58	0.7	0	0.62	0	0	95.9	0	0
Truck-Util	0	0	0	0	0	0	0	1	0
Pickup	0	0	0	1.85	0	0	0	1.85	96.3

4.2.3. Improved Experimental Results Based on IoU Loss

In this section, we will verify the effects of replacing the original L2 loss with IoU loss in the position regression loss. The improved algorithm in this section involves the following experimental conditions: the anchor for clustering of the vehicle detection scenario described in this paper, multi-scale training, using label smoothing, and using mixup for the first 10 epochs of training. The situation of the loss function is as follows: using the IoU loss as position loss, using the L2 norm loss function as confidence loss, and using cross entropy as class loss.

According to Table 5 and Figure 11, when the position regression loss function is modified as the IoU loss function, the model performance is significantly improved. Except for trucks, its precision in predicting the other classes is higher than that of Faster-RCNN with high precision. The YOLOv3 algorithm has two main shortages: firstly, this algorithm is not very accurate in regression of object position; secondly, it has a low recall rate. For the problem of inaccurate regression of object position, the IoU loss in this section is irrelevant to scale, which can reduce the influence of different scales and improve the model's performance.

Table 5. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with IoU loss function in terms of precision.

Type	Faster-RCNN (%)	Original YOLO (%)	Improved YOLO with IoU (%)
Suv	75.10	71.80	90.29
Sedan	74.78	88.55	90.32
Taxi	85.83	89.55	90.32
Van	85.50	82.15	92.92
Truck-Box	89.95	76.94	95.23
Bus	90.01	82.94	95.90
MiniVan	77.27	74.48	89.84
Truck-Util	95.53	40.82	94.89
Pickup	90.89	35.54	93.73
mAP	84.99	71.42	92.60

In Figure 12, we compare the omission ratio and average omission ratio for each class of object. It can be seen that, after employing the IoU loss function, the omission ratio is lower than that of the original algorithm and the other algorithm in comparison. The output end of the YOLOv3 algorithm will predict 10,647 bounding boxes; some of these 10,647 bounding boxes will be filtered through the confidence threshold value, and the rest of the bounding boxes will continue to participate in the operation of the NMS algorithm. In the NMS algorithm, there is a threshold value of IoU. If the IoU of the selected bounding box and surrounding bonding boxes is higher than the threshold value, the surrounding bounding boxes will be deleted. From this perspective, if the model has predicted the correct target, but the location of predicted target is not very accurate, then this predicted target might be deleted in the NMS process due to inaccurate location, which will reduce the model's precision.

According to the above result, it can be seen that, after using the IoU loss, the precision is improved, the omission rate is reduced, and the model performance is further improved.

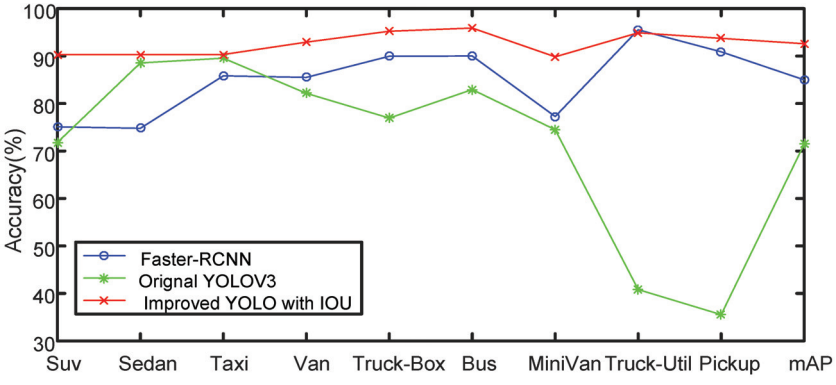


Figure 11. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with IoU loss function in terms of precision.

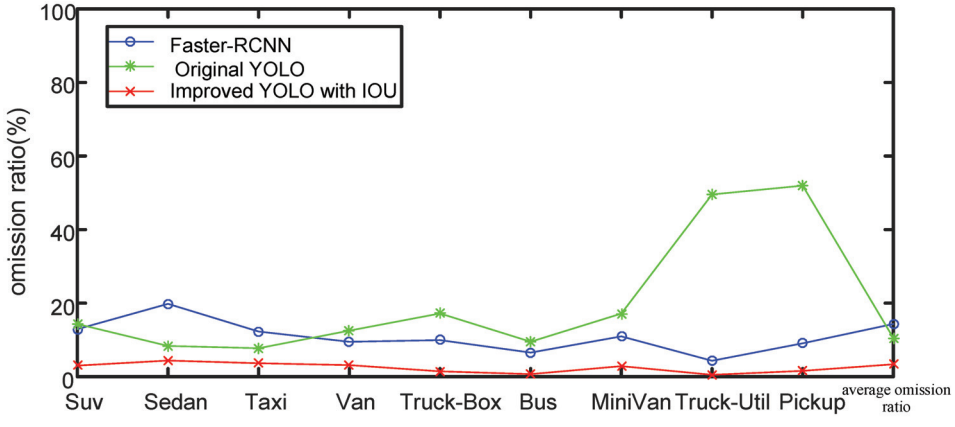


Figure 12. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with IoU loss function in terms of omission ratio.

4.2.4. Improved Experimental Results Based on GIOU Loss

In this section, we will verify the effects of replacing the original L2 loss with GIOU loss in the position regression loss. The improved algorithm 1 in this section requires the following experimental conditions: the anchor for clustering of the vehicle detection scenario described in this paper, multi-scale training, not using label smoothing, and using mixup for the first 10 epochs of training. The loss function is as follows: using the GIOU loss as position loss, using the L2 norm loss function as confidence loss, and using cross entropy as class loss. The improved algorithm 2 in this section requires the following experimental conditions: the anchor for clustering of the vehicle detection scenario described in this paper, multi-scale training, using label smoothing, and using mixup for the first 10 epochs of training. The loss function is as follows: using the GIOU loss as position loss, using the L2 norm loss function as confidence loss, and using cross entropy as class loss.

As shown in Table 6 and Figure 13, the two improved algorithms have very similar performance. The average precision of improved algorithm 2 is 0.22% higher than that of improved algorithm 1. They also have very close performance in precision for each class, and the difference is that the improved algorithm 1 does not use label smoothing, while the improved algorithm 2 uses it. According to the

experimental results in this section, it can be seen that GIOU can also be used as the position regression loss function to well optimize the model.

Table 6. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with GIOU loss function in terms of precision.

Type	Faster-RCNN (%)	Original YOLO (%)	Improved YOLO with IoU (%)	Improved YOLO with GIOU (%)
Suv	75.10	71.80	89.63	90.30
Sedan	74.78	88.55	89.97	90.34
Taxi	85.83	89.55	89.91	89.99
Van	85.50	82.15	93.03	92.53
Truck-Box	89.95	76.94	95.82	95.82
Bus	90.01	82.94	96.04	96.36
MiniVan	77.27	74.48	89.56	89.83
Truck-Util	95.53	40.82	95.36	95.4
Pickup	90.89	35.54	92.99	93.74
mAP	84.99	71.42	92.48	92.70

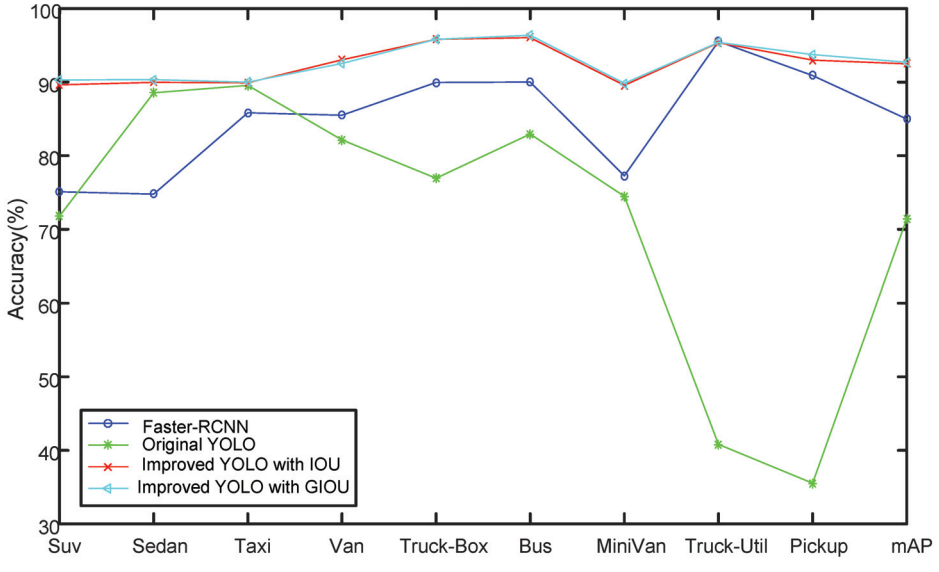


Figure 13. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with GIoU loss function in terms of precision.

As shown in Figure 14, similar to the comparison in terms of precision, the omission ratio of improved algorithms in this section is significantly lower than that of the Faster-RCNN algorithm and the original YOLO algorithm.

To sum it up, we believe that GIOU and IoU can be used as the position loss function to well regress the target position, accelerate the model's convergence speed, reduce the training volume of the model and improve the final performance of the model.

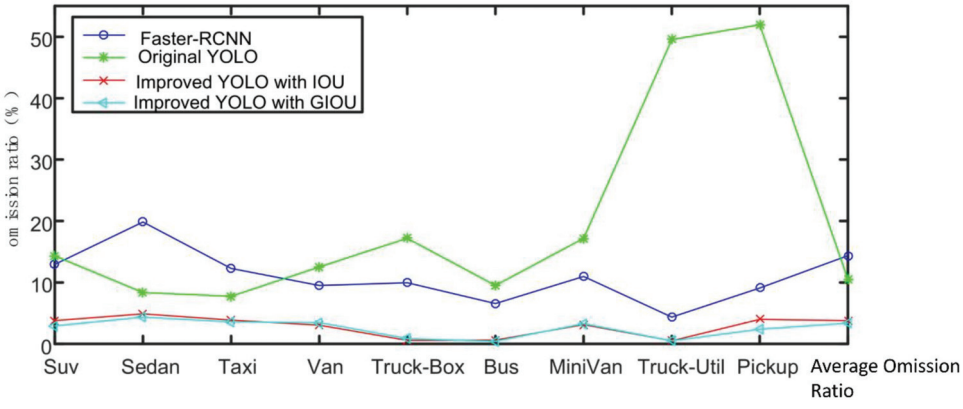


Figure 14. Comparison of Faster-RCNN, original YOLO and improved YOLO algorithm with GIOU loss function in terms of omission ratio.

5. Conclusions

In this article, we propose a distributed system based on smart streetlights, build the dataset UA-DETRAC-LITE-NEW, and improve the YOLOv3 algorithm in various respects. On the intelligent streetlight platform, we have realized fine-grained automobile detection for SUVs, sedans, taxis, commercial vehicles, small commercial vehicles, vans, buses, trucks and pick-ups; we have conducted scale optimization for the vehicle target and improved the model by introducing multi-scale training and anchor clustering; as a result, the detection accuracies for trucks and pick-ups are improved by 26.98% and 16.54%, respectively, and average accuracy is increased by 8%. The label smoothing and mixup methods are used to improve the model's generalization ability, and its generalization ability is increased by 16.01% compared to the original YOLO algorithm. By combining the position regression loss function of IOU or GIOU for optimization, the system accuracy can reach 92.7%, which is 21.28% higher than that of the original YOLOv3 algorithm. For further development, we can continue to optimize the YOLOv3 algorithm from the perspective of confidence loss.

Author Contributions: F.Y. proposed the idea of this article and finished the manuscript. Z.H. led the research project. D.Y. finished the code and performed the experiment. Y.F. performed experimental results analysis. K.J. participated in the experiment. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Sichuan Province Science and Technology Program Research Project (No.2019YJ0174).

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

References

1. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 383–398. [CrossRef]
2. Maddio, S. A Compact Circularly Polarized Antenna for 5.8-GHz Intelligent Transportation System. *Antennas Wirel. Propag. Lett.* **2017**, *16*, 533–536. [CrossRef]
3. Herrera-Quintero, L.F.; Vega-Alfonso, J.C.; Banse, K.B.A.; Carrillo Zambrano, E. Smart ITS Sensor for the Transportation Planning Based on IoT Approaches Using Serverless and Microservices Architecture. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 17–27. [CrossRef]
4. Ferreira, D.L.; Nunes, B.A.A.; Obraczka, K. Scale-Free Properties of Human Mobility and Applications to Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3736–3748. [CrossRef]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325.
7. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Chu, W.; Liu, Y.; Shen, C.; Cai, D.; Hua, X.-S. Multi-Task Vehicle Detection With Region-of-Interest Voting. *IEEE Trans. Image Process.* **2018**, *27*, 432–441. [CrossRef] [PubMed]
10. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
11. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
12. Cao, C.-Y.; Zheng, J.-C.; Huang, Y.-Q.; Liu, J.; Yang, C.-F. Investigation of a Promoted You Only Look Once Algorithm and Its Application in Traffic Flow Monitoring. *Appl. Sci.* **2019**, *9*, 3619. [CrossRef]
13. Lyu, S.; Chang, M.-C.; Du, D.; Li, W.; Wei, Y.; Del Coco, M.; Carcagnì, P.; Schumann, A.; Munjal, B.; Choi, D.-H.; et al. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6.
14. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.-C.; Qi, H.; Lim, J.; Yang, M.-H.; Lyu, S. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]
15. Lyu, S.; Chang, M.-C.; Du, D.; Wen, L.; Qi, H.; Li, Y.; Wei, Y.; Ke, L.; Hu, T.; Del Coco, M.; et al. UA-DETRAC 2017: Report of AVSS2017 & IWT4S Challenge on Advanced Traffic Monitoring. In Proceedings of the Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–7.
16. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
17. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
19. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
20. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv* **2018**, arXiv:1710.09412.
21. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Appl. Sci.* **2019**, *9*, 3775. [CrossRef]

algorithms

MDPI

Article

An IoT System for Social Distancing and Emergency Management in Smart Cities Using Multi-Sensor Data

Rosario Fedele [1] and Massimo Merenda [1,2,*]

[1] Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), University Mediterranea of Reggio Calabria, 89124 Reggio Calabria, Italy; rosario.fedele@unirc.it
[2] HWA srl-Spin Off dell'Università Mediterranea di Reggio Calabria, Via Reggio Campi II tr. 135, 89126 Reggio Calabria, Italy
* Correspondence: massimo.merenda@unirc.it; Tel.: +39-0965-1693-441

Received: 11 August 2020; Accepted: 3 October 2020; Published: 7 October 2020

Abstract: Smart cities need technologies that can be really applied to raise the quality of life and environment. Among all the possible solutions, Internet of Things (IoT)-based Wireless Sensor Networks (WSNs) have the potentialities to satisfy multiple needs, such as offering real-time plans for emergency management (due to accidental events or inadequate asset maintenance) and managing crowds and their spatiotemporal distribution in highly populated areas (e.g., cities or parks) to face biological risks (e.g., from a virus) by using strategies such as social distancing and movement restrictions. Consequently, the objective of this study is to present an IoT system, based on an IoT-WSN and on algorithms (Neural Network, NN, and Shortest Path Finding) that are able to recognize alarms, available exits, assembly points, safest and shortest paths, and overcrowding from real-time data gathered by sensors and cameras exploiting computer vision. Subsequently, this information is sent to mobile devices using a web platform and the Near Field Communication (NFC) technology. The results refer to two different case studies (i.e., emergency and monitoring) and show that the system is able to provide customized strategies and to face different situations, and that this is also applies in the case of a connectivity shutdown.

Keywords: smart city; emergency management; monitoring; social distancing; neural network; internet of things; wireless sensors network; near field communication technology; computer vision

1. Introduction

The application of Information and Communication Technologies (ICTs) in real contexts is a key factor for the development of smart cities, where proper levels of quality of life and environment, safety, and sustainability are the main targets [1].

The benefits related to the use of ICTs affect everyday life in both normal and emergency conditions. The occurrence of accidental or catastrophic natural events (e.g., sudden floods, earthquakes, fires, etc.) and the adoption of inadequate management strategies are the main causes of the occurrence of emergency conditions in structures and infrastructures. These causes can affect the structural health status of the abovementioned assets, which in turn can affect the health and safety of highly populated areas (i.e., urban contexts, buildings, occasional assembly points for crowds, theme parks, etc.). Significant improvements in disaster management are expected if artificial and human intelligence are integrated (e.g., Disaster City Digital Twin paradigm; cf. [2]), affecting the efficiency of real-time monitoring (by social and remote sensing), of the data analysis (to detect and monitor human activities, damages, and relief needs), and of the scenario simulations (for various and fair allocation of resources, or for training and planning purposes). Consequently, the main disaster management stages (e.g.,

preparedness and risk reduction, disaster response, recovery) can be boosted if proper information sharing, which includes different systems and stakeholders, is carried out [3].

Furthermore, emergency conditions are defined as the presence or the spreading of high concentrations of pollutants (environmental risk), or of diseases and epidemics caused by bacteria or viruses (biological risk). In the first case, thresholds can be determined, and pollutants can be monitored (e.g., using ICT solutions based on fixed and mobile sensor nodes that actively include citizens in the information gathering [4]) to verify that the pollutants' concentrations are under the given thresholds, or to trigger predefined alarms. In the second case, it is crucial to manage the spatiotemporal distribution of the crowds by using technologies that allow for the implementation of specific and well-designed measures, such as physical distancing (or social distancing) and movement restrictions [5], which are well-known strategies to reduce infection and mortality risks [6,7].

1.1. Literature Review on Available Solutions

In order to face the above problems, several approaches have been proposed. Among them, noteworthy examples of Internet of Things Wireless Sensor Networks (IoT -WSNs), monitoring platforms and recommender systems are reported in the following (Table 1), also exploiting the use of machine learning (ML) in IoT systems [8]. Importantly, the applicability of the WSNs mainly depends on the lifetime of the sensor nodes, and, for this reason, it is important to design this type of system while bearing in mind this crucial aspect and selecting the more convenient energy efficient routing protocol [9–11]. Other important design parameters are [10,12,13]: (i) the limited storing and computational resources of each sensing nodes, (ii) the costs (i.e., cheap sensors are prone to failure, while expensive sensors need good housing and cannot be used for dense deployments), (iii) the position of each sensing node, which cannot be predetermined and depends on the accessibility of the point where the node should be placed, (iv) the sensing nodes' deployment (to collect the needed data, to have the required coverage and connectivity, to extend the network lifetime, and to minimize energy consumption), and (v) the minimum number of time slots required to aggregate data along the edges of a data-gathering tree spanning all the nodes in a WSN (a.k.a., minimum aggregation delay), if the gathered data are aggregated before the transmission to the control center. The solution presented in this paper was designed while bearing in mind all the design parameters mentioned above, focusing, in particular, on maximizing the exploitation of the nodes' storing and computational resources, on minimizing the system cost and on optimizing the system deployment.

Table 1. Examples of Wireless Sensor Networks (WSN) and Internet of Things (IoT)-WSN solutions for emergency and disaster management.

Reference	Main Characteristics	Limitations
[14]	WSN based on energy-efficient wireless sensor nodes equipped with an ultrasonic sensor, which were tested in a field experiment (explosion in a building) to confirm functionality and reliability in terms of collision-free data transmission during the emergency.	Buildings only; Ultrasonic sensor only; Explosion emergency only; Does not consider the overcrowding.
[15]	WSN based on the idea of monitoring the earthquake precursors (e.g., unusual movement of animals, ground water pressure, radon emission, etc.), which was designed for early earthquake warnings and disaster management.	Earthquake emergency only; Does not consider the overcrowding.
[16]	WSNs used together with Unmanned Aerial Vehicles (UAV) for monitoring, forecast, early warning, information fusion and sharing, logistics, evacuation, search and rescue mission, damage assessment.	Does not consider the overcrowding.

Table 1. *Cont.*

Reference	Main Characteristics	Limitations
[17]	WSN paradigm for real-time applications in smart cities, which aims at balancing performance and energy consumption, and uses the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) optimization technique to find the shortest data transmission path.	Paradigm; Does not consider the overcrowding.
[18]	WSN based on a method inspired by biological intracellular signaling, which was designed to perform smog pollution sensing, and taking into account the ad hoc demand routing protocol (AODV) and bellman-ford and interzone routing protocol (IERP) for data transmission.	Air pollution only; Does not consider the overcrowding.
[19]	IoT-WSN based on an evidence-based interactive trust management system for disaster management.	Medical emergency only; Communications between autonomous and adaptive nodes.
[20]	WSN that aims at detecting, in a disaster scenario, moving people without 'tracking devices' (i.e., carrying out the so-called Device free Passive Localization, DfPL).	Does not consider the emergency detection.
[21]	IoT-WSN based on smart fire sensors, cameras, and a Convolutional Neural Network (CNN), acting as a surveillance monitoring system for detecting disasters that occur in a remote area (e.g., a forest).	Remote area only; Does not use sensors for structural monitoring.
[22]	IoT-WSN based on machine learning algorithms that run in a cloud server and includes a modular redundancy fault tolerant scheme to obtain an accurate prediction from sensor data (gas and force sensors) managed by the ultra-low power MSP430 board and a Raspberry Pi, which was designed for early warning in an industry environment.	Gas and force sensors only; Industry environment only; Does not consider overcrowding.
[23]	IoT-WSN that uses the Advanced Adaptive Wavelet Sampling Algorithm (AAWSA) for prolonging the lifetime and power consumption of sensor nodes that include several sensors (i.e., moisture sensors, pressure sensor, rain gauge, tilt meters, and strain gauge), which was developed for disaster prediction in an urban region.	Does not consider overcrowding.
[24]	IoT-based architecture that collects real-time data from the city (from existing sensors at home, parking, vehicular networking, surveillance, weather and water monitoring system, etc.), implemented in the Hadoop ecosystem that allows the processing of Big Data, to obtain a "Smart Digital City".	Architecture; Based on existing sensors.

Another group of solutions for the problems mentioned above is represented by software platforms (Table 2).

Table 2. Examples of platforms for emergency and disaster management.

Reference	Main Characteristics	Limitations
[25]	2D and 3D WebGIS-based platform that has a scalable network architecture and uses a three-tier software architecture, which was designed for effective landslide multilevel management, and an emergency response.	Landslide emergency only; Does not consider overcrowding.
[26]	SENS-ME platform that aims at exploiting the functionalities of Commercial Off-The-Shelf (COTS) smartphones to carry out opportunistic networking, mobile sensing, and distributed information processing.	Does not consider emergency detection. Smartphones as sensors.
[27]	Flood disaster management system (FDMS) that carries out environmental model selection and disaster-related data binding.	Flood disaster only.

Table 2. *Cont.*

Reference	Main Characteristics	Limitations
[28]	DECATASTROPHIZE (a.k.a., DECAT) platform that aims at managing disasters or multiple and/or simultaneous natural and man-made hazards by means of a Geospatial Early-warning Decision Support System (GE-DSS) that allows early warning, decision making, rapid mapping, impacts assessment and mitigation, and geospatial data/information dissemination.	Does not consider overcrowding.
[29]	A web platform developed by the Emergency and Security Coordinating Centre to improve the decision making process of the Canary Islands' Authorities, which provides a geographical and temporal incident distribution and which is able to forecast and classify incidents.	Emergency and security incident distribution only.
[30]	A Building Information Modeling (BIM)-based platform that was designed for building fire emergency management in a dynamic way, i.e., using building users' behavior decisions (e.g., escape, wait for rescue, and fire extinguishing) and both fire and users' positions, which plans action routes and provides visual route guidance.	Buildings only; Fire emergency only; Does not consider overcrowding.
[31]	A smartphone-based platform for city-wide crowd management (through a "heat map-like" system for a real-time overview of the spatiotemporal distribution of crowds in given areas and through specific messaging for real-time, smart, adaptive emergency response and evacuation strategies), which aims at having smart crowds in smart cities and which was used in at least three European countries (UK, Netherlands, Switzerland).	Smartphones as sensors; Does not consider emergency detection.
[32]	A cloud-based architecture for emergency management and first-responders localization (landmark-based and landmark-free), which aims at supporting coordinated emergency management in smart cities based on the localization of first responders during crisis events.	Localization of first responders only.
[33]	Smart disaster management system for transportation applications in smart cities, which gathers information from multiple sources and locations (using VAENTS, i.e., Vehicular Ad hoc Networks, such as Vehicle-to-Vehicle, V2V, Vehicle-to-Infrastructure, V2I, or smartphones or other technologies), detects the point of incidence, makes strategies and decisions (using, e.g., high-performance computing, HPC), and propagates the information to vehicles and other nodes in real time.	Traffic incident only.

Other important and innovative examples of ICT solutions for emergency management refer to: (a) smartphone-based information systems [34]; (b) mobile post-disaster management systems based on free and open source technologies [35]; (c) satellite remote sensing for disaster management [36]; (d) exploiting data from social media [37]; (e) using deep learning to identify survivors in debris from images gathered by smart infrastructures [38]; (f) using an IoT ecosystem (wellbeing-wearable and home automation system sensors with varying communication protocols) to empower the elderly population to self-manage their own health and stay active, healthy, and independent as long as possible within a smart and secured living environment [39].

Recommender systems technology aims at reducing the consumer over-choice due to the huge amount of information available on the web [40]. These systems use information (e.g., location, preferences, past interaction, item features, etc.), gathered through few interactions with its user, to create a customized selection of items (e.g., recommendation list) that can interest the user and can facilitate a better user experience [40,41]. Recommender systems could be further improved if Machine Learning (ML; users extract the features from raw data to feed the algorithm) and Deep Learning (DL; the algorithm automatically extracts features from raw data) algorithms were used for information retrieval [40]. Noteworthy applications of recommender systems in smart cities include: 1) a mobile IoT recommender system for users that need to find Park-and-Ride infrastructures to switch from a private to a public transportation mode [42]; 2) an autonomous situation-aware evacuation route recommender

architecture, optimized in real time, to obtain smart buildings [43]; 3) a recommendation system based on the concept of Social IoT (SIoT), i.e., use and reuse data generated by various IoT applications, and adapt the services provided by the single IoT system, to improve the user experience [44].

Based on the literature, different strategies can be adopted to derive/recognize safe routes and assembly points. Traditionally, the problems above can be solved using Shortest Path Finding (SPF) algorithms that belong to the mathematical field of Combinatorial Optimization. Noteworthy examples are: (1) Prim's algorithm, which can be used to derive the Minimum Spanning Tree (MST; i.e., a tree diagram without circles) [45]; and (2) the Dijkstra's and the A* algorithms, which allow one to derive the shortest path between two points [46]. In the Matlab environment, it is possible to find different tools (e.g., "shortestpath" and "digraph") that allow one to solve this problem [47].

This study also refers to a bidirectional exchange of information between a platform and its users, which can be satisfied using NFC technology [48].

The efficiency of the solutions presented above is strictly related to the real-time knowledge of the spatiotemporal distribution of the population, which is fundamental for risk analysis and emergency management (i.e., for an estimation of human exposure and vulnerability) [49]. One of the most used techniques to model or track moving crowds is Computer Vision [50]. Convolutional Neural Networks (CNNs) represent the key algorithms in computer vision, and recently [51], enhanced versions of the tree growth and firefly metaheuristics algorithms have been proposed to automatize the hyperparameters' optimization process (i.e., find the right set of hyperparameters that allow one to obtain the best CNN model accuracy), obtaining better performances (in terms of image classification accuracy and the use of computational resources) than the traditional approaches (that require time expertise). Furthermore, faster regions CNN (Faster R-CNN) is considered one of the most important techniques for automatic pedestrian detection from video [52], and, if automatic color enhancement (ACE) is used, good performances (in terms of recognition rate and offset of target selection) can be obtained because of the reduced susceptibility to the diversity of pedestrians' appearances and the light intensity in specific scenarios (e.g., a subway) [52]. Other suitable techniques, such as satellite imagery processing, could be used as an aid for the automatic recognition of vegetation, landslides, and geospatial data [53,54]. Note that an easy detection of a full-body person in perspective can be carried out using other methods, such as Mask R-CNN [55], Pose2Seg [56], and EfficientDet [57]. Finally, Convolutional Neural Networks are methods specialized in a grid-like structure or multiple arrays form [58,59].

1.2. Objectives and Scopes

In light of the above, previous works [60,61] presented an innovative IoT-WSN platform that was designed as a decision support tool for an Italian theme park. The main limit of the proposed platform was to provide solutions (emergency plans) that neglected the people density (number and spatiotemporal distribution). However, the literature review reported above allows one to identify solutions, i.e., the recommender system and the computer vision applications, which can be effectively used to improve the abovementioned tool. Hence, for this reason, in the study described in the following sections, we report the results of the integration of these solutions into the previous platform.

The main novelty of the study refers to the integration in an all-in-one solution, for the first time as per the authors' knowledge, of available and new algorithms and ICT solutions that allow one to solve two important problems related to the theme park, i.e., the overcrowding and the emergency management, which were presented as two case studies in this paper.

Consequently, the remaining part of the paper is organized as follows. The following section (Section 2) describes the IoT system, based on the platform cited above (Hardware in Section 2.1), and how the system uses input data (Software in Section 2.2) to carry out the main objective of this study, i.e., obtain a multipurposes IoT-WSN-based system that can be used by authorities for emergency and crowd management, and by authorized customers as a decision support tool. Subsequently, Section 3 describes two case studies that were used to validate the proposed system, including dataset

generation and algorithms application. Section 4 reports the main results related to the case studies mentioned above, which were derived from simulations. Finally, the last two sections contain the main conclusions derived from this study and some anticipatory remarks about future works.

2. IoT System Description

In this section, the main components of the IoT system presented in this paper will be described. The system consists of two components, i.e., a hardware (i.e., an IoT-WSN) and a software (NN-based and SPF algorithms for emergency management and social distancing purposes).

2.1. Hardware: A Multisensor IoT-WSN

The IoT-WSN proposed in this paper consists of wireless sensing nodes, which are fed using a suitable tuned photovoltaic energy harvester [62]. Figure 1 shows the main components of the first prototype of one wireless sensing node of the WSN. In particular, the figure cited above shows (1) the power supply unit of the sensing node (which consists of a 5 W-18 V polycrystalline silicon photovoltaic panel, a 12 V-20 A recharge circuit, and a 12 V-12 Ah battery), and (2) an ultralow-power IoT board [63], which is the core of the sensing node.

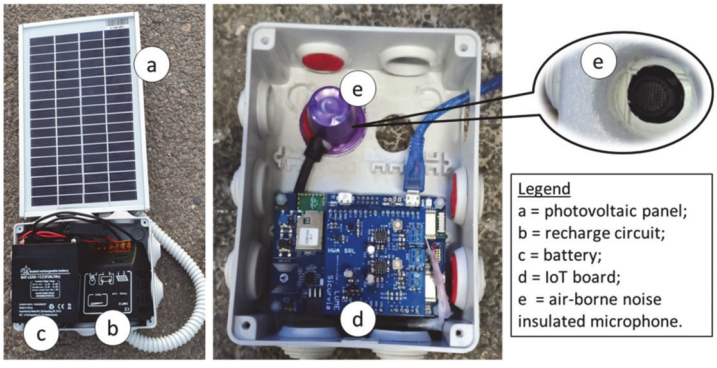


Figure 1. First prototype of the wireless sensing node of the proposed WSN.

In more detail, the IoT board includes several MEMS sensors (i.e., a 3D accelerometer, a microphone, a temperature/humidity sensor, magnetometer, barometer, etc.) and a microcontroller. The IoT board can wirelessly transmit the data gathered by the sensors, using different standards and protocols, i.e., Wi-Fi, Bluetooth, and NFC tags. It is important to underline that the NFC protocol allows for further applications, e.g., the interaction/exchange of information with the theme park visitors during normal and emergency conditions, also in the absence of a remote connection, thus allowing a backup communication channel with the users. Note that, to obtain a more detailed environmental and structural monitoring, the IoT board was also equipped with a smoke sensor (to detect carbon monoxide, liquid petroleum gas, and smoke), a flame sensor, and an additional microphone. The latter was added to receive the vibro-acoustic response [64–66] of the structure on which the units were installed. In more detail, the additional microphone was isolated from the airborne noise through a cover (inside the box) and isolating material (between the box and the structure), and was able to receive the acoustic signals (a.k.a., vibro-acoustic signature of the structure) that travelled into the structure. Proper analyses of these acoustic signals (i.e., feature extraction, multidomain analysis for cracks identification and monitoring, structural health status classification; [64–66]) allow one to carry out the Structural Health Monitoring (SHM) of the structures where the sensing node is attached, which is also applicable in the case of road monitoring, where the use of self-powered and smart sensors is envisioned [67].

The study reported in this paper refers to two case studies that were carried out using, as a reference place, an Italian theme park. Figure 2a shows the map of the park, where lines point out boundaries and main routes (solid and dashed, respectively), numbers (from 1 to 6) point out the positions in which the sensing nodes of the WSN were installed, and the positions of the two exits (called Exit 1 and Exit 2) and the two assembly points (called AP 1 and AP 2) are indicated. Figure 2 also shows three prototypes of wireless sensing nodes installed at different points of the theme park, i.e., a road pavement near the Exit 1 (see Figure 2b, which refers to point 1 in Figure 2a), a masonry structure (historical military fort; see Figure 2c, which refers to point 5 in Figure 2a), a light pole (see Figure 2d, which refers to point 6 in Figure 2a). Importantly, the prototype cited above have been subsequently equipped with Wi-Fi cameras that allow computer vision analyses aimed at defining the spatiotemporal distribution of the visitors of the theme park per given area.

Figure 2. (**a**) Map of the theme park where the WSN was installed and three self-powered wireless sensing nodes installed on (**b**) a road pavement, (**c**) a masonry structure, and (**d**) a light pole in the park.

2.2. Software: NN-Based and SPF Algorithms for Emergency Management and Social Distancing

The framework of the proposed IoT system is depicted in Figure 3. In particular, the system includes the WSN (which was described in the previous section and is represented by box 1 in Figure 3) and a procedure, consisting in several steps (represented by boxes 2 to 13 in Figure 3), which allows one to analyze the data that come from the WSN (i.e., sensing nodes and cameras).

The sensing nodes of the WSN (see box 1 in Figure 3) gather data from the environment using their sensors, while the Wi-Fi cameras detect the people's position in a given area. Sensing nodes and cameras are fed by a photovoltaic-based power supply system. The aforementioned data are sent, with a proper timing (e.g., every five minutes), to a local server (see box 2 in Figure 3) for analysis/backup purposes. At the same time, the IoT board converts the collected data to text strings (JSON format) and sends the converted data to a web server (see box 3 in Figure 3) using the MQTT protocol (e.g., the MQTT broker Eclipse Mosquitto™) for backup purposes. Note that, sampling and data transmission frequencies are defined so as to minimize the transmission cost, latency, network bandwidth, and resource requirements, and to increase data privacy and data transfer reliability.

The data stored in the web local server can be accessed by an authorized customer of the system (i.e., subscribing to a specific topic) for further analyses, while the data stored in the local server are analyzed as follows using three different algorithms (see boxes 4–6 in Figure 3).

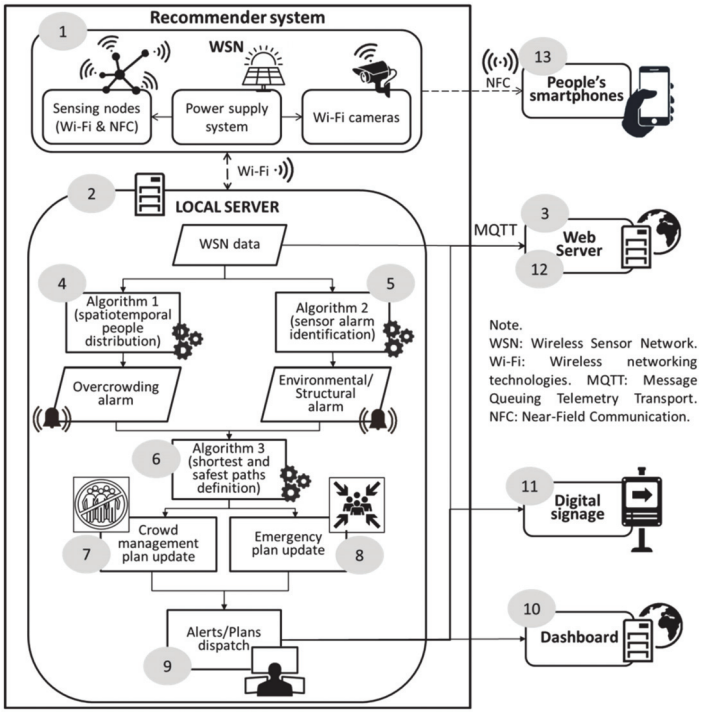


Figure 3. Framework of the proposed IoT system.

The first algorithm (Algorithm 1, represented in box 4 in Figure 3) is used to process the data that come from the cameras using computer vision to define the spatiotemporal distribution of the people in each area and trigger the overcrowding alarm (if people in a given area exceed a predefined threshold). Hence, Algorithm 1 can be used as an automatic tool when the social distancing measure must be respected or to limit the access to a given area, avoiding overcrowding. In more detail (cf. Table 3), this algorithm uses object detection frameworks to classify and locate objects in a visual field. Facial detection would take an image input of some kind, check for the person or face class of the objects, and locate them in the frame. Additionally, facial recognition would pick out eyes, mouths, and various other features to compare them to a known dataset. The applications detect humans in the visual field via processing blocks pre-trained by crunching a huge number of images with a deep learning artificial intelligence system. The extracted features, as the number of humans and the direction of the people flow, are used in the following to determine the spatiotemporal distribution.

Table 3. Main characteristics of the Algorithm #1.

Scopes	Input	Output	Steps
Count the number of people entering and leaving a given area using data from cameras.	• Video stream from cameras installed at strategic points of the theme park.	• Number of people entering and leaving a given area.	- Receive data from cameras; - Count the number of people entering and leaving a given area using data from cameras.

The second algorithm (Algorithm 2; see box 5 in Figure 3) is used to process the raw data from the sensors (cf. Table 4). In particular, raw data are preprocessed in order to extract meaningful features (i.e., statistical indicators such as maximum, minimum, average, root mean square, etc.), which are used as input in a NN for clustering purposes. This NN is used to identify the occurrence of one or

more environmental and/or structural alarms detected by one or more sensing nodes of the WSN. The output of the NN cited above is one of all the possible combinations of alarm. The number of alarm combinations is equal to the permutations of the number of sensing nodes without repetitions and can be calculated using the following expression:

$$A = \sum_{i} \frac{N!}{K_i! \cdot (N - K_i)!}, \quad (1)$$

with i = 1, . . . , 6, and where N is the length of the array {1, 2, 3, 4, 5, 6} that represent the sensing nodes of the WSN, and K is an integer that shows how many nodes detected an alarm at the same time. Note that the WSN used in this study consists of six sensor nodes. Hence, the number of alarm combinations A without repetitions (e.g., the condition "alarm actives at node 1 + alarm actives at node 3" is equal to the condition "alarm actives at node 3 + alarm actives at node 1") is 63.

Table 4. Main characteristics of the Algorithm #2.

Scopes	Input	Output	Steps
Classify environmental and structural conditions and trigger related alarms when predefined thresholds are exceeded.	• Sensor data; • Thresholds.	• Features from sensor data; • Emergency (environmental, and/or structural) alarm(s).	- Receive sensor data; - Extract features from sensor data; - Classify sensor data identifying environmental and structural conditions based on predefined thresholds.

The third algorithm (Algorithm 3, represented by box 6 in Figure 3) is used to define the shortest and the safest paths if the Environmental and/or Structural alarm or the Overcrowding alarm is active in at least one node of the WSN. In particular (cf. Table 5), the Algorithm 3 automatically updates: (1) when the overcrowding alarm is active, a crowd management plan that consists of an image that highlights, with red areas and white signals, the overcrowded areas (see two examples in Figure 6); (2) when the environmental/structural alarm is active, an Emergency plan that consists of an image (see two examples in Figure 8) that shows in which nodes the alarm is active (with red circles and yellow triangles), if an exit is closed (with a no entry sign), and the exit or the assembly point that must be reached (with green and white signs, respectively). In more detail, Algorithm 3 is based on the graphs shown by Figure 4 (i.e., graph in Figure 4a was used in the case study 1, while graph in Figure 4b was used in the case study 2) and was developed using Matlab tools (i.e., using the "shortestpath" and "digraph" functions [47]). It takes as input the connections among the graphs and the alarm conditions, and takes the people in each area as weight; it then returns the sequence of nodes that should be passed through, which corresponds to the safest path for the people who, because of active alarm/s, must reach a predefined point (which could be an exit or an assembly point) from their starting node.

In summary, the fastest (i.e., on paths traveled by few people) and safest (i.e., avoiding nodes where an alarm is active) path is suggested at the same time, considering that the spatiotemporal distribution of the people could affect the speed of the flow to reach the available exits or, alternatively, the assembly points.

Note that Algorithm #3 is based on the assumption that all the edges of the graph have the same endless capacity, and the main parameter used to evaluate the availability of each edge is the number of people that are on the edge when an alarm is activated. Based on this assumption, the objective of the algorithm is not the flow of people that can move on the graph per unit of time but to recognize and provide, as a solution, the less busy path to reach an exit or an assembly point. Importantly, although the Matlab function "shortestpath" was used, Algorithm #3 does not select the "shortest" path but selects the path that includes edges that allow one to reach a final point (exit or assembly point) while avoiding slowdowns, even though the path is the longest one.

Table 5. Main characteristics of the Algorithm #3 (consisting of two parts).

Scopes	Input	Output	Steps
• Close area(s) in case of overcrowding alarm; • Find the shortest and the safest paths in case of an emergency alarm.	For part 1: • Graph of the park; • Number of people in each area of the park; • Number of people entering and leaving each area of the park.	For part 1: • Graph of the park with closed and open areas; • Number of people leaving overcrowded area(s); • Number of people entering and leaving not overcrowded area(s).	For part 1: - Compare number of people (P) inside an area with a threshold (T); - If P>T, trigger the overcrowding alarm; - Close overcrowded area(s); - Define the flows of people leaving the overcrowded area(s). - Update the digital signage and the dashboard.
	For part 2: • Graph of the park; • Active alarm(s) at the sensing nodes.	For part 2: • Graph of the park with nodes with activated alarms; • Shortest and safest paths to reach an available exit(s) or assembly point(s).	For part 2: - Update graph's node(s) N with activated alarm(s); - If N is(are) closed to exit(s), exit(s) is(are) closed. - If all exits are closed, the assembly points are activated; - Shortest and the safest paths are calculated to leave all the nodes and reach an available exit(s) or assembly point(s). - Update the dashboard.

Figure 4. Graphs used by the Algorithm 3 in case of (**a**) Overcrowding alarm (Case study 1: monitoring scenario and social distancing); (**b**) Environmental and/or Structural alarm (Case study 2: emergency scenario and emergency management).

Finally, the local server dispatches the plans defined above using different ways, i.e., (1) using internet via a dashboard of the IoT system (e.g., an open-source server-side platform dashboard for controlling IoT devices and visualizing the sensor data in real time was used in this study (Thingsboard [68]); see box 10 in Figure 3; Figure 5a) that can be accessed by the users of the IoT system (e.g., staff members or visitors of the park); (2) using a digital signage that is carefully spread throughout the park (see box 11 in Figure 3); (3) using internet via the abovementioned web server (see box 12 in Figure 3); (4) via the NFC interface of the sensing nodes (see box 13 in Figures 3 and 5b) in case of connectivity shutdown or if the user is close to the node and has a smartphone.

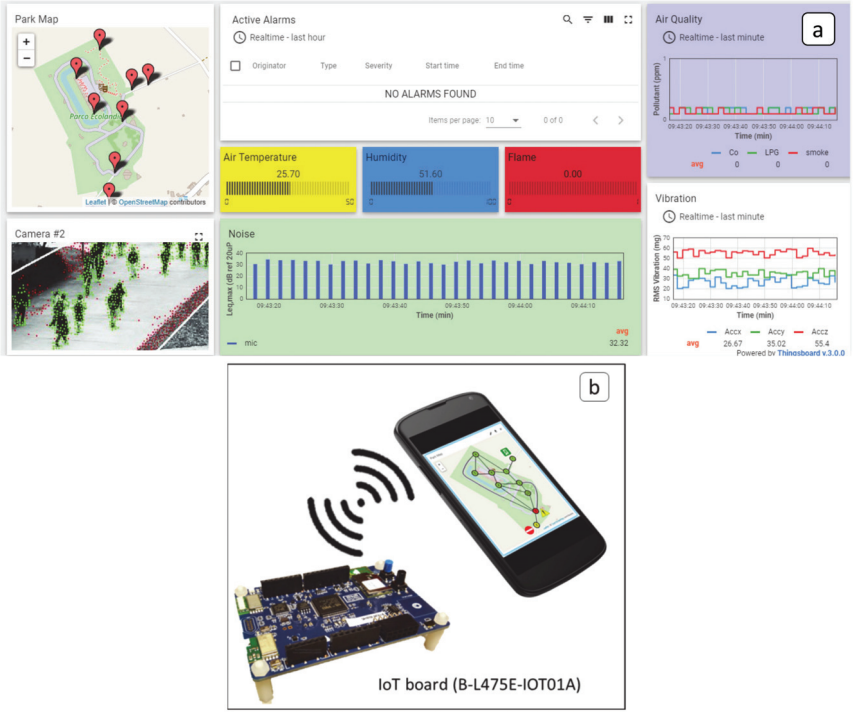


Figure 5. (**a**) IoT system dashboard; (**b**) Emergency plan provided by the NFC module-smartphone connection in the form of a smartphone compatible map.

3. Case Studies

This section of the paper contains the description of two simulated cases studies in which the IoT system was applied to face two different problems, i.e., an emergency scenario and a monitoring scenario, occurring in the abovementioned theme park.

3.1. Case Study 1: Monitoring Scenario (Social Distancing)

The first case study refers to a monitoring scenario. In particular, the IoT system is envisioned as a tool to control the spatiotemporal distribution of the people in a given area and to trigger, automatically, the overcrowding alarm that allows one to respect social distancing. Figure 6 shows two examples of a crowd management plan, where the overcrowded areas are highlighted in red, while white signals indicate which sensing node detected the overcrowding alarm.

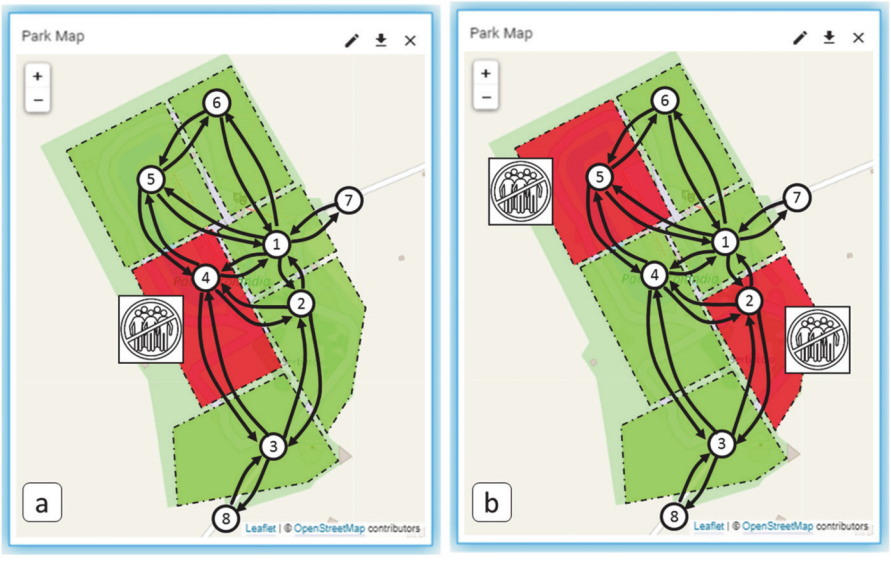


Figure 6. Two examples of a crowd management plan in case of an overcrowding alarm active at (**a**) node 4, and (**b**) nodes 2 and 5.

3.1.1. Case Study 1: Data Generation

As described above (cf. description of Figure 3), the data for this application consists of video from Wi-Fi cameras sent using a Raspberry Pi. The used cameras provide a video with 30 fps and a resolution of 1280x960 pixels. Figure 7 reports an example of the output obtained from the Algorithm 1 (note that, in this case, the algorithm counts the people between the two white lines on the pavement).

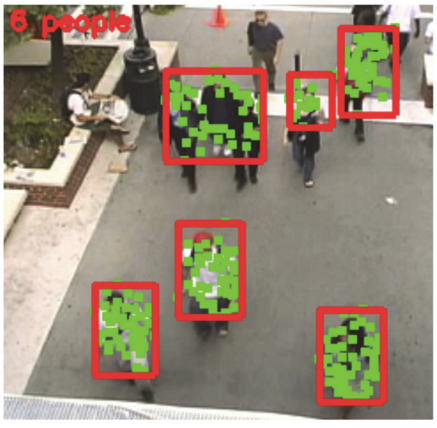


Figure 7. Example of the application result of the Algorithm 1 (computer vision).

3.1.2. Case Study 1: Algorithms

Based on the above, a computer vision algorithm—the Algorithm 1—with the following characteristics can be used to derive the spatiotemporal distribution of the people in each monitored area of the park (see Figure 4a) from the dataset described above (i.e., video from Wi-Fi cameras) and is used to trigger the overcrowding alarm. The software was developed using the programming language Python.

In pursuing the implementation of person detection and tracking, we use a two-step procedure for (a) the detection of the position of the people in the video frame, carried out periodically due to the large computational cost and implemented with the built-in OpenCV algorithm HOG (Histogram of Oriented Gradients) + Linear SVM model [69], and (b) the tracking of the people's movement around the video frames, using the built-in OpenCV algorithm Discriminative Correlation Filter with Channel and Spatial Reliability (DCF-CSR) [70].

For every camera installed in every single node, the output of the software task is the value of people actually in the area, with a refresh rate of 1 sec. The maximum number of people admitted in every single area is a priori known, so an alarm is raised if the number of people in the area is above the threshold. If the overcrowding alarm is detected by Algorithm 1, the Algorithm 3 is triggered to define a crowd management plan, as shown in the previous section. In this case, the "digraph" function [46] is automatically activated to derive the shortest path that the people in the overcrowded area must follow to leave the area.

3.2. Case Study 2: Emergency Scenario (Emergency Management)

The second case study refers to an emergency scenario, in which sensor alarms are active and the IoT system is envisioned as a tool to control the environmental and structural conditions of the areas and assets of the park and to automatically recognize several sensor alarm conditions and provide, in real-time, customized emergency plans that show safe paths for the park's visitors, who need to reach an exit or an assembly point from one of the six areas of the park because of the emergency.

Figure 8 shows two examples of emergency plans, which contain: (i) a graph that indicates the connections among the nodes, exits, and assembly points; (ii) sensing nodes that are represented by red and green circles if an alarm is active or not, respectively; (iii) exits that are represented by yellow and green circles if the exit is safe or not, respectively; (iv) yellow triangles with a black exclamation point that indicate the point where the alarm is active; (v) green signals that indicate an available exit and red access-denied signals that point out an unavailable exit (closed); and (vi) white signals that show the assembly points.

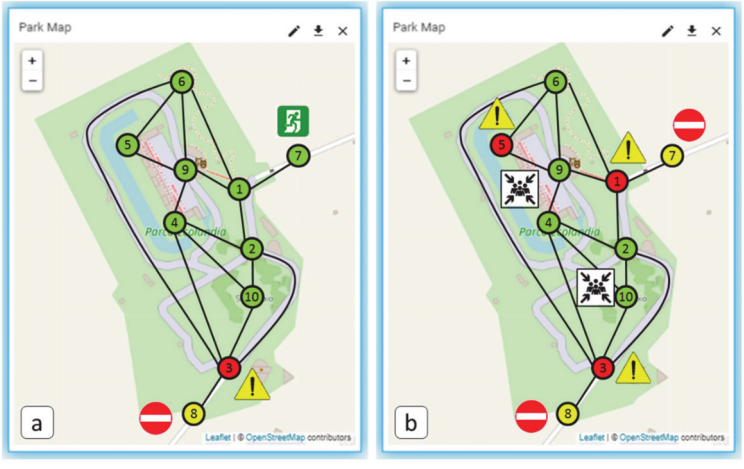


Figure 8. Two examples of emergency plans in case of an environmental/structural alarm active at (**a**) node 3, and (**b**) nodes 1, 3, and 5.

3.2.1. Case Study 2: Dataset Generation

The dataset for this case study consists of one matrix (see Figure 9), which is used to train the Algorithm 2 (described below) and contains sensor data (10 different sensors per each sensing node of the WSN). This matrix simulates the 64 possible park conditions (i.e., the condition 0—no sensor

alarm—together with the 63 sensor alarm conditions defined above) and a number from 0 to 63 that represents all the possible park conditions. In more detail, the matrix was created using a MATLAB script that generates, for each sensor, a suitable value in the range of operations using a random function with a Gaussian distribution. Furthermore, some of the data is "faulted" to inject alarm conditions. The matrix contains 32,000 records of data (10 sensors for each of the six sensor nodes) and has been randomized and standardized during the preprocessing task. The last column of the matrix represents, for each record, the labeled alarm condition.

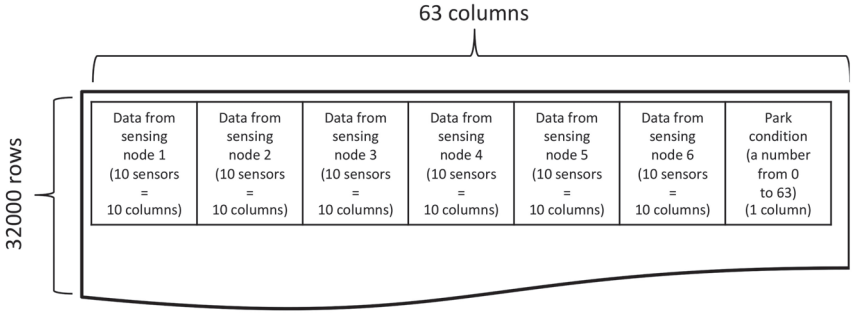


Figure 9. Table description of the input matrix used by the algorithm for emergency management.

Note that every row of the above matrix is a complete record of a single sensors' data acquisition and has a 1D grid-like structure (1×63). Importantly, the position of every column is fixed, i.e., containing a specific order of the sensor data (accelerometer, temperature, humidity, etc.). Hence, every row is a different set of sensors' data in a specific column position for a single acquisition. In this way, the NN is able to learn the relationship between data related to each alarm condition, using every single row as a full set of information.

3.2.2. Case Study 2: Algorithms

As described above (cf. description of Figure 3), in an emergency scenario the Algorithm 2; Algorithm 3 are used. In particular, Algorithm 2 is a clustering NN, which was used in this study to detect environmental/structural alarm from the above matrix and has the following characteristics: (i) Input size = 63; (ii) Output size = 64; (iii) Hidden layers = 55; and (iv) Batch size = 50.

Furthermore, Algorithm 2 uses the Stochastic Gradient Descent (SGD) as an optimizer function (i.e., an iterative method for optimizing an objective function with suitable smoothness properties). Meanwhile, the activation function REctified Linear Unit (relu) is used for hidden layers (relu is the most commonly used activation function in neural networks, especially in NNs). The output layer uses the Softmax function, which takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. The outputs of the algorithm are the nodes in which the alarms are not active. Using this information, the definition of the graph G is modified accordingly, thus eliminating the unsafe nodes and the possibility of passing through a damaged infrastructure, creating a simplified graph G^*.

Moreover, data from Algorithm 1 are also used to update the weights of the graph connections. If an environmental or structural alarm is detected by Algorithm 2, the Algorithm 3 is used to define an Emergency plan, using the graph G^* as inputs. In this case, the "digraph" function [46] is automatically activated to derive the fastest and safest paths that the people near to each node must follow to reach an exit (if available) or an assembly point (if the exits are closed).

4. Results and Discussions

Algorithm 1 was tested using Anaconda, which is a free and open-source distribution of the Python and R programming languages for scientific computing. The performances were estimated

using several prerecorded videos that compared the output to the manual counting of the people, showing an accuracy above 90%.

Algorithm 2 was tested using Keras, a deep-learning library that allows for easy and fast prototyping. Algorithm 2 showed a test accuracy of about 85%.

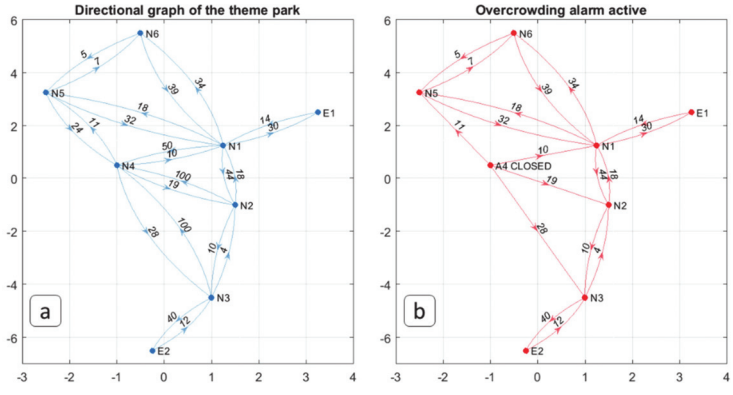
Algorithm 3 (see Figure 10) was tested using Matlab, taking as inputs the output of Algorithms 1 and 2 and modifying, accordingly to the above results, the G graph. When it is executed with data coming from Algorithm 1, the Algorithm 3 shows the areas that are closed because of the activation of the overcrowding alarm. In the case of data coming from Algorithms 1 and 2, the Algorithm 3's output represents the safest and fastest path to reach an exit or assembly point in case of active environmental or structural alarms.

(a)Algorithm 3 (part 1): Crowd Management	**(b) Algorithm 3 (part 2): Emergency Management**
1) Create a weighted graph using the Matlab function "digraph";	1) Create a weighted graph using the Matlab function "graph";
2) Define the array "thresholds";	2) Import the array "EMERGENCY" from the Algorithm 2;
3) For each area:	3) Find the targets of the algorithm from the exits and the assembly points based on the array above;
3.1) Count the people coming in (P_IN) and out (P_OUT) the area based on the weights of the graph;	4) Find the node(s) Ni (with i=1,…,6 when alarms are detected in all the nodes) where the alarm(s) was/were detected;
3.2 Calculate the people inside each area (P_IN-P_OUT);	For each node in which the alarm is active:
3.3 If P_IN > P_OUT and P_IN > threshold, trigger the overcrowding alarm at the i-th area;	5.1) Create a new graph Matlab function "graph" indicating active alarm(s) and closed exit(s);
3.4) Close the area removing the edges directed to the area;	5.2) Find the possible safe paths with the Matlab function "shortestpath";
4) Create a new graph where active alarm(s) and closed area(s) are indicated.	5.3) Select the fastest path among the paths above using the weights;
	5) For each node in which the alarm does not active: repeat the steps from 6.1 to 6.3;
	6) Create a new graph where active alarm(s) and closed exit(s) are indicated.

Figure 10. Pseudo-code of the Algorithm 3 for: (**a**) crowd and (**b**) emergency management.

The following figures (Figures 11 and 12) show the results of the Algorithm 1 in two simulated cases of overcrowding alarm according to those shown by Figure 6. Note that Figure 11a is an example of a graph of the theme park without alarm, while Figure 11b is an example of a graph with alarm. The latter was automatically generated because of the activation of the overcrowding alarm at node 4, which led to the closing of the area number 4 (please note that the name of node 4 changed from "N4" to "A4 CLOSED"). Meanwhile, Figure 11c is a table that shows the overall conditions and that consists in the columns: (i) "Area", which reports the six areas into which the theme park was divided; (ii) "People_in", which reports the number of people entering in each area (which will be provided by the Algorithm 1 in the real application of the IoT system); (iii) "People_out", which reports the number of outgoing people from each area (provided by the Algorithm 1 in the real application); (iv) "People_inside", which reports the number of people that are inside each area at the time t_1 (when the crowd management plan was generated), which was calculated by subtracting the entering and outgoing people in relation to each area (note that only the positive values are reported, while the null and the negative ones are represented with zeros); (v) "Threshold", which reports the maximum number of people allowed to stay at the same time in each area; and (vi) "Alarm", which can be "NO" and "ACTIVE" if the overcrowding alarm is inactive and active, respectively. Note that, when an area

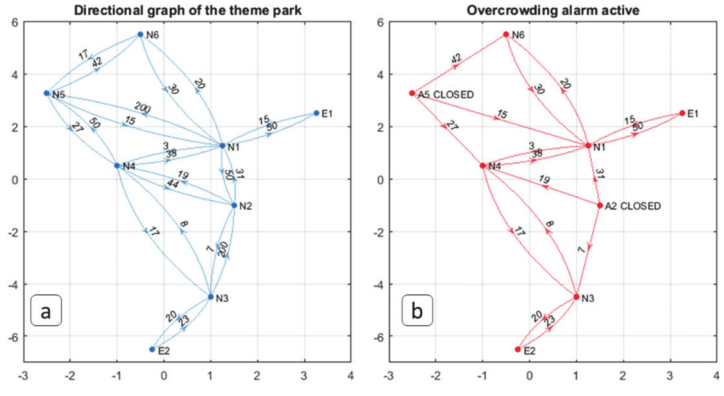
is closed, it is only possible to leave the area (see the arrows on the graphs), and specific messages, shown by the digital signages installed in the park, prevent the access to the closed area.



c

Area	People_in	People_out	People_inside	Threshold	Alarm
'Area#1'	113	176	0	50	'NO'
'Area#2'	67	128	0	120	'NO'
'Area#3'	50	144	0	150	'NO'
'Area#4'	274	68	206	200	'ACTIVE'
'Area#5'	34	63	0	180	'NO'
'Area#6'	41	44	0	100	'NO'

Figure 11. Example of the results of the Algorithm 3 in case of an overcrowding alarm at node 4, where (**a**) is the graph without an alarm, (**b**) is the graph in case of an alarm, and (**c**) is a table showing the overall condition.



c

Area	People_in	People_out	People_inside	Threshold	Alarm
'Area#1'	129	323	0	50	'NO'
'Area#2'	294	57	237	120	'ACTIVE'
'Area#3'	47	228	0	150	'NO'
'Area#4'	57	149	0	200	'NO'
'Area#5'	267	84	183	180	'ACTIVE'
'Area#6'	62	47	15	100	'NO'

Figure 12. Example of the results of the Algorithm 3 in case of an overcrowding alarm at nodes 2 and 5, where (**a**) is the graph without an alarm, (**b**) is the graph in case of an alarm, and (**c**) is a table that shows the overall condition.

As in Figure 11, Figure 12 shows the results of the Algorithm 3 in case of an overcrowding alarm, but, in this case, this alarm was detected by nodes 2 and 5, contemporaneously.

The next figures (Figures 13–18) show two examples of the Algorithm 3's output, which were obtained by carrying out two simulations according to Figure 8 (i.e., environmental/structural alarm active at node 3 (see Figures 13–15) and at nodes 1, 3, and 5 (Figures 16–18)).

In particular, Figure 13a shows the weighted graph (the weights are the people on the path between two nodes) of the theme park without an alarm, while Figure 13b shows the graph in case of an alarm. In more detail, as shown by Figure 13b, when an alarm is detected by the Algorithm 2 at an i-th node (node 3 in this case), the Algorithm 3: (1) changes the name of the i-th node in the graph to point out the node in which the alarm is active (see "N3 ALARM" in Figure 13b); (2) removes the edge between the i-th node and the other node without an alarm; (3) if the i-th nodes are those that are near one exit (i.e., nodes 1 and 3), the name of the exit changes as well (see "E2 CLOSED" in Figure 13b).

Note that, in order to improve the clarity of the figure below, the position of node 6 was defined so as to show the node 6-node 3 connection in a better way, and that, in case of an alarm, the weights were removed from the related graphs.

Figures 14 and 15 show the safest (i.e., that do not include nodes with active alarms) and fastest (i.e., that consider the people on the edges as a resistance to the crowd flow) paths. These paths can be used by the people near each node of the system (from node 1, N1, to node 6, N6) to face the environmental/ structural emergency detected by node 3 (N3). Importantly, the Algorithm 3 also provides a path for people that are near N3 where the alarm was detected. In addition, a table that summarizes the overall conditions is reported in Figure 15c. This table shows: (i) the "Start Node", which is the node from which the paths start; (ii) the "End Node", which is the final destination of the people reaching a safe place (i.e., an assembly point); (iii) the safest and fastest path, consisting of the edges between the i-th node (Ni) and the n-th node (Ni+3); (iv) the node(s) where the alarm(s) was/were detected.

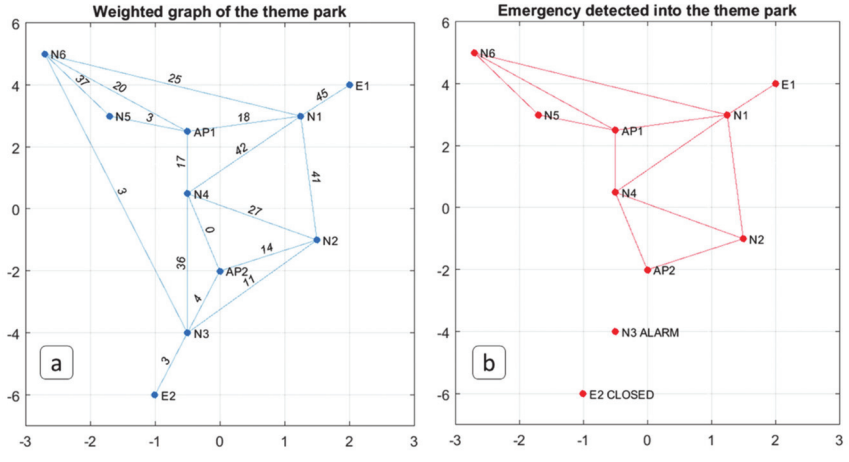


Figure 13. Example of the results of the Algorithm 3 in case of an environmental/structural alarm at node 3 (N3), where (**a**) is the graph before the alarm triggering and (**b**) is the graph after the alarm triggering.

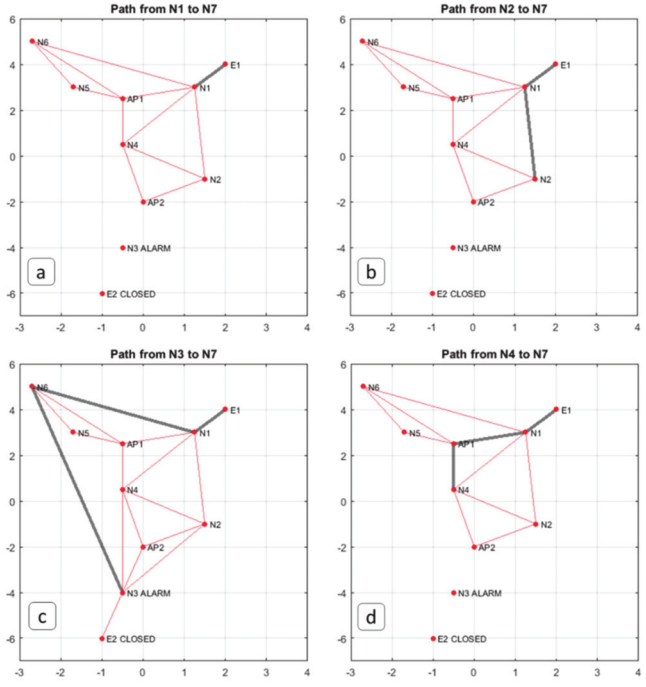
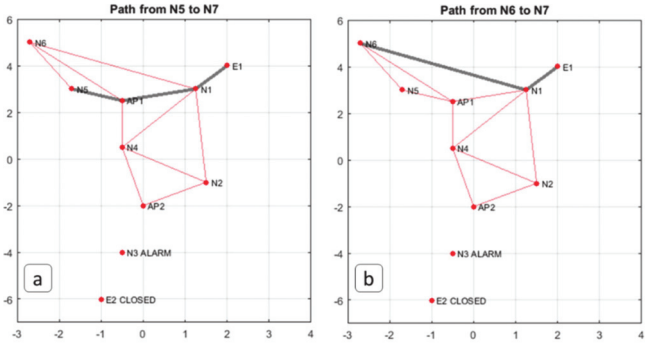


Figure 14. Example of the safest and fastest paths provided by the Algorithm 3 in case of an environmental/structural alarm at node 3 (N3), which can be used by people at (**a**) node 1 (N1), (**b**) node 2 (N2), (**c**) node 3 (N3), and (**d**) node 4 (N4) to reach the exit 1 (E1 = N7).



c

Start_Node	Ni	Ni+1	Ni+2	Ni+3	End_Node	Alarm
'Node 1'	1	7	0	0	'Exit 1'	'NO'
'Node 2'	2	1	7	0	'Exit 1'	'NO'
'Node 3'	3	6	1	7	'Exit 1'	'ACTIVE'
'Node 4'	4	9	1	7	'Exit 1'	'NO'
'Node 5'	5	9	1	7	'Exit 1'	'NO'
'Node 6'	6	1	7	0	'Exit 1'	'NO'

Figure 15. Example of the safest and fastest paths provided by the Algorithm 3 in case of an environmental/structural alarm at node 3 (N3), for people at (**a**) nodes 5 (N5) and (**b**) 6 (N6), and (**c**) is a table that shows the overall condition.

Finally, Figures 16–18 report a second example of the application of the Algorithm 3, which was derived according to that shown in Figure 8b. In this case, nodes 1, 3, and 5 detected an alarm.

Consequently, the exits 1 and 2 (herein called E1 and E2) were closed, and the target of the Algorithm 3 became the Assembly Points 1 and 2 (that are also indicated as nodes 9 and 10, or N9 and N10, in the figure below).

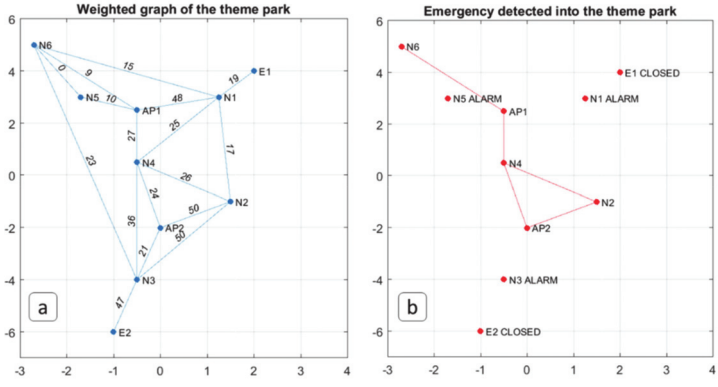


Figure 16. Example of the results of the Algorithm 3 in case of an environmental/structural alarm at nodes 1, 3, and 5 (N1, N3, and N5), where (**a**) is the graph before the alarm triggering and (**b**) is the graph after the alarm triggering.

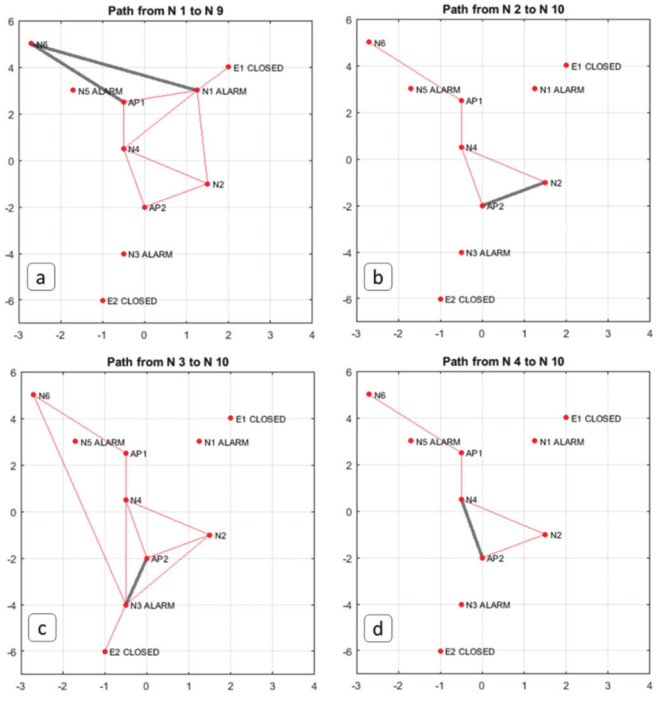
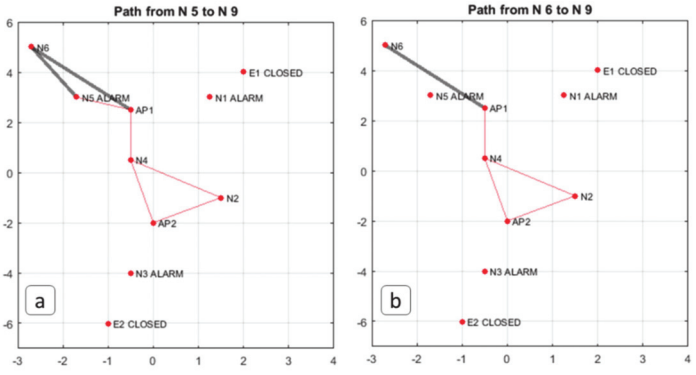


Figure 17. Example of the safest and fastest paths provided by the Algorithm 3 in case of an environmental/structural alarm at nodes 1, 3, and 5 (N1, N3, and N5), which can be used by the people at (**a**) node 1 (N1), (**b**) node 2 (N2), (**c**) node 3 (N3), and (**d**) node 4 (N4) to reach the Assembly Points 1 and 2 (AP1 and AP2, which are also nodes 9, N9, and 10, N10).



c

Start_Node	Ni	Ni+1	Ni+2	Ni+3	End_Node	Alarm
'Node 1'	1	6	9	0	'Ass. Point 1'	'ACTIVE'
'Node 2'	2	10	0	0	'Ass. Point 2'	'NO'
'Node 3'	3	10	0	0	'Ass. Point 2'	'ACTIVE'
'Node 4'	4	10	0	0	'Ass. Point 2'	'NO'
'Node 5'	5	6	9	0	'Ass. Point 1'	'ACTIVE'
'Node 6'	6	9	0	0	'Ass. Point 1'	'NO'

Figure 18. Example of the safest and fastest paths provided by the Algorithm 3 in case of an environmental/structural alarm at nodes 1, 3, and 5 (N1, N3, and N5), for the people at (**a**) nodes 5 (N5) and (**b**) 6 (N6), and (**c**) is a table that shows the overall condition.

Note that Table 6 and Equation 2 report the results of a scalability analysis carried out considering a system consisting in 6, 60, 240, 480, 960 and 1200 nodes, two exits and two assembly points. Alarms were simulated considering that, in each case, 20% of the nodes (randomly selected and indicated in the following with "A") triggered an emergency alarm. The results refer to one of the worst conditions, i.e., the definition of the safest and shortest path to reach the exit 2 from node 1 (cf. Figure 2a) considering that: (i) node 1 is adjacent to the exit 1; (ii) the exit 1 is closed; and (iii) the people around node 1 have to cross the whole park while avoiding all the emergencies (i.e., the nodes "A") to reach the exit 2. In each of the six abovementioned cases, the system's response times, using a desktop PC equipped with Intel® Core™ i5-7400 CPU @ 3.00GHz, 8.00 GB of RAM, were calculated by averaging the results of 50 simulations for each case. The results of the scalability analysis described above are reported in Table 6.

Table 6. Results of the scalability analysis.

Condition	# of Node That Make Up the System	# Nodes in Alarm (i.e., 20% of All the Nodes)	Average Time of Response of the Algorithm 3 (s)
1	6	1	0.5513
2	60	12	0.9253
3	240	48	1.7147
4	480	96	2.9778
5	960	192	5.9606
6	1200	240	7.3738

The results in Table 6 were used to define the law that better describes how the response time of the algorithm 3 decays if the number of nodes increases (Equation (2) was derived).

$$t = 0.1706 \cdot N^{0.487}, \quad (2)$$

where *t* is the response time of the algorithm 3 (seconds) as a function of the number of sensing nodes *N*.

Finally, all the algorithms have been developed and tested in a desktop PC and in a Raspberry PI, showing a computational time compatible with the level of risk associated with the safety application (<3 sec) in the case of actual nodes. The possibility to run a Raspberry PI on a Wireless Local Area Network (WLAN) is also powerful in the case of a network communication interruption, as the algorithms can be implemented in low computational and standalone devices.

Future works include (1) the implementation on low computational devices (e.g., microcontrollers), (2) the comparison with other simple neural network classifiers, such as a K-Means algorithm or a tree-based method (e.g., random forest or decision tree), or Convolutional Neural Networks and (3) the usage of a "flow graph" created by means of a min-cut max flow algorithm in order to maximize the number of people per unit of time who can move out of the park (e.g., using the number of people per unit of time who can use the path as capacities, and the actual number of people who are expected to use it at any given time based on a decision for evacuation as a flow value. In this way, crowds can be redirected from all the nodes simultaneously, and the flow is maximized).

5. Conclusions

In this study, an IoT system consisting of a system and algorithms that allow environmental and structural monitoring and emergency management is presented. Using powerful Machine Learning and Shortest Path Finding algorithms, fed with simulated datasets, useful information is gathered from sensors (environmental and structure-related), cameras, and using NFC technology (number and location of visitors) in order to obtain an alarm detection, safe path suggestion, and social distancing alarm for users and platform managers. Raw and elapsed data are sent to a platform dashboard for the additional online monitoring of environmental and structural conditions. The implementation is envisioned on low computational devices such as microcontrollers, which are also powerful in the case of a network communication interruption because they do not require a remote data exchange.

Author Contributions: Conceptualization, M.M. and R.F.; methodology, M.M.; software, R.F.; validation, M.M. and R.F.; investigation, R.F.; writing—original draft preparation, R.F.; writing—review and editing, M.M.; supervision, M.M. 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. Bibri, S.E. The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. *Sustain. Cities Soc.* **2018**, *38*, 230–253. [CrossRef]
2. Fan, C.; Zhang, C.; Yahja, A.; Mostafavi, A. Disaster City Digital Twin: A vision for integrating artificial and human intelligence for disaster management. *Int. J. Inf. Manag.* **2019**, (in press). [CrossRef]
3. Sakurai, M.; Murayama, Y. Information technologies and disaster management – Benefits and issues. *Prog. Disaster Sci.* **2019**, *2*, 100012. [CrossRef]
4. Bacco, M.; Delmastro, F.; Ferro, E.; Gotta, A. Environmental Monitoring for Smart Cities. *IEEE Sens. J.* **2017**, *17*, 7767–7774. [CrossRef]
5. Government of South Australia COVID-19. Available online: https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/how-to-protect-yourself-and-others-from-coronavirus-covid-19/limits-on-public-gatherings-for-coronavirus-covid-19 (accessed on 14 April 2020).
6. Caley, P.; Philp, D.J.; McCracken, K. Quantifying social distancing arising from pandemic influenza. *J. R. Soc. Interface* **2008**, *5*, 631–639. [CrossRef]
7. Stein, R.A. COVID-19 and rationally layered social distancing. *Int. J. Clin. Pract.* **2020**, *74*, e13501. [CrossRef]
8. Merenda, M.; Porcaro, C.; Iero, D. Edge machine learning for ai-enabled iot devices: A review. *Sensors* **2020**, *20*, 2533. [CrossRef]
9. Nakas, C.; Kandris, D.; Visvardis, G. Energy efficient routing in wireless sensor networks: A comprehensive survey. *Algorithms* **2020**, *13*, 72. [CrossRef]

10. Jin, Z.; Jian-Ping, Y.; Si-Wang, Z.; Ya-Ping, L.; Guang, L. A survey on position-based routing algorithms in wireless sensor networks. *Algorithms* **2009**, *2*, 158–182. [CrossRef]
11. Hedar, A.R.; Abdulaziz, S.N.; Sewisy, A.A.; El-Sayed, G.A. Adaptive scatter search to solve the minimum connected dominating set problem for efficient management of wireless networks. *Algorithms* **2020**, *13*, 35. [CrossRef]
12. Meghanathan, N. A, Benchmarking algorithm to determine minimum aggregation delay for data gathering trees and an analysis of the diameter-aggregation delay tradeoff. *Algorithms* **2015**, *8*, 435–458. [CrossRef]
13. El Khamlichi, Y.; Tahiri, A.; Abtoy, A.; Medina-Bulo, I.; Palomo-Lozano, F.A. Hybrid algorithm for optimal wireless sensor network deployment with the minimum number of sensor nodes. *Algorithms* **2017**, *10*, 80. [CrossRef]
14. Erd, M.; Schaeffer, F.; Kostic, M.; Reindl, L.M. Event monitoring in emergency scenarios using energy efficient wireless sensor nodes for the disaster information management. *Int. J. Disaster Risk Reduct.* **2016**, *16*, 33–42. [CrossRef]
15. Rahman, M.U.; Rahman, S.; Mansoor, S.; Deep, V.; Aashkaar, M. Implementation of ICT and Wireless Sensor Networks for Earthquake Alert and Disaster Management in Earthquake Prone Areas. Proceedings of International Conference on Computational Modeling and Security, Bangalore, India, 11–13 February 2016; Volume 85, pp. 92–99.
16. Erdelj, M.; Król, M.; Natalizio, E. Wireless Sensor Networks and Multi-UAV systems for natural disaster management. *Comput. Netw.* **2017**, *124*, 72–86. [CrossRef]
17. Jain, B.; Brar, G.; Malhotra, J.; Rani, S. A novel approach for smart cities in convergence to wireless sensor networks. *Sustain. Cities Soc.* **2017**, *35*, 440–448. [CrossRef]
18. Alam, S.; De, D. Bio-inspired smog sensing model for wireless sensor networks based on intracellular signalling. *Inf. Fusion* **2019**, *49*, 100–119. [CrossRef]
19. Uma Priyadarsini, P.S.; Sriramya, P. Disaster management using evidence-based interactive trust management system for wireless sensor networks by Internet of Things. *Comput. Electr. Eng.* **2019**, *75*, 164–174. [CrossRef]
20. Deak, G.; Curran, K.; Condell, J.; Asimakopoulou, E.; Bessis, N. IoTs (Internet of Things) and DfPL (Device-free Passive Localisation) in a disaster management scenario. *Simul. Model. Pract. Theory* **2013**, *35*, 86–96. [CrossRef]
21. Cui, F. Deployment and integration of smart sensors with IoT devices detecting fire disasters in huge forest environment. *Comput. Commun.* **2020**, *150*, 818–827. [CrossRef]
22. Pillai, A.S.; Chandraprasad, G.S.; Khwaja, A.S.; Anpalagan, A. A service oriented IoT architecture for disaster preparedness and forecasting system. *IoT* **2019**, 100076. (in press). [CrossRef]
23. Tao, Z. Advanced Wavelet Sampling Algorithm for IoT based environmental monitoring and management. *Comput. Commun.* **2020**, *150*, 547–555. [CrossRef]
24. Rathore, M.M.; Paul, A.; Hong, W.H.; Seo, H.C.; Awan, I.; Saeed, S. Exploiting IoT and big data analytics: Defining Smart Digital City using real-time urban data. *Sustain. Cities Soc.* **2018**, *40*, 600–610. [CrossRef]
25. Chen, W.; He, B.; Zhang, L.; Nover, D. Developing an integrated 2D and 3D WebGIS-based platform for effective landslide hazard management. *Int. J. Disaster Risk Reduct.* **2016**, *20*, 26–38. [CrossRef]
26. Aloi, G.; Briante, O.; Di Felice, M.; Ruggeri, G.; Savazzi, S. The SENSE-ME platform: Infrastructure-less smartphone connectivity and decentralized sensing for emergency management. *Pervasive Mob. Comput.* **2017**, *42*, 187–208. [CrossRef]
27. Qiu, L.; Du, Z.; Zhu, Q.; Fan, Y. An integrated flood management system based on linking environmental models and disaster-related data. *Environ. Model. Softw.* **2017**, *91*, 111–126. [CrossRef]
28. Damalas, A.; Mettas, C.; Evagorou, E.; Giannecchini, S.; Iasio, C.; Papadopoulos, M.; Konstantinou, A.; Hadjimitsis, D. Development and Implementation of a DECATASTROPHIZE platform and tool for the management of disasters or multiple hazards. *Int. J. Disaster Risk Reduct.* **2018**, *31*, 589–601. [CrossRef]
29. Pérez-González, C.J.; Colebrook, M.; Roda-García, J.L.; Rosa-Remedios, C.B. Developing a data analytics platform to support decision making in emergency and security management. *Expert Syst. Appl.* **2019**, *120*, 167–184. [CrossRef]
30. Ma, G.; Wu, Z. BIM-based building fire emergency management: Combining building users' behavior decisions. *Autom. Constr.* **2020**, *109*, 102975. [CrossRef]
31. Franke, T.; Lukowicz, P.; Blanke, U. Smart crowds in smart cities: Real life, city scale deployments of a smartphone based participatory crowd management platform. *J. Internet Serv. Appl.* **2015**, *6*, 1–19. [CrossRef]

32. Palmieri, F.; Ficco, M.; Pardi, S.; Castiglione, A. A cloud-based architecture for emergency management and first responders localization in smart city environments. *Comput. Electr. Eng.* **2016**, *56*, 810–830. [CrossRef]
33. Alazawi, Z.; Alani, O.; Abdljabar, M.B.; Altowaijri, S.; Mehmood, R. A smart disaster management system for future cities. In Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities, Co-Located with MobiHoc, Philadelphia, PA, USA, 11–14 August 2014; pp. 1–10.
34. Astarita, V.; Festa, D.C.; Giofrè, V.P.; Guido, G.; Stefano, G. Mobile for emergencies M4EM: A cooperative software tool for emergency management operations. *Procedia Comput. Sci.* **2018**, *134*, 433–438. [CrossRef]
35. He, Y.; Zhang, D.; Fang, Y. Development of a mobile post-disaster management system using free and open source technologies. *Int. J. Disaster Risk Reduct.* **2017**, *25*, 101–110. [CrossRef]
36. Kaku, K. Satellite remote sensing for disaster management support: A holistic and staged approach based on case studies in Sentinel Asia. *Int. J. Disaster Risk Reduct.* **2019**, *33*, 417–432. [CrossRef]
37. Hiltz, S.R.; Hughes, A.L.; Imran, M.; Plotnick, L.; Power, R.; Turoff, M. Exploring the usefulness and feasibility of software requirements for social media use in emergency management. *Int. J. Disaster Risk Reduct.* **2020**, *42*, 101367. [CrossRef]
38. Loureiro, A.L.D.; Miguéis, V.L.; da Silva, L.F.M. Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decis. Support Syst.* **2018**, *114*, 81–93. [CrossRef]
39. Kor, A.L.; Yanovsky, M.; Pattinson, C.; Kharchenko, V. SMART-ITEM: IoT-enabled smart living. In Proceedings of the FTC 2016—Proceedings of Future Technologies Conference, San Francisco, CA, USA, 6–7 December 2016; pp. 739–749.
40. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 5. [CrossRef]
41. Kunaver, M.; Požrl, T. Diversity in recommender systems—A survey. *Knowl. Based. Syst.* **2017**, *123*, 154–162. [CrossRef]
42. Di Martino, S.; Rossi, S. An Architecture for a Mobility Recommender System in Smart Cities. *Procedia Comput. Sci.* **2016**, *58*, 425–430. [CrossRef]
43. Lujak, M.; Billhardt, H.; Dunkel, J.; Fernández, A.; Hermoso, R.; Ossowski, S. A distributed architecture for real-time evacuation guidance in large smart buildings. *Comput Sci. Inf. Syst.* **2017**, *14*, 257–282. [CrossRef]
44. Saleem, Y.; Crespi, N.; Rehmani, M.H.; Copeland, R.; Hussein, D.; Bertin, E. Exploitation of social IoT for recommendation services. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things, WF-IoT, Reston, VA, USA, 12–14 December 2016; pp. 359–364.
45. Vijayalakshmir, D.; Kalaivani, R. Minimum Cost Spanning Tree using Matrix Algorithm. *Int. J. Sci. Res. Publ.* **2014**, *4*, 1–5.
46. Goyal, A.; Mogha, P.; Luthra, R.; Sangwan, N. Path finding: A* or Dijkstra's? . *Int. J. IT Eng.* **2013**, *2*, 1–15.
47. Khuller, S.; Raghavachari, B. Graph and Network Algorithms. Available online: https://it.mathworks.com/help/matlab/graph-and-network-algorithms.html?s_tid=CRUX_lftnav (accessed on 20 September 2020).
48. NFC Forum NFC Technology. Available online: https://nfc-forum.org/what-is-nfc/ (accessed on 20 September 2020).
49. Freire, S. Modeling of spatiotemporal distribution of urban population at high resolution—Value for risk assessment and emergency management. In *Lecture Notes in Geoinformation and Cartography*; Springer: Berlin, Germany, 2010; pp. 53–67.
50. Chan, A.B.; Liang, Z.S.J.; Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Anchorage, AK, USA, 23–28 June 2008.
51. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M. Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics. *Algorithms* **2020**, *13*, 67. [CrossRef]
52. Qu, H.; Wang, M.; Zhang, C.; Wei, Y. A study on faster R-CNN-based subway pedestrian detection with ACE enhancement. *Algorithms* **2018**, *11*, 192. [CrossRef]
53. Solano, F.; Di Fazio, S.; Modica, G. A methodology based on GEOBIA and WorldView-3 imagery to derive vegetation indices at tree crown detail in olive orchards. *Int. J. Appl. Earth Obs. Geoinf.* **2019**, *83*, 101912. [CrossRef]
54. Lanucara, S.; Praticò, S.; Modica, G. Harmonization and interoperable sharing of multi-temporal geospatial data of rural landscapes. *Smart Innov. Syst. Technol.* **2019**, *100*, 51–59.

55. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
56. Zhang, S.H.; Li, R.; Dong, X.; Rosin, P.; Cai, Z.; Han, X.; Yang, D.; Huang, H.; Hu, S.M. Pose2Seg: Detection free human instance segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Volume 2019, pp. 889–898.
57. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 10778–10787.
58. Bengio, Y.; Goodfellow, I.J.; Courville, A. *Deep Learning: Chapter 9—Convolutional Networks*; MIT Press: Cambridge, MA, USA, 2016.
59. Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *7553*, 436–444. [CrossRef]
60. Fedele, R.; Merenda, M.; Praticò, F.G.; Carotenuto, R.; Della Corte, F.G. Energy harvesting for IoT road monitoring systems. *Instrum. Mes. Metrol.* **2018**, *17*, 605–623. [CrossRef]
61. Merenda, M.; Praticò, F.G.; Fedele, R.; Carotenuto, R.; Corte, F.G. Della A real-time decision platform for the management of structures and infrastructures. *Electronics* **2019**, *8*, 1180. [CrossRef]
62. Merenda, M.; Iero, D.; Carotenuto, R.; Corte, F.G.D. Simple and low-cost photovoltaic module emulator. *Electronics* **2019**, *8*, 1445. [CrossRef]
63. STMicroelectronics IoT board (Model: Kit Discovery B-L475E-IOT01A). Available online: https://www.mouser.it/Search/Refine?Ntk=P_MarCom&Ntt=160178092 (accessed on 20 September 2020).
64. Fedele, R.; Praticò, F.G. Monitoring infrastructure asset through its acoustic signature. In Proceedings of the INTER-NOISE 2019 MADRID, Spain—48th International Congress and Exhibition on Noise Control Engineering, Madrid, Spain, 16–19 June 2019.
65. Fedele, R.; Praticò, F.G.; Carotenuto, R.; Corte, F.G.D. Structural health monitoring of pavement assets through acoustic signature. In Proceedings of the 10th International Conference on the Bearing Capacity of Roads, Railways and Airfields, BCRRA, Athens, Greece, 28–30 June 2017; pp. 869–875.
66. Praticò, F.G.; Fedele, R.; Naumov, V.; Sauer, T. Detection and monitoring of bottom-up cracks in road pavement using a machine-learning approach. *Algorithms* **2020**, *13*, 81. [CrossRef]
67. Praticò, F.G.; Della Corte, F.G.; Merenda, M. Self-powered sensors for road pavements. In Proceedings of the 4th Chinese-European Workshop on Functional Pavement Design, CEW, Delft, The Netherlands, 29 June–1 July 2016; pp. 1365–1374.
68. ThingsBoard. ThingsBoard IoT Open Source Plataform. Available online: https://thingsboard.io/ (accessed on 20 September 2020).
69. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
70. Lukežič, A.; Vojíř, T.; Čehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *Int. J. Comput. Vis.* **2018**, *126*, 671–688. [CrossRef]

algorithms

MDPI

Article

Hierarchical Modelling for CO_2 Variation Prediction for HVAC System Operation

Ibrahim Shaer * and Abdallah Shami *

Department of Electrical and Computer Engineering, Western University, 1151 Richmond St., London, ON N6A 3K7, Canada
* Correspondence: ishaer@uwo.ca (I.S.); abdallah.shami@uwo.ca (A.S.)

Abstract: Residential and industrial buildings are significant consumers of energy, which can be reduced by controlling their respective Heating, Ventilation, and Air Conditioning (HVAC) systems. Demand-based Ventilation (DCV) determines the operational times of ventilation systems that depend on indoor air quality (IAQ) conditions, including CO_2 concentration changes, and the occupants' comfort requirements. The prediction of CO_2 concentration changes can act as a proxy estimator of occupancy changes and provide feedback about the utility of current ventilation controls. This paper proposes a Hierarchical Model for CO_2 Variation Predictions (HMCOVP) to accurately predict these variations. The proposed framework addresses two concerns in state-of-the-art implementations. First, the hierarchical structure enables fine-tuning of the produced models, facilitating their transferability to different spatial settings. Second, the formulation incorporates time dependencies, defining the relationship between different IAQ factors. Toward that goal, the HMCOVP decouples the variation prediction into two complementary steps. The first step transforms lagged versions of environmental features into image representations to predict the variations' direction. The second step combines the first step's result with environment-specific historical data to predict CO_2 variations. Through the HMCOVP, these predictions, which outperformed state-of-the-art approaches, help the ventilation systems in their decision-making processes, reducing energy consumption and carbon-based emissions.

Keywords: energy consumption reduction; HVAC systems; CO_2 variations prediction; convolutional neural networks; transfer learning; ensemble learning

Citation: Shaer, I.; Shami, A. Hierarchical Modelling for CO_2 Variation Prediction for HVAC System Operation. *Algorithms* **2023**, *16*, 256. https://doi.org/10.3390/a16050256

Academic Editors: Gloria Cerasela Crisan, Ha Duy Long and Elena Nechita

Received: 27 April 2023
Revised: 12 May 2023
Accepted: 15 May 2023
Published: 17 May 2023

1. Introduction

The amount of energy consumed by buildings and industrial facilities accounts for around 70% of the total energy consumption in the US [1], and it constitutes a primary source of energy consumption in the majority of European countries [2]. Heating, Ventilation, and Air Conditioning (HVAC) systems represent a controllable energy consumer that fluctuates based on the thermal needs and occupants' comfort requirements [3]. Maintaining indoor air quality (IAQ) is an essential requirement for occupants due to the downstream effect of poor ventilation on the occupants' health [4]. An array of health-related problems arises with poor ventilation, such as decreased productivity [4,5] and increased risks of COVID-19 infections [6,7] which can reduce a human's quality of life [8]. Examples of these potential health-related hazards are encountered on a daily basis. Vehicle-based indoor contaminant levels increase in houses and buildings with indoor garages, increasing the risk of developing cancer and inducing fires [9,10]. Moreover, households within industrial cities suffer from elevated levels of heavy metal contamination, which contributes to health-related problems [11]. In a similar context, Liao et al. [12] uncovered a direct association between extended exposure to cooking oil fumes and deterioration in sleep quality. These dire consequences can be tackled with proper ventilation systems that provision fresh outdoor air to facilitate indoor air circulation [13].

The increased buildings' energy bills due to increased oil prices coupled with strict legislation to stifle Greenhouse Emissions (GHG) and the comfort needs of occupants are pushing the HVAC system operators to a critical juncture [14]. As a result, there is a need to match the occupants' comfort requirements while reducing the energy consumption of the HVAC systems by properly activating them. The demand-controlled ventilation (DCV) methods alter the ventilation systems based on the ventilation demands of the indoor environment. Since these demands are centred around the occupants' requirements, the prediction of the occupant's number changes is essential for controlling the ventilation loads. The regulatory bodies for efficient building development in North America and Europe [15] did not reach a consensus on concrete strategies or methods to utilize for measuring occupancy. In that regard, the literature and practical implementations suggest two main strategies [16]. The first method involves inferring the number of occupants by video surveillance cameras or the collection of occupants' identities, which raises some privacy concerns. The second method depends on proxy estimators of changes in occupancy using either motion and thermal energy sensors [17], Wi-Fi signals [18], or CO_2 levels [19], or a combination of sensors.

The collection of ground truth data to reflect the number of occupants is infeasible in real time; thus, resorting to proxy estimators is a more practical approach with the already existing sensor infrastructure. In regard to proxy estimators, changes in occupancy can be estimated using Wi-Fi signals, CO_2 concentration level changes, and PassiveInfrared (PIR) sensors. However, employing Wi-Fi signals or activity levels to that end hinges on very loose assumptions. The Wi-Fi signals pre-suppose that occupants are connected to access points or hold a single device [20] while the PIR sensors are oblivious to static occupants and can potentially overestimate the number of occupants [21]. These factors favour CO_2 concentration changes that vary with the occupancy changes, which either stem from changes in their number or their activity, both with similar effects to the ventilation demands [22].The occupant's respiratory activity directly affects the indoor CO_2 concentrations, which highlights the distinctive property of CO_2 concentrations as a more accurate representation of occupancy changes.

The predictions of CO_2 concentration changes reveal two things about the environment, both signalling changes in the ventilation demands of the indoor environment. The lagged effect of the current collective occupants' respiration can be reflected in the prediction of CO_2 concentration changes. Similarly, due to its hysteresis property, the prediction of CO_2 concentration changes shows the lagged effect of the current ventilation controls on indoor CO_2 concentrations [20]. Therefore, these predictions serve to piece together different parts of the indoor environment that can aid DCV in producing an informed decision about its activity. As a result, the optimization of HVAC system energy consumption while maintaining the indoor occupants' comfort is a downstream effect of accurate predictions of the variation in CO_2 concentrations [23]. The reduction in HVAC system energy consumption aligns with the overall sustainability goals of reducing the buildings' carbon footprint.

The abundance of computing resources and the miniaturization of sensors have promoted the implementation of Machine Learning (ML) and Deep Learning (DL) applications in estimating indoor environmental conditions using different sensors [24]. The state-of-the-art approaches to predicting occupancy lack the incorporation of time dependencies or suffer from transferability issues. Therefore, to address these issues, this paper proposes a Hierarchical Model for CO_2 Variation Predictions (HMCOVP), which divides the problem of predicting CO_2 variations into two complementary sub-problems, each solved using supervised Machine Learning (ML) techniques. The first sub-problem uses lagged versions of time-series environmental features transformed into images to predict the variations' future direction, which can be increasing, decreasing, or constant. The second sub-problem combines the predictions of the first step with the historical difference between the minimum and maximum value (e2s) of the feature to predict the variations in CO_2 concentrations in a pre-defined time window. The proposed framework serves to achieve two main goals.

The first goal is to accurately predict the CO_2 variations. The second goal is to facilitate the transferability of the model to other buildings or rooms, which is a missing component in state-of-the-art implementations. The second goal is achieved by the framework's hierarchical structure that enables the fine-tuning of specific blocks of the proposed architecture. The HMCOVP is applied to a publicly available dataset [25], encompassing CO_2 levels, humidity, pressure, temperature, and Passive Infrared (PIR) Count, and its transferability is evaluated in different spatial settings.

The main contributions of this paper are as follows:

- Propose a hierarchical framework, including Convolutional Neural Networks (CNNs), transfer learning, and supervised learning that accurately predicts CO_2 variations to serve as proxy estimators of occupancy and provide feedback about the utility of the current ventilation system controls;
- Utilize a novel time-series-to-image data transformation strategy that reflects the time-correlation aspect of time-series data in general and environmental sensory data in particular;
- Evaluate and compare the proposed approaches with state-of-the-art approaches applied to the same dataset in terms of prediction accuracy using different history and future time windows;
- Evaluate the proposed approach to different office spaces using transfer learning and re-tuning techniques.

The rest of the paper is organized as follows. Section 2 discusses the related work to shed light on the novelty of this manuscript. Section 3 explains some preliminaries pertaining to the dataset. Section 4 describes the different steps of the proposed methodology. Section 5 details the experimental procedure. Section 6 investigates the results of applying the proposed methodology, and Section 7 concludes the paper.

2. Related Work

The field of occupancy estimation through environmental data has incorporated ML and DL techniques. However, the research works differ regarding the features used for estimating the occupancy and the utility of the developed models in connection to HVAC systems.

The work of Dong et al. [26] is flagship research evaluating the usefulness of environmental sensors in capturing occupancy. A feature engineering module is employed to output the most relevant features out of lagged versions of CO_2 concentrations and acoustic levels. The authors report a 73% accuracy in occupancy estimation using Artificial Neural Networks, a Support Vector Machine, and Hidden Markov Models. The authors of [27,28] employ low-complexity ML methods [29] such as Classification And Regression Trees (CART), and Random Forest (RF) for the approach in [27] and an Extreme Learning Machine Classifier for the approach in [28] to estimate occupancy. Candanedo et al. [27] uniquely incorporate the time aspect in their feature engineering steps, whereas the methodology in [28] designs wrapper and filter methods to address the noisiness of the raw sensory data. Similarly, Kallio et al. [30] explored different combinations of lagged indoor environment data to predict CO_2 concentrations and made available the dataset used throughout their paper. To predict CO_2 in different future time horizons, the resultant features are fed to Ridge Regression (RR), Decision Trees (DT), RF, and Multi-Layer Perceptron (MLP) algorithms.

Different types of Deep Neural Networks (DNNs) such as CNNs and Long Short-Term Memory (LSTM) were utilized for occupancy predictions. The work by Chen et al. [24] combines CNNs and bi-directional LSTM to predict different classes of occupancy. The data in [31] incorporate HVAC sensory data with indoor climate sensory data to infer the number of occupants using particle filtering and neural networks. In a similar manner, Li et al. [23] tackle DCV by proactively predicting CO_2 concentrations. The authors of [32] uniquely tackle the transferability of occupancy estimation models using a transfer learning approach. To that end, the authors build their models using data collected in a small

conference room. Later, the developed models are evaluated by transferring them to a large room. This process is achieved because of the layer-wise architecture of their developed CNN and LSTM models.

Some of the literature identified the use of Wi-Fi signals to estimate occupancy. The authors of [18] collect the Channel State Information (CSI) from commercial Internet of Things (IoT) devices. A pipeline of DL models, encompassing Autoencoders and LSTM, is proposed to predict occupancy. Similarly, Wang et al. [33] utilize Wi-Fi usage to predict occupancy trends using RF, DNNs, and LSTMs.

Despite advancing the state-of-the-art in terms of occupancy estimation, there are some limitations to the current literature. A general limitation concerns the absence of publicly available datasets, hindering the comparison or replication of different approaches. The authors of [23,30] broke this trend by relying on the same dataset that is used throughout the paper. In terms of the methodology, time dependencies are not incorporated in most previous works, which constitute a critical aspect in occupancy estimation and HVAC-related operations [34]. We compare our methodology with the applied methodology in [30], which is referred to as Feature Engineering for CO_2 Predictions (FECOP). The limitations of this work cover four main aspects:

- Utility: Their work predicts CO_2 concentrations, such that a value of concentration can drastically vary from one spatial setting to another. For example, a specific prediction value can be interpreted differently in a room with two or 12 people. Mapping CO_2 concentrations to occupancy represents a physical modelling exercise, which varies depending on the studied space. Both these aspects are addressed when predicting the future variations of CO_2 concentrations.
- Feature Engineering: When linked to occupancy, the pressure feature is indicative of invasive airflow introduced by the occupants entering or leaving a specific space. This detail is overlooked by excluding this feature from the feature engineering step. Their methodology involves a tedious feature engineering step, resulting in many extracted features;
- Results: Their reported results are not categorized based on the capacities of each room. This factor is instrumental because of the drastic changes in the relationship between environmental features in different spatio-temporal modalities.
- Transferability: This aspect is missing among most of the state-of-the-art methods. The developed models lack the structural disposition for fine-tuning, which jeopardises their utility in multi-zonal spaces of different capacities or different buildings. This characteristic is instrumental when encountering an environment with a limited amount of data.

This paper addresses these limitations using a framework that decomposes the problem into two complementary sub-problems and encompasses data transformation, transfer learning, hierarchical modelling, and ensemble learning. The data transformation method adheres to the time-dependent nature of the environmental data. Applying transfer learning facilitates the training and testing processes and the transferability of the proposed approach to different spatial resolutions. The ensemble models improve the predictions' accuracy by practically implementing the concept of the crowd's wisdom.

3. Data Preliminaries

This section explores the dataset used for training and testing the developed models and the exploratory data analysis essential for finding the historical and prediction time horizons.

3.1. Dataset Description

The dataset used includes IAQ data collected over a period of one year in 2019 in the Nordic climate. As a result, the dataset includes 22.6 million samples collected from 62 nodes mounted in 13 rooms [25]. The IAQ features include temperature (°C), relative humidity (%), pressure (hecto Pascals), CO_2 concentration (ppm), and activity

levels ranging from 0 to 12. Each sample was captured with one-minute granularity. The activity level was calculated using PIR sensors that aggregate movement levels within a five-second interval. The collected data covered 13 rooms, including 11 cubicles that can fit 2–3 people and 2 meeting rooms that can fit 12 people. The code made available by the authors produces continuous chunks of data in separate datasets, which reduced the total number of samples to 6.1 million datapoints distributed over all the rooms [30]. The steps to obtain these datasets are as follows [30]:

1. Data from different sensors in each room are aggregated;
2. Gaps of less than two minutes are interpolated;
3. Continuous data samples of high variability in CO_2 levels are extracted as testing set to evaluate the developed methodology.

3.2. Exploratory Data Analysis

The exploratory data analysis is fundamental to estimating the prediction and history time windows predicated on the studied space. These time frames best reflect the existing relationship between the CO_2 concentration levels, and the lagged versions of activity levels, humidity, pressure, temperature, and CO_2 levels. This analysis is important to validate the existence and the extent of time correlations between IAQ features; a characteristic that rationalises any modelling-related decision.

Many methods can be applied to quantify the relationship between lagged versions of IAQ features and the CO_2 concentration. The original work's feature engineering step extracted lagged versions of features to predict future CO_2 using ML techniques such as RR. Good results with a low Mean Absolute Error (MAE) were obtained using RR, which assumes a linear relationship between the features and the response variable. Therefore, a linear function can successfully quantify the relationship between lagged versions of the environment features and the CO_2 concentrations. These assumptions align with the guiding principles of the Pearson Correlation Coefficient (PCC), which measures the strength of the linear relationship between two continuous variables.

The environmental features and their lagged versions are denoted by X_i and the CO_2 level by C. Given an environmental feature X_i and C, the PCC represented by ρ is as follows:

$$\rho_{X_i,C} = \frac{\mathbf{E}[(X_i - \mu_{X_i})(C - \mu_C)]}{\sigma_{X_i}\sigma_C} \tag{1}$$

such that:

- $\mathbf{E}$ is the expectation
- μ_{X_i} is the mean of X_i
- μ_C is the mean of C
- σ_{X_i} is the standard deviation of X_i
- σ_C is the standard deviation of C

The analysis that follows is conducted on a single room denoted by room00, which fits 12 people. The lag time distribution with respect to each dataset when the correlation between CO_2 and other features is below a certain threshold is displayed in Figure 1. This calculation allows inferring the time window whereby a lagged feature can explain the changes in CO_2 concentrations. The correlation thresholds involved are $\{0.2 \longrightarrow 1\}$ with a 0.1 increment. If the threshold is not found, meaning the dataset has a higher correlation than the threshold, a value of -10 is returned to emphasize the distinction in the distribution figure.

Figure 1 shows the lag time distribution with respect to each parameter using correlation thresholds. The displayed thresholds are determined based on a comprehensive evaluation of the obtained correlations for each feature. Both activity levels and CO_2 displayed high correlation values of above 0.5 in the majority of the datasets. These results justify the results in [30], whereby the lowest MAE is the result of combining these two features. This observation can be inferred by small peaks for both parameters for lag time 0.

Moreover, the figure shows that in the majority of the datasets, a high positive correlation exists ($\geq$0.5) at a lag time above 20 min, which also includes the lag time of −10. On the other hand, the distribution results demonstrate a weaker positive relationship ($\leq$0.2) with the CO_2 concentrations. The datasets with no occupancy changes have contributed to the dilution of the results with no or low correlation results.

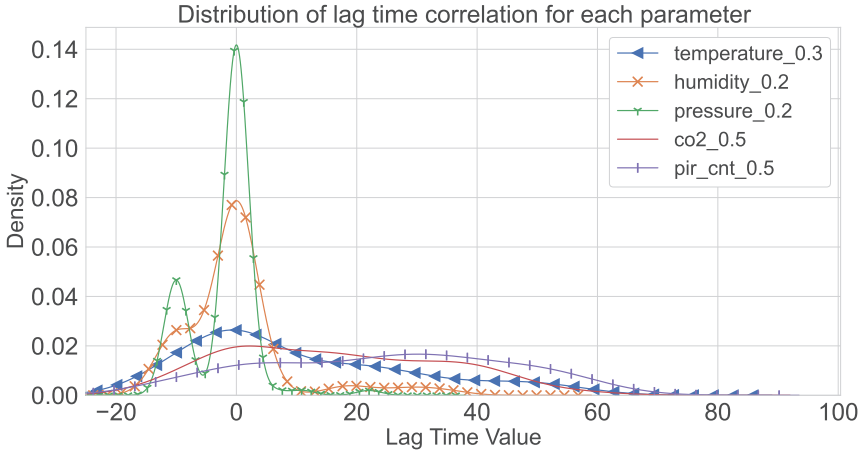


Figure 1. Distribution of Correlations.

These correlation results do not prove that there is no complementarity between the environmental features, rather it determines the reasonable prediction time windows. Therefore, in the applied methodology, all the parameters were used with a variable lag time of 5, 10, 15, and 20 min. These lag times incorporate all the levels of the existing correlations between environmental features. The choice of prediction horizons is also connected to HVAC systems. The prediction horizons of 5–20 min are equivalent to a HVAC control loop that varies within these ranges [23,35,36], dictated by the studied space. As a result, predictions within the pre-defined time horizons serve as indicators of the environment's future state, in terms of CO_2 concentration changes, if the HVAC systems keep their currently defined ventilation and heating/cooling setpoints. The control system can leverage this piece of information to calibrate its subsequent decisions, especially in use cases that depend on data-driven control achieved through Reinforcement Learning techniques.

4. Method: Hierarchical Model for CO_2 Variation Predictions (HMCOVP)

This section describes the steps applied to obtain accurate predictions of the variations in CO_2 concentrations over a future time window. Figure 2 depicts the methodology referred to as the HMCOVP, such that each practical step is detailed based on the input to this step, the operation taking place, and its output. The following subsections explain each step of predicting CO_2 variation direction and predicting CO_2 variations. Time-series to Image Transformation and CNN Individual Learners are the building blocks to predict the CO_2 future direction. Outputs of decreasing CO_2 variation direction and ensemble learners are the foundations for CO_2 variation predictions.

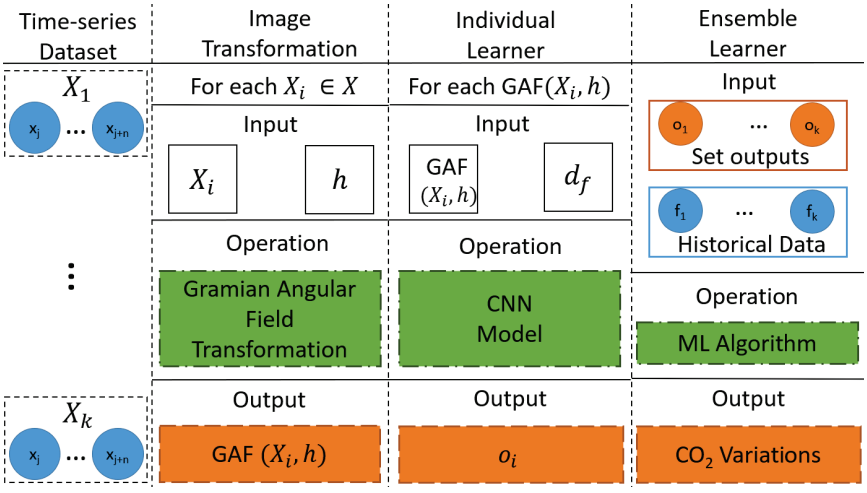


Figure 2. Methodology Steps.

4.1. Time-Series to Image Transformation

The input represents time-series data of environmental features defined by $X = \{X_1, X_2, ..., X_k\}$, such that each X_i where $i \in [1, k]$ describes the time-series data of a single environmental feature. The discussion about the extracted environmental features is provided in the experimental setup. Accordingly, $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$, such that n is the number of observations. To simplify the subsequent equations, $x_j \in X_i$ would represent values of a feature i at timestamp j.

The exploratory data analysis showed that a lag time of up to 20 min for each environmental feature can be used to predict the CO_2 concentration changes for a conference room that fits 12 people. As such, there is a need to reflect the existing time correlation in the feature engineering step. Python packages can be used for extracting time-based features such as the Time Series Feature Extraction Library [37] and Time Series FeatuRe Extraction on the basis of the Scalable Hypothesis test [38] (tsfresh). The extraction of hand-crafted features resulting from these two packages requires incorporating domain-specific insights, which are not available for the case under study. Additionally, the large feature space necessitates integrating feature selection or transformation steps. To address the limitations of the hand-crafted extraction of features, DNNs create their own features by the composition of multiple non-linear functions, each representing combinations of the input dataset features. One concern with DNNs, in their crude form, is their inability to model the time aspect, despite the manual process of including lagged versions of the features. Therefore, a feature engineering stage is incorporated to reflect the underlying time correlation between lagged versions of the available variables.

Wang and Oates [39] have devised the Gramian Angular Field (GAF) method for the transformation of time series data into a symmetrical 2D matrix. As opposed to other time series-to-image transformations, the resulting GAF image exposes the temporal dependencies between data points in a time series. Towards obtaining GAF images, two steps are followed. In the first step of the transformation, the time series data are normalized between $[0, 1]$. The normalization process is conducted in the following equation, whereby $\tilde{x_j}$ is the normalized value:

$$\tilde{x_j} = \frac{x_j - min(X_i)}{max(X_i) - min(X_i)} \quad (2)$$

In the second step, the normalized data is transformed into the polar coordinate system that is calculated as follows:

$$\begin{aligned} \phi_j &= \arccos(\tilde{x}_j),\ 0 \leq \tilde{x}_j \leq 1 \\ r &= \frac{t_j}{N},\ t_j \in \mathbb{N} \end{aligned} \tag{3}$$

After obtaining the transformation to polar coordinates for each data point, the next step is to generate a matrix that reflects the data dependencies between the new form of observations. The GAF technique exploits the angular representation by calculating the pairwise trigonometric sum between the points to reflect the temporal correlations. Considering a time frame of k seconds, such that $\theta_{j,j+1} = \cos(\phi_j + \phi_{j+1})$, the $k \times k$ GAF matrix is defined as follows:

$$\begin{bmatrix} \theta_{0,0} & \theta_{0,1} & \dots & \theta_{0,k-1} \\ \theta_{1,0} & \theta_{1,1} & \dots & \theta_{1,k-1} \\ \vdots & \ddots & \dots & \dots \\ \theta_{k-1,0} & \theta_{k-1,1} & \dots & \theta_{k-1,k-1} \end{bmatrix} \tag{4}$$

The GAF matrix fits the requirements of the use case under study. First, it exposes the temporal dependencies present in the original time series as the time increases progressively from top-left to bottom-right. Second, the polar coordinates expose the relative correlations between data points with the help of superposition. Lastly, the time-series information is retained in the GAF matrix, which is beneficial when there is a need to integrate original data in any envisioned process. These original values can be extracted from the diagonal values. Each value in the GAF matrix is in the range of $[-1, 1]$, which can be easily transformed into an image.

4.2. Individual Learners

The GAF images resulting from the time-series transformation process reflect the correlation by colour intensities. In their turn, these intensities translate the time correlation aspect of environmental sensory data through spatial resolution. CNNs, one of the variations of DNNs, are a strong candidate to find patterns in the produced grid-like structures and the output variable of concern. This selection stems from a CNN's capability to infer spatial dependencies in a gird-structured input via its filters that extract specific features. Moreover, a CNN's architecture is characterized by its sparse connections and weight sharing, which enables any developed model to be swiftly retrained and tested [40] compared to other resource-intensive and data-intensive DNN models such as LSTM. These properties are desirable for CNN models' deployment in a resource-constrained and dynamically changing environment.

Predicting the variations in CO_2 values is split into two steps. The first step is achieved using the individual learners considering the future direction of CO_2 variations as its output variable. The direction is divided into three groups, decreasing equivalent to 0, increasing equivalent to 2, and constant equivalent to 1. The new definition of the output variables transforms the task at hand from regression to a classification task, which allows exploiting the wealth of literature for CNN-related classification tasks. The second step is achieved using the ensemble learner explained in the following sections.

A large body of research was conducted in the field of image recognition that experimented with different CNN architectures to obtain good accuracy results for their respective tasks. Therefore, a Transfer Learning (TL) approach is adopted to extract spatial features in the GAF representations to exploit these accumulated contributions. TL is a methodology to transfer the knowledge gained from an extensive source dataset to improve the learning process in the target dataset [41]. In DL, this methodology is implemented by transferring the weights learned from the source model to the target model. The differences between the source task, trained on the ImageNet dataset [42], and target tasks,

trained on GAF representations, necessitate re-tuning the model's weights using the target dataset or augmenting the initial model with extra layers so that the resulting model can accrue some of the characteristics of the target dataset. The employment of a TL approach has several benefits. The available pre-trained CNN models have proven their merit in a similar domain, saving the time needed to train and tune CNN models from scratch. The addition of more layers and consequently the exponential growth in the number of parameters are required to extract more abstract features from the input images. This step is integral for pattern recognition, which pre-conditions the existence of a large amount of data. The limitations of the deployment environment in terms of the constrained resources and gathered data drive the adoption of the TL approach.

The newly developed CNN model is built on each of the environmental features, such that the predictor of each variable is the feature's GAF image for the history time window and the response variable is the direction of the variation of CO_2 concentration levels for the future time window. The past and future time windows are denoted by h and f. Assuming that the CNN model is denoted by a function Φ_i built on the data from the environmental feature X_i, the predictor input GAF image by $GAF_{(X_i,h)}$, and the prediction is d_f, the created CNN model on the feature X_i is as follows:

$$\Phi_i(GAF_{(X_i,h)}, d_f) \tag{5}$$

The following example is given to clarify the notation, the notions of history and future time windows, and the expression in general. If the current time is $t = 10$ with a one-minute granularity, and the history and future time windows are 5 min for a feature X_1, the GAF image represents X_1 values from $t = 5$ to $t = 10$. The future time frame encompasses the values at $t = 11$ and $t = 16$. Assuming that CO_2 at $t = 11$ is equal to 545 and CO_2 and at $t = 16$ is equal to 547, which means that $d_f = 1$.

4.3. Ensemble Learner

The dataset is split into training, validation, and testing sets. Models of each individual feature are developed using the training set. Since each individual learner outputs a probability prediction of each class, the probability of predicting decreasing change on the validation set is retained. This method is adopted to avoid overfitting the individual learners on the validation set. The individual learner's classification task predicts the direction of future CO_2 variations, instead of the variation itself. Therefore, the learner's output probability values are combined with other features, describing the historical e2s values of each environmental feature, as inputs to an ensemble model to predict CO_2 variations. This method is referred to as a Stacked Generalization ensemble. These features are obtained from the validation set. The set of outputs of all the individual features is denoted by O_t such that $o_{it} \in O$ represents the output of environmental feature i at the current time t. The set of historical features is defined as F_t such that f_{it} represents the historical environmental e2s values. The ensemble learner is denoted by Ψ. The predictors of the ensemble learner are represented by $L = O \cup F$ and the output is p_f, such that p represents the variation of CO_2 concentrations in $f = t + w$, whereby w represents the future time window. The notation of the ensemble learner is as follows:

$$\Psi(L, p_f) \tag{6}$$

The exploratory data analysis has shown a varying correlation between CO_2 values and the other environmental sensors. Because correlation values do not fully reflect the relationship between different environmental features, all the features were included in the Individual Learner step. This step yielded learners that are skillful in modelling the relationship between each environmental feature and the CO_2 variation direction. However, since all of the features contribute to some extent to this variation, there is a need to reflect the relative importance of each model. Assigning weights based on the classification results is a method to convey the contribution of each model to the regression

task. The environmental features' predictions can be replaced by averaging the outcomes of each feature. However, this method does not explore the potential interactions existing between the input variables and undermines the classification of weak individual learners. Therefore, in this paper, different ensemble methods, representing supervised ML models, are explored and evaluated.

5. Experimental Setup

This section discusses the experimental procedure, which explains the different steps undertaken to solve the issue under study. After that, details of the implementation procedure are outlined and the evaluation criteria are investigated.

5.1. Experimental Parameters

The analysis of HMCOVP commences by applying it to a large conference room, referred to as room00 and then later the models are evaluated on a smaller conference room, referred to as room01. Room00 can fit 12 people, with an area comprising 28.0 m^2 and a volume of 89.6 m^3 while room01 is of area 14.2 m^2 and a volume of 38.3 m^3 [25]. This process is followed to evaluate the transferability of the developed models to a room of different capacities. The hierarchical structure of HMCOVP promotes its transferability, which addresses a common concern in the IAQ-related literature. The following paragraphs explain the parameters involved in the experimental procedure summarized in Figure 3.

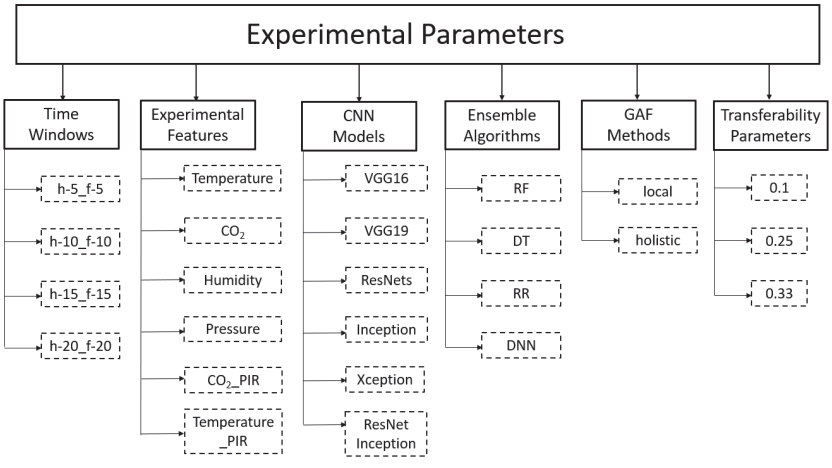


Figure 3. Summary of Experimental Parameters.

History and Future Time Windows: The EDA highlighted the relationship between lagged environmental features and the CO_2 concentrations to infer a reasonable prediction time window. This EDA showed that a prediction horizon of 20 min is viable for the available dataset. The analysis includes the results of applying HMCOVP with the same time window for history and future time horizons. These parameters are subsets of the ones evaluated in FECOP. The remainder of the possible combinations will be considered in future work, given that the current formulation is not concerned with the memory constraints of the deployment environment. The history and the future time horizon will be referred to as "h-{w}_f-{w}", whereby w represents the time window. Time windows of 5, 10, 15, and 20 min were considered. The combinations of the history and future time windows experimented with are h-5_f-5, h-10_f-10, h-15_f-15, and h-20_f-20.

The time series to image transformation step yielded several datasets that vary depending on the historical and future time windows. A 1-minute sliding window is implemented to produce larger datasets. For example, for h-5_f-5, a single row is formed by predictors representing GAF images from $t = 0$ to $t = 5$ and the CO_2 variations from $t = 5$ to $t = 10$.

The next row would represent the GAF images from $t = 1$ to $t = 6$ and the CO_2 variations from $t = 6$ to $t = 11$. The resulting process yields datasets of different sizes that range between 328,647 to 334,677 data points for training data and 74,085 to 74,730 data points for testing data for different history and future time horizons. The training and testing data points for different history and future time windows are summarized in Table 1.

Table 1. Dataset Sizes.

History (Minutes)	Future (Minutes)	Training Data Testing Data
h-5	f-5	334,677 74,730
h-10	f-10	332,667 74,514
h-15	f-15	330,657 74,300
h-20	f-20	328,647 74,085

Experimental Features: The original dataset consists of five environmental features: pressure, temperature, humidity, CO_2 concentration, and activity levels. The activity level or PIR is a categorical variable formed of 0.5 increments from 0 to 12, amounting to 25 variables. The PIR feature space is first reduced to four categories, each representing eight contiguous activity levels, except 0, which is a category on its own. The one-hot encoding schema would create many 0–1 variables, which cannot be transformed into an image representation. Therefore, the PIR feature is transformed into two features, denoted by temperature_PIR and CO_2_PIR. These two features are chosen because they exhibit the highest correlation with the PIR feature. The quantities are obtained by calculating the ratio of CO_2 and temperature to the PIR feature. Six environmental features are obtained following this feature engineering step that includes CO_2, temperature, pressure, humidity, CO_2_PIR, and temperature_PIR.

GAF Representation: Normalization is a fundamental step in the time series to image transformation. Two main methods can branch out from this step, referred to as "local" and "holistic" approaches. Each of these methods targets a specific angle of the studied phenomenon. On the one hand, the "local" method calculates the minimum and the maximum in a defined w. This representation accentuates the small differences in such windows, reflected by the colour intensities of the GAF representation. On the other hand, the "holistic" approach calculates the minimum and the maximum over the whole datasets of a single environmental feature. The resulting GAF images are in concert with the dynamics in the whole dataset, which means that the images' colour intensities mirror the differences in values in a w compared to value differences over the whole dataset. Both of these approaches are investigated in the experimental procedure. The existence of outliers is fundamental in the GAF representation, affecting the accuracy of CO_2 variation predictions.

The EDA provided interesting insights into each environmental feature. Features such as CO_2, CO_2_PIR, and humidity displayed highly skewed distributions with large differences between the minimum and maximum values, to the detriment of the "holistic" approach. If these features are normalized based on these extremes, the majority of the data will fall within a small margin, and the resulting differences in GAF images will be minuscule. As a result, the CNNs will fail to infer any meaningful relationship from the GAF representation. A log transformation is applied to these features to mitigate the effect of outliers and to shrink the values' ranges. Other features, such as pressure and temperature features displayed little variations. The exponential function is applied to each of these features to amplify these small differences. These concerns are not shared with the "local" approach, since its GAF representation depends on the minimum and maximum

calculated in a time window. Therefore, the differences in feature values will be mirrored in the GAF representation despite their value ranges resulting from any transformation.

Individual CNN Models: The research community provides ample image recognition models for individual learners that can be applied to the generated datasets. These pre-trained CNN models include VGG16 and VGG19 [43], ResNets [44], ResNet_inception [45], Inception [46], and Xception [47] with different respective architectures. The experimental procedure evaluated all these CNN models. Keen readers can find the details of their respective architectures in the papers listed with every CNN model. Each of their architectures includes many convolutional and Fully Connected Layers (FCL). In the experimental procedure, the last layer that outputs the classification result is removed, given that a different dataset with different outputs is used in this study. Therefore, the remainder of the layers is augmented with a global average pooling layer and Fully Connected Layers (FCL). The obtained CNN models include either no FCLs, one FCL, or two FCLs. The number of neurons of these FCLs ranges from 64 to 4096 neurons.

Ensemble Learner Algorithms: Different algorithms were evaluated for the ensemble learning step. In particular, the same algorithms used in FECOP were evaluated following HMCOVP. These ML algorithms include RF, DT, RR, and DNNs. This step is followed to emphasize the distinction between the two methodologies in terms of the validity of feature engineering techniques.

Transferability Parameters: The transferability of the developed models is evaluated by applying them to a small room, denoted by room01, that fits two people. When the results were unsatisfactory, part of the training data of the target room is used to tune the parameters of the source room's model. The fraction of the used training data is $\{0.1, 0.25, 0.33\}$. Moreover, this experimental analysis included re-tuning or re-training the ensemble learners. Future works will include a more profound evaluation of the HCOMVP transferability.

5.2. Experimental Procedure

In the training phase of the methodology, the dataset is split into training, validation, and testing datasets. The training dataset is used to train the individual learners. The validation dataset is used to obtain the outputs of each of the individual learners that are used with the historical e2s values of environmental features in the validation set as inputs to train the ensemble learners. Such a split is followed to avoid overfitting the ensemble learner if fed with the outputs of the individual learners applied to the training set. After obtaining the trained individual learners and the ensemble learner, these learners are evaluated on the testing set.

5.3. Evaluation Metrics

The HMCOVP is evaluated against FECOP [30] and 1D CNNs, a popular method for time-series prediction. Since the problem of predicting CO_2 variation prediction is a regression task, multiple evaluation metrics such as Mean Square Error (MSE) or Mean Absolute Error (MAE) can be used. Choosing one of these metrics depends on the experiments' main goals.

The MSE metric emphasises the large errors while minimizing the effect of the smaller ones. On the other hand, the MAE equalises the effect of all individual errors. No anomaly detection method was applied to sanitize the gathered data, which suggests the existence of data corrupted with noise. This noise originates either from the environment itself or from the inaccuracies of the sensing technologies. Such circumstances call for a robust evaluation metric that can offset the noise's effect. On a different note, the small prediction errors that are continually produced by models can result in long-term consequences to the HVAC systems. The predictions of CO_2 variations affect the ventilation systems that need to maintain the indoor CO_2 concentrations within acceptable levels and the estimation of current occupancy changes dictating the HVAC system's operation. Consequently, small deviations, while insignificant in the short term, can accumulate unnecessary activation

of HVAC systems, increasing the building's Carbon footprint and energy expenditure. Large deviations of predictions originate either from models' under-fitting data, which can be identified if it is a shared trend among different input values or from the existence of anomalous data that the model fails to fit. The anomalous data can be manifested by wrong sensor readings, corrupting the data with noise, or the occurrence of rare events that have little implications on long-term energy expenditure. As such, the robustness requirement and the equal importance of small and large deviations favour the use of MAE as an evaluation metric.

The MAE evaluation metric is used to find the best set of hyper-parameters on the validation set and to compare the performance of the HMCOVP, FECOP, and 1D CNNs on the testing set. The methods are evaluated on a set of CO_2 variations exceeding pre-defined thresholds, including $\{5, 10, 20, 40, 50, 75, 100\}$. These thresholds unveil different methods' capability to predict big variations in CO_2 concentration, which can potentially activate the HVAC systems and determine the instantaneous energy expenditure. The average MAE combining all these thresholds is employed to highlight the performance differences between the proposed methodology and the other approaches, providing a comprehensive overview of the proposed methods' performance under different circumstances. The training and instance-wise testing times are incorporated into the evaluation to gauge the methods' ability to learn in resource-constrained environments. Different CO_2 variation thresholds and execution times are important factors of the deployment of proper solutions for building operators. The accurate predictions of CO_2 changes are envisioned to facilitate the ventilation system's decision-making processes. Therefore, the reduction in the superfluous activation of these systems translates into limiting energy expenditure. In the grand scheme of things, this methodology contributes to the mitigation of Carbon emissions, which can be quantified depending on the ore used for electricity production.

5.4. Implementation

The different algorithms used throughout this paper were built using sklearn [48] and Keras [49] python libraries. The developed models were evaluated on Windows 10 PC with a 3.00 GHz 24-Core AMD Threadripper processor, 128 GB of RAM, and 8 GB Nvidia GeForce RTX 3060 Ti GPU. The code is made available on the GitHub repository (https://github.com/Western-OC2-Lab/hierarchical-CO2, accessed on 14 May 2023).

6. Results and Discussion

This section analyses and discusses the results obtained from applying the HMCOVP. Firstly, it compares the performance of the HMCOVP using different parameters over different combinations of history and future time windows. After that, the best parameters are evaluated against competing methodologies. Lastly, the transferability of the models is investigated when applied to a smaller conference room.

6.1. Parameter Selection

Under different history and future time windows, this subsection explains the effect of parameters, including the GAF representation method, CNN models and their hyper-parameters, and ensemble learners. To that end, a subset of 10,000 data points for each history and future window combination (h&f) are used for parameter selection. A trial-and-error approach is conducted to find a subset that can balance training and evaluation times and underfitting avoidance. All the combinations of features were evaluated using a Grid Search method. More profound and less time-demanding approaches [50] will be investigated in future work. The sample of 10,000 data points is split into training, validation, and testing sets following the explained experimental procedure. Since many parameters are involved in the parameter selection procedure, this section explains a subset of these experiments. Keen readers and practitioners can refer to the code available through the GitHub repository for a comprehensive overview of the obtained results and the effect of all parameters. As such, the analysis in this section is restricted to the effect of different

ensemble learners, the h&f combinations, and CO_2 variation thresholds on the performance of the HMCOVP.

Table 2 summarises the findings of the parameter selection grouped by h&f. The best-performing combination of parameters and hyper-parameters with respect to each h&f are highlighted in bold. To highlight the significance of these differences in the MAE, it is important to map the obtained MAE values to their physical representations. The CO_2 variation predictions help the ventilation systems in their decision-making process. Therefore, as previously mentioned, small MAEs can accumulate to falsely trigger the ventilation system. Moreover, less accurate CO_2 variation predictions imply inaccurate prediction of the current effect of occupants and ventilation systems on CO_2 concentration variations. Both of these factors contribute to reflecting an imprecise image of the environment in the HVAC decision-making systems. Accordingly, HVAC systems can be activated either early or late, contributing to potential violations of indoor environmental requirements and an increase in energy expenditure. As such, the MAE reflects a fundamental aspect of the HVAC system operation.

Table 2. Parameter Selection on the Training Dataset.

History and Future Time Window (in Minutes)	Ensemble Algorithm	CNN Model	CNN FCL	Method	MAE
h-5_f-5	RR	VGG_16	[64]	holistic	1.61
	DT	Resnet_152	[256]	local	0.65
	RF	VGG_16	[512]	holistic	**0.4**
	NN	Resenet_152	[512, 256]	holistic	1.3
h-10_f-10	RR	VGG_19	[4096]	local	3.25
	DT	VGG_19	[512]	local	0.84
	RF	Resnet_152	[128, 64]	local	**0.765**
	NN	Resnet_101	[256, 128]	local	2.63
h-15_f-15	RR	VGG_16	None	local	5.54
	DT	Resnet_152	[256, 128]	local	**0.98**
	RF	Xception	None	local	1.22
	NN	Resnet_101	[256]	local	4.48
h-20_f-20	RR	Resnet_50	[128]	holistic	6.07
	DT	Resnet_101	[128, 64]	local	1.18
	RF	VGG_19	[4096]	holistic	**0.84**
	NN	Resnet_50	[128, 64]	local	4.91

The variants of decision trees are the best-performing algorithms, represented by DT and RF. This shows that the non-linearity defines the relationship between the individual learners' predictions and the historical e2s environmental features manifested through the superior MAE of non-linear algorithms (NN, RF, and DT) compared to the RR algorithm. The expansion of the prediction window contributed to a systematic increase in the MAE for all algorithms. This consensus is broken by the RF algorithm such that its MAE value decreases when the prediction horizon increases from 15 to 20 min. However, with more runs to execute, this exception will be reversed to conform to the trend. This increase in MAE is expected with the expansion of the future time window, given the increased probability of uncertainties affecting the environment, including the CO_2 concentrations. In terms of the GAF representation, the best-performing models for each h&f combination do not highlight a preferable method. However, most ensemble learners display a better performance with the local method. Therefore, the CNN models extract representative information of CO_2

variation direction with better accuracy when fed with GAF representations that accentuate local differences within a time window.

The summarized results in Table 2 have shown some parameter combinations that clearly outperform others. However, a clear limitation of the MAE parameter in this application is that it is skewed toward the values that constitute the majority of the response variable. In order to clarify this caveat, Figure 4 shows the percentage distributions of different absolute values of thresholds defined in the evaluation criteria. These results suggest that more than 60% of the variations are under 5 ppm. Therefore, a model with a low MAE is not fully representative of its performance on drastic changes in data. These radical changes are of higher importance for HVAC systems and building operators, but they are less common in the studied dataset. As a result, the crude MAE is replaced by a metric that averages the MAEs over each of the defined thresholds. This new metric is referred to as **Thresh_MAE**.

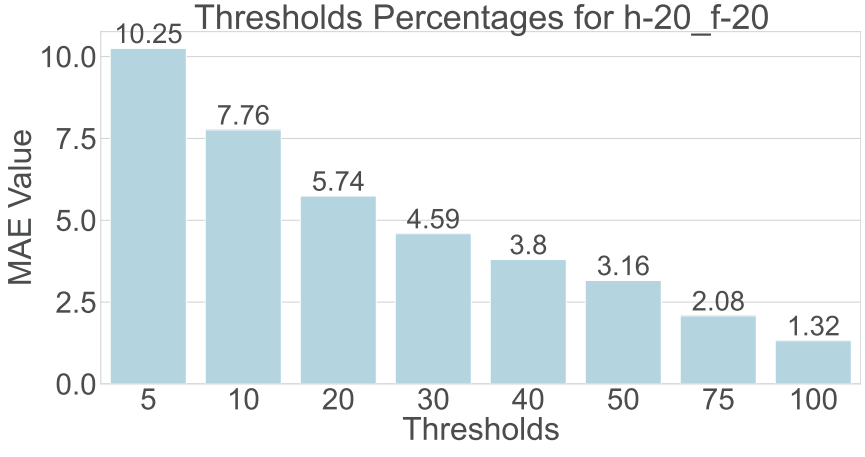
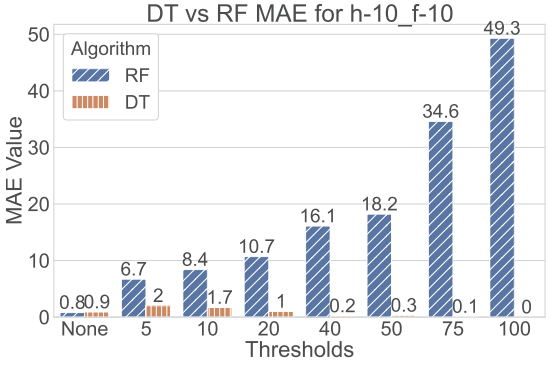
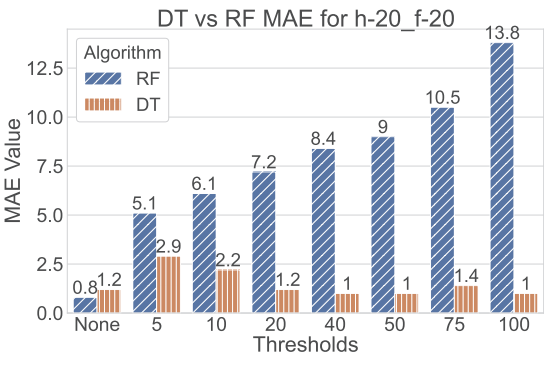


Figure 4. Thresholds' Data Percentages.

Figure 5 shows the differences in MAE for each of the defined thresholds for the two best-performing models in Table 2 that use DT or RF for ensemble learning. Figure 5a shows that RF combined with the other parameters included in Table 2 outperforms its DT counterpart when no thresholds are imposed. However, the performances start to diverge with increasing thresholds until a huge gap in performance is prominent. This trend shows that the RF model performs well on small variations in CO_2 while DT outperforms RF when more drastic changes are involved. A similar observation can be applied to Figure 5b, whereby DT outperforms RF when thresholds are considered. However, a notable separation exists between the results of the RF for higher thresholds with the expansion of the prediction time horizon. This discrepancy is attributed to the abundance of more extreme variations with the expansion of future time windows. This fact enables the models to better map the relationships between these extreme values and the input variables, manifested by the lower MAE for higher thresholds.



(**a**) DT vs. RF for h-10_f-10



(**b**) DT vs. RF for h-20_f-20

Figure 5. MAE for Different Thresholds.

Following the newly defined metric, **Thresh_MAE**, the best-performing parameters are shown in Table 3. As opposed to pure MAE, DT clearly outperformed all other ensemble learners and showed superior performance in predicting drastic variations. This observation is demonstrated by the decrease in Thresh_MAE value compared to MAE value, when DT is involved, for all h&f combinations. All the CNN models included FCLs, which emphasize the existing distinction between the source and target datasets in terms of the interactions between the extracted CNN features. Lastly, the local method of GAF representation dominates the better-performing combinations. This result shows that CNNs produce more informative features for CO_2 direction prediction when the effect of small differences in values are magnified, resulting in more accurate classifications. The parameters included in Table 3 qualify for the next stage in the evaluation pipeline by training them on the whole training data and evaluating their performance on the testing test. The small **Thresh_MAEs** obtained in Table 3 unveil that the models overwhelmingly captured the factors affecting the environment. However, these small differences show that some environmental conditions are not considered, potentially playing a role in deciding the future variations in CO_2 concentrations. These conditions may include the uncertainty of the environment that can be challenging to incorporate. Another factor can be connected to the interpolation strategy that is not reflective of the environmental dynamics. Lastly, the effect of the ventilation systems' activation is excluded from the gathered data. This condition can affect the dynamics of the CO_2 variation in connection with other environmental features.

Table 3. Best-Performing Combinations.

h&f	Ensemble	CNN Model	CNN FCL	Method	Thresh_MAE
h-5_f-5	DT	Xception	[512]	local	0.11
h-10_f-10	DT	Resnet_50	[128, 64]	local	0.6
h-15_f-15	DT	Resnet_152	[256, 128]	local	1.0
h-20_f-20	DT	Resnet_101	[128, 64]	local	1.49

6.2. HMCOVP vs. FECOP vs. 1D CNN

Table 4 summarises the performance of the three methods, our proposed methodology HMCOVP, FECOP, and 1D CNN, which are evaluated using **Thresh_MAE** and their training times. The best-performing combination configurations of hyper-parameters for the FECOP and 1D CNN are applied to the testing set. Similarly, the HMCOVP was implemented using the configurations outlined in Table 3. The HMCOVP significantly outperforms the other two methods in the prediction of the future CO_2 variations in all

the h&f combinations. The results prove the GAF images are better representatives of the time correlation aspects than the local features extracted by the 1D CNN to reflect these correlations. Additionally, the combination of image representations and numerical features present in the HMCOVP outperforms the rigorous feature engineering process of FECOP. While the HMCOVP incorporates some of the FECOP features, the distinctive quality of the features obtained from CNN models contributed to its superior performance. In some way, the HMCOVP combines the numeric features extracted in the FECOP and the time correlation found in the 1D CNN approach. These results prove the significance of hierarchical modelling.

Table 4. HMCOVP vs. FECOP vs. 1D-CNN.

Parameter	Thresh_MAE			Training Time (min)		
Methodologies	**HMCOVP**	**FECOP**	**1D-CNN**	**HMCOVP**	**FECOP**	**1D-CNN**
h-5_f-5	**10.14**	40.89	2331.11	229.69	2.1	36.5
h-10_f-10	**14.48**	52.52	8969.98	360.34	2.275	22.62
h-15_f-15	**19.37**	66.83	9201.81	716.54	2.83	19.16
h-20_f-20	**2**7.74	77.21	10,128.83	381.78	3.62	17.84

The training times highlight salient differences between the methods. The 1D CNN method training time decreases with the expansion of the time window. This trend is expected given that fewer data are available for training when this happens. On the other hand, this trend is reversed for the FECOP. This phenomenon can be attributed to the feature engineering step, which extracts lagged versions of each environmental feature. As such, more features are extracted with the expansion of the time window, contributing to an increase in training time. Lastly, the HMCOVP does not exhibit any trend with the changes in the time window. Different configurations with varying model complexities contribute to the non-uniformity of training times. Compared to the FECOP and 1D CNN, the HMCOVP takes significantly more time to train given that it requires training six DL models and one ensemble model. The training of individual models can be executed in parallel, which significantly reduces the total training time. Therefore, a tradeoff exists between training times and the accuracy of the developed models.

6.3. Transferability Assessment

After proving its superior performance in predicting CO_2 variations in room00, the next stage assesses the transferability of the developed models by applying them to a smaller room, referred to as room01, which only fits two people. The FECOP models' results are obtained by training the models on the training set of room01. Both the HMCOVP and FECOP approaches are evaluated on the testing set of room01 using the **Thresh_MAE** metric. The evaluation criteria include the testing time per instance to infer the computational footprint of these methods. As for the HMCOVP, the models developed using room00's training data are applied to the testing data of room01 without fine-tuning any of its individual learners or ensemble learner. As such, the transferability of the developed models is assessed in different spatial settings.

Table 5 summarises the results of the outlined process. In terms of predictive performance, the untuned HMCOVP outperforms its counterpart in every h&f combination. The HMCOVP models performed best with the h-15_f-15 and h-20_f-20 compared to other combinations. The h-15_f-15 combination performed best in terms of its **Thresh_MAE**, breaking the established trend of increasing **Thresh_MAE** with the increase in the prediction horizon. The h-20_f-20 experienced the least performance percentage gap in **Thresh_MAE** from room00 to room01. This observation means that the larger room's environment dynamics acquired through hierarchical modelling closely resemble those of the smaller room in the bespoke combinations. As for smaller prediction horizons, it is expected that smaller rooms

would exhibit different variations. In fact, changes in the small room's environment, such as the existence of occupants, can momentarily affect the CO_2 concentrations as opposed to the more spacious rooms. Lastly, the differences in the capacities of both rooms result in changes in the ranges of activity level-related features; thus, affecting the HMCOVP models' accuracy. Consistent with the previous analysis, the FECOP approach is a less time and resource-intensive approach, manifested by its lower per-instance testing time. A 6–7$\times$ speedup is obtained by the FECOP approach; however, at the expense of its underwhelming prediction performance. The speedup benefit is blurred when the individual learners' are parallelised.

Table 5. Room01 Result Comparisons.

Parameters	**Thresh_MAE**		**Time/Instance (ms)**	
Methodologies	**HMCOVP**	**FECOP**	**HMCOVP**	**FECOP**
h-5_f-5	41.11	49.35	6.24	0.67
h-10_f-10	43.14	52.4	6.48	1.14
h-15_f-15	39.90	55.9	14.64	2.16
h-20_f-20	49.93	54.2	10.44	1.44

The differences in feature scales between the two rooms require incorporating some of these missing characteristics into the developed HMCOVP model. Thanks to the hierarchical structure of the HMCOVP, the induction of this novel information can be realized on the level of individual and ensemble learners. As a result, the next step can include one of these scenarios. The first scenario re-tunes the individual CNN models by freezing some layers and training others. The second scenario replaces the old ensemble trained on room00 training data with a newer one trained using room01's training data. While both scenarios are viable, choosing one of them depends on the obtained results and the nature of the data. The results have shown that the greatest performance divergence occurs in the ensemble learning phase as the individual learners almost produce the same accuracy results in both rooms. This gap can be surmised by the differences in e2s values and their effect on the future CO_2 variations. Therefore, the ensemble learners are retrained using a subset of room01's training data and evaluated using its testing data.

Figure 6 shows the effect of retraining the ensemble learner with different training sizes. The HMCOVP's retrained ensemble model regressed in performance when applied to the h-5_f-5 combinations, regardless of the integrated training set's size. This observation essentially suggests that the tuning process for this combination should incorporate individual models. On the other hand, a significant performance improvement is noted for all the other combinations. The greatest improvement is observed for the h-20_f-20 combinations, whereby its performance is the closest to the model applied to room00. For all the improved combinations, retraining the ensemble model with a quarter of the training set yields the best results. The results of these combinations show that incorporating the historical numerical values with the individual learners' outputs contributes to performance enhancement. However, this alteration does not fully capture the dynamics of the target set, representing the small conference room. To mitigate the performance gap in CO_2 variation predictions, the individual learners should be tuned the same as the ensemble learner. This aspect will be addressed in future work.

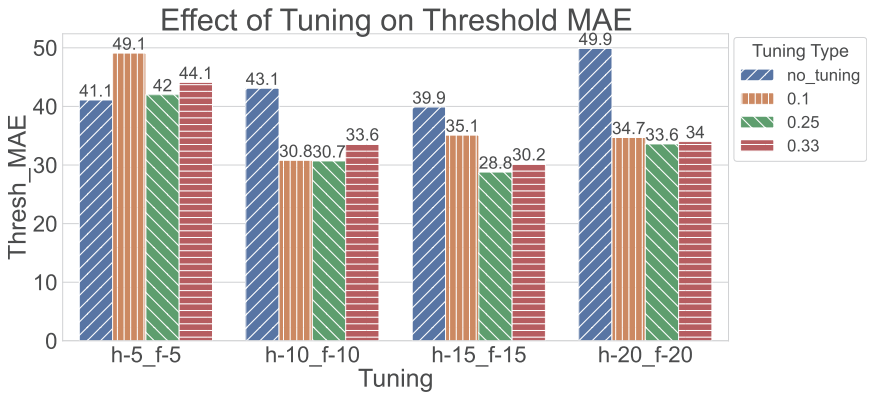


Figure 6. Effect of Tuning.

7. Conclusions

Buildings are one of the main contributors to energy consumption, whereby HVAC systems represent a controllable energy consumer. Specifically, ventilation systems are activated based on the requirements and existence of occupants. Therefore, the prediction of occupants and the utility of current ventilation control can be achieved through accurate predictions of proxy indicators, CO_2 concentration changes, in particular, are key to the optimization of HVAC systems. This work proposes a hierarchical modelling approach, termed the HMCOVP, with the goal of accurately predicting CO_2 concentration changes. The HMCOVP incorporates the time aspect by creating image-based lagged environment features and a hierarchical structure that enables the transferability of the developed models. These two features are missing in the state-of-the-art approaches. The HMCOVP was evaluated using a host of history and future time windows in a large office room and outperformed the state-of-the-art approaches by 400% using the mean absolute error metric. The transferability of the HMCOVP is investigated by applying and re-tuning the developed models in a smaller room and promising results are obtained. Overall, the HMCOVP successfully predicted future CO_2 concentration changes in different spatial settings, facilitating the decision-making process of ventilation systems. Future work will experiment with ways to shorten the training time of the HMCOVP using feature selection techniques and enhance the transferability performance.

Author Contributions: Conceptualization, I.S. and A.S.; methodology, I.S.; software, I.S.; validation, I.S. and A.S.; formal analysis, I.S.; investigation, I.S.; resources, I.S.; data curation, I.S.; writing—original draft preparation, I.S.; writing—review and editing, I.S. and A.S.; visualization, I.S.; supervision, A.S.; project administration, A.S.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented and used in this study are openly available in Zenodo at (10.5281/zenodo.3774723).

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

References

1. The Rockefeller Foundation and Deutsche Bank Group. Building Retrofit Paper - Final 3-1-12 vEDIT. Rockefellerfoundation. Available online: https://www.rockefellerfoundation.org/wp-content/uploads/United-States-Building-Energy-Efficiency-Retrofits.pdf (accessed on 16 May 2020).
2. Berardi, U. Building energy consumption in US, EU, and BRIC countries. *Procedia Eng.* **2015**, *118*, 128–136. [CrossRef]

3. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [CrossRef]
4. Satish, U.; Mendell, M.J.; Shekhar, K.; Hotchi, T.; Sullivan, D.; Streufert, S.; Fisk, W.J. Is CO_2 an indoor pollutant? Direct effects of low-to-moderate CO_2 concentrations on human decision-making performance. *Environ. Health Perspect.* **2012**, *120*, 1671–1677. [CrossRef] [PubMed]
5. Du, B.; Tandoc, M.C.; Mack, M.L.; Siegel, J.A. Indoor CO_2 concentrations and cognitive function: A critical review. *Indoor Air* **2020**, *30*, 1067–1082. [CrossRef]
6. Lewis, D. Why indoor spaces are still prime COVID hotspots. *Nature* **2021**, *592*, 22–25. [CrossRef] [PubMed]
7. Berry, G.; Parsons, A.; Morgan, M.; Rickert, J.; Cho, H. A review of methods to reduce the probability of the airborne spread of COVID-19 in ventilation systems and enclosed spaces. *Environ. Res.* **2022**, *203*, 111765. [CrossRef]
8. World Health Organization. *A Conceptual Framework for Action on the Social Determinants of Health*; World Health Organization: Geneva, Switzerland, 2010.
9. Xin, J.; Duan, Q.; Jin, K.; Sun, J. A reduced-scale experimental study of dispersion characteristics of hydrogen leakage in an underground parking garage. *Int. J. Hydrogen Energy* **2023**, *48*, 16936–16948. [CrossRef]
10. Mallach, G.; St-Jean, M.; MacNeill, M.; Aubin, D.; Wallace, L.; Shin, T.; Van Ryswyk, K.; Kulka, R.; You, H.; Fugler, D.; et al. Exhaust ventilation in attached garages improves residential indoor air quality. *Indoor Air* **2017**, *27*, 487–499. [CrossRef]
11. Cao, S.; Chen, X.; Zhang, L.; Xing, X.; Wen, D.; Wang, B.; Qin, N.; Wei, F.; Duan, X. Quantificational exposure, sources, and health risks posed by heavy metals in indoor and outdoor household dust in a typical smelting area in China. *Indoor Air* **2020**, *30*, 872–884. [CrossRef]
12. Liao, W.; Liu, X.; Kang, N.; Song, Y.; Wang, L.; Yuchi, Y.; Huo, W.; Mao, Z.; Hou, J.; Wang, C. Associations of cooking fuel types and daily cooking duration with sleep quality in rural adults: Effect modification of kitchen ventilation. *Sci. Total Environ.* **2023**, *854*, 158827. [CrossRef]
13. Cakyova, K.; Figueiredo, A.; Oliveira, R.; Rebelo, F.; Vicente, R.; Fokaides, P. Simulation of passive ventilation strategies towards indoor CO_2 concentration reduction for passive houses. *J. Build. Eng.* **2021**, *43*, 103108. [CrossRef]
14. Metcalf, G.E.; Weisbach, D. The design of a carbon tax. *Harv. Envtl. L Rev.* **2009**, *33*, 499. [CrossRef]
15. Act, E.P. *Energy Policy Act of 2005*; US Congress: Washington, DC, USA, 2005.
16. Esrafilian-Najafabadi, M.; Haghighat, F. Occupancy-based HVAC control systems in buildings: A state-of-the-art review. *Build. Environ.* **2021**, *197*, 107810. [CrossRef]
17. Kleiminger, W.; Mattern, F.; Santini, S. Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches. *Energy Build.* **2014**, *85*, 493–505. [CrossRef]
18. Zou, H.; Zhou, Y.; Yang, J.; Spanos, C.J. Towards occupant activity driven smart buildings via WiFi-enabled IoT devices and deep learning. *Energy Build.* **2018**, *177*, 12–22. [CrossRef]
19. Scislo, L.; Szczepanik-Scislo, N. Air quality sensor data collection and analytics with iot for an apartment with mechanical ventilation. In Proceedings of the 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 22–25 September 2021; Volume 2, pp. 932–936.
20. Sun, K.; Zhao, Q.; Zou, J. A review of building occupancy measurement systems. *Energy Build.* **2020**, *216*, 109965. [CrossRef]
21. Naylor, S.; Gillott, M.; Lau, T. A review of occupant-centric building control strategies to reduce building energy use. *Renew. Sustain. Energy Rev.* **2018**, *96*, 1–10. [CrossRef]
22. Chen, Z.; Jiang, C.; Xie, L. Building occupancy estimation and detection: A review. *Energy Build.* **2018**, *169*, 260–270. [CrossRef]
23. Li, C.; Cui, C.; Li, M. A proactive 2-stage indoor CO_2-based demand-controlled ventilation method considering control performance and energy efficiency. *Appl. Energy* **2023**, *329*, 120288. [CrossRef]
24. Chen, Z.; Zhao, R.; Zhu, Q.; Masood, M.K.; Soh, Y.C.; Mao, K. Building occupancy estimation with environmental sensors via CDBLSTM. *IEEE Trans. Ind. Electron.* **2017**, *64*, 9549–9559. [CrossRef]
25. Räsänen, P.; Koivusaari, J.; Kallio, J.; Rehu, J.; Ronkainen, J.; Tervonen, J.; Peltola, J. VTT SCOTT IAQ Dataset. 2020. Available online: https://zenodo.org/record/3774723#.ZGOobXbMKUk (accessed on 16 June 2022).
26. Dong, B.; Andrews, B.; Lam, K.P.; Höynck, M.; Zhang, R.; Chiou, Y.S.; Benitez, D. An information technology enabled sustainability test-bed (ITEST) for occupancy detection through an environmental sensing network. *Energy Build.* **2010**, *42*, 1038–1046. [CrossRef]
27. Candanedo, L.M.; Feldheim, V. Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. *Energy Build.* **2016**, *112*, 28–39. [CrossRef]
28. Masood, M.K.; Soh, Y.C.; Jiang, C. Occupancy estimation from environmental parameters using wrapper and hybrid feature selection. *Appl. Soft Comput.* **2017**, *60*, 482–494. [CrossRef]
29. Shaer, I.; Shami, A. Sound Event Classification in an Industrial Environment: Pipe Leakage Detection Use Case. In Proceedings of the International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 30 May-3 June 2022; pp. 1212–1217
30. Kallio, J.; Tervonen, J.; Räsänen, P.; Mäkynen, R.; Koivusaari, J.; Peltola, J. Forecasting office indoor CO_2 concentration using machine learning with a one-year dataset. *Build. Environ.* **2021**, *187*, 107409. [CrossRef]
31. Golestan, S.; Kazemian, S.; Ardakanian, O. Data-driven models for building occupancy estimation. In Proceedings of the Ninth International Conference on Future Energy Systems, Karlsruhe, Germany, 12–15 June 2018; pp. 277–281.

32. Stjelja, D.; Jokisalo, J.; Kosonen, R. Scalable Room Occupancy Prediction with Deep Transfer Learning Using Indoor Climate Sensor. *Energies* **2022**, *15*, 2078. [CrossRef]
33. Wang, Z.; Hong, T.; Piette, M.A.; Pritoni, M. Inferring occupant counts from Wi-Fi data in buildings through machine learning. *Build. Environ.* **2019**, *158*, 281–294. [CrossRef]
34. Laaroussi, Y.; Bahrar, M.; El Mankibi, M.; Draoui, A.; Si-Larbi, A. Occupant presence and behavior: A major issue for building energy performance simulation and assessment. *Sustain. Cities Soc.* **2020**, *63*, 102420. [CrossRef]
35. Yu, L.; Sun, Y.; Xu, Z.; Shen, C.; Yue, D.; Jiang, T.; Guan, X. Multi-agent deep reinforcement learning for HVAC control in commercial buildings. *IEEE Trans. Smart Grid* **2020**, *12*, 407–419. [CrossRef]
36. Wang, Y.; Velswamy, K.; Huang, B. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes* **2017**, *5*, 46. [CrossRef]
37. Barandas, M.; Folgado, D.; Fernandes, L.; Santos, S.; Abreu, M.; Bota, P.; Liu, H.; Schultz, T.; Gamboa, H. TSFEL: Time series feature extraction library. *SoftwareX* **2020**, *11*, 100456. [CrossRef]
38. Christ, M.; Braun, N.; Neuffer, J.; Kempa-Liehr, A.W. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing* **2018**, *307*, 72–77. [CrossRef]
39. Wang, Z.; Oates, T. Imaging time-series to improve classification and imputation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
40. Osaku, D.; Gomes, J.F.; Falcão, A.X. Convolutional neural network simplification with progressive retraining. *Pattern Recognit. Lett.* **2021**, *150*, 235–241. [CrossRef]
41. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1–40.
42. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
43. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
45. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
46. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
47. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
48. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
49. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 16 May 2023).
50. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [CrossRef]

algorithms

MDPI

Article

Learning from Imbalanced Datasets: The Bike-Sharing Inventory Problem Using Sparse Information †

Giovanni Ceccarelli [1], Guido Cantelmo [2], Marialisa Nigro [3,*] and Constantinos Antoniou [4]

[1] Dott SAS, 75002 Paris, France; giovanni@ridedott.com
[2] Department of Technology, Management and Economics, Technical University of Denmark, 2800 Lyngby, Denmark; guica@dtu.dk
[3] Department of Civil, Computer and Aeronautical Engineering, Roma Tre University, 00146 Rome, Italy
[4] Department of Mobility Systems Engineering, Technical University of Munich, 80333 Munich, Germany; c.antoniou@tum.de
* Correspondence: marialisa.nigro@uniroma3.it
† This paper is an extended version of our paper published in the proceedings of the 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2021, 2021 (Heraklion, Crete, 16–17 June 2021).

Abstract: In bike-sharing systems, the inventory level is defined as the daily number of bicycles required to optimally meet the demand. Estimating these values is a major challenge for bike-sharing operators, as biased inventory levels lead to a reduced quality of service at best and a loss of customers and system failure at worst. This paper focuses on using machine learning (ML) classifiers, most notably random forest and gradient tree boosting, for estimating the inventory level from available features including historical data. However, while similar approaches adopted in the context of bike sharing assume the data to be well-balanced, this assumption is not met in the case of the inventory problem. Indeed, as the demand for bike sharing is sparse, datasets become biased toward low demand values, and systematic errors emerge. Thus, we propose to include a new iterative resampling procedure in the classification problem to deal with imbalanced datasets. The proposed model, tested on the real-world data of the Citi Bike operator in New York, allows to (i) provide upper-bound and lower-bound values for the bike-sharing inventory problem, accurately predicting both predominant and rare demand values; (ii) capture the main features that characterize the different demand classes; and (iii) work in a day-to-day framework. Finally, successful bike-sharing systems grow rapidly, opening new stations every year. In addition to changes in the mobility demand, an additional problem is that we cannot use historical information to predict inventory levels for new stations. Therefore, we test the capability of our model to predict inventory levels when historical data is not available, with a specific focus on stations that were not available for training.

Keywords: bike sharing; rebalancing problem; inventory level; machine learning; random forest; imbalanced data

Citation: Ceccarelli, G.; Cantelmo, G.; Nigro, M.; Antoniou, C. Learning from Imbalanced Datasets: The Bike-Sharing Inventory Problem Using Sparse Information. *Algorithms* **2023**, *16*, 351. https://doi.org/10.3390/a16070351

Academic Editors: Gloria Cerasela Crisan, Ha Duy Long and Elena Nechita

Received: 24 June 2023
Revised: 13 July 2023
Accepted: 19 July 2023
Published: 22 July 2023

1. Introduction

Bike-sharing systems are one of the most popular and environmentally friendly forms of shared mobility. Traditional sharing systems allow the pickup and drop-off of bikes at fixed stations (or throughout an operational area if the system is free-floating) and have proven to be an effective solution for first-/last-mile mobility [1].

To keep a high quality of service, bike-sharing operators face two major problems, namely, the *optimal inventory problem* and the *rebalancing problem* [2]. The fact that bike-sharing users can take a bicycle from and return it to any station in the system leads to an imbalanced state, in which some stations are full while others stay empty. This non-homogeneous distribution of bicycles lowers the overall level of service of the system [3]. The *rebalancing problem* consists in reorganizing the fleet location in time and space to

re-establish the optimal level of service [4], an operation that can be performed when the system is shut down or the demand is low (static rebalancing) or while it is running (dynamic rebalancing) [1,5]. To reduce operational costs generated by rebalancing operations, it is fundamental to know the correct number of bicycles (and available docks if the service is station-based) required to achieve the optimal level of service, i.e., the target inventory values for the rebalancing procedure. This problem is known as the *optimal inventory problem* [6], and since inventory levels depend on user behavior [7], it is extremely challenging. While many rebalancing studies assume that the target inventory level for each station is known, to the best of the authors' knowledge, only a few studies have focused on combining rebalancing with target-level computation.

This paper tests specific machine learning (ML) techniques that estimate the inventory level required to address the station-based bike-sharing static rebalancing problem. Specifically, the use of decision tree classifiers—most notably a random forest classifier (RFC) and gradient tree-boosting classifier (GTBC)—is investigated. Due to the vast amount of data publicly available, decision trees have already been used to study bike-sharing systems [8,9]. However, the majority of machine learning models adopted in the literature (e.g., decision trees but also neural networks and support vector machines) assume the data to be well-balanced. This assumption is not met in the case of the inventory problem [10]. Therefore, in this paper, we propose an *ad hoc* iterative resampling procedure that allows to accurately predict inventory levels using several features, including historical data, in the case of imbalanced datasets. The main contributions of this paper are summarized below:

1. The inventory problem is formulated as a classification problem that can be easily solved using decision trees (or any other state-of-the-art classifier);
2. While traditional classifiers over-represent the majority class, this paper presents a novel resampling technique that better leverages data and provides better estimates for rare observations;
3. The proposed algorithm can be used to compute both an upper-bound and a lower-bound value for the bike-sharing inventory problem, thus yielding to different possible configurations;
4. Although mainly based on historical data, the proposed approach can also be used to solve the inventory problem for new stations, for which historical information is not available;
5. The proposed model is easily implementable into an ITS-based decision support system for also supporting bike-sharing companies in a day-to-day framework, thus helping in improving operations.

Points 4 and 5, i.e., predictions for new stations and the proposal of a rolling horizon approach to make day-to-day forecasts, are completely new developments with respect to the previous presentation of this work [10].

The method has been applied and validated using the data of the New York City Bike service.

The remainder of the paper is organized as follows: Section 2 introduces the relevant literature, namely, the existing research on inventory and rebalancing problems, but also provides a short overview of the most popular solutions to deal with imbalanced datasets in ML. As a consequence, the main gaps of the literature are identified, and possible solutions are discussed. The methodology is reported in Section 3: specifically, (i) how the target values of the inventory problem used as the benchmark in the paper are computed; (ii) the inventory problem formulation using ML classifiers to make previsions (for both existing and new stations); and (iii) the proposed iterative resampling technique to deal with data imbalance. The application and related results are reported in Section 4, followed by the conclusion.

2. Related Research

2.1. The Inventory Problem and the Rebalancing Problem

Developing an inventory model for a traditional station-based bike-sharing service is particularly challenging as it needs to capture two features: (1) the demand for bicycles; (2) the demand for docks—i.e., finding an empty dock where the user can return the rental bicycle [11]. Therefore, to simply find the number of bicycles that serves the demand is not sufficient [7].

The uneven distribution of vehicles and docks in the system causes some stations to be empty (or entirely full), creating shortages of both bicycles and docks. To prevent this shortage, several studies introduced rebalancing techniques that aim at evenly redistribute bicycles and docks in the system [12]. Traditionally, the problem is solved using either optimization techniques [12] or ML models [8]. Optimization techniques translate the problem into a mathematical language and focus on tractability and convergence properties. The most common approach is to formulate the problem as a one-commodity pickup-and-delivery capacitated vehicle-routing problem, which is then solved using mixed-integer linear programming [12]. This approach has also been extended to other station-based shared mobility services, such as car sharing [13]. The complexity of the algorithm depends on whether the objective is to achieve a complete rebalancing (all stations are jointly optimized) or a partial one (only a subset of stations is optimized), with the former case being far more complex than the latter [14]. Traditional algorithms for solving the problem include tabu search [15] and branch-and-cut [16]. However, as exact formulations are not suited for real-life instances, heuristic models have been developed to solve the problem in practice [9,17].

In recent years, other authors have proposed using ML instead of optimization techniques to solve the rebalancing problem. The most popular models include decision trees [8,9], neural networks [18], deep neural networks [19,20], and clustering techniques [2]. The main convenience of these models is that they are suited for large-scale, real-life applications and require limited assumptions of user behavior.

The models discussed up to this point also present several differences in the operational approach. For instance, [15] provides a price mechanism to support rebalancing, while [9] assumes that rebalancing operations are performed by the sharing company; these approaches are translated into different parameters and objective functions. However, most repositioning studies, including those using ML, assume that the inventory level (i.e., the optimal number of docks/bicycles at any given time) is known from historical data or by using an existing demand model [7]. To date, only a few studies have focused on how to compute the optimal target levels while considering the rebalancing problem.

Among the studies that have focused on the inventory levels, [21] proposed a mixed-integer program formulation to find the inventory levels that minimize the cost of rebalancing. Alternatively, [22] formulated the inventory problem as a nonstationary Markov chain model that computes the most likely optimal inventory levels during the day. Ref. [23] identifies an upper bound and a lower bound for the inventory level using historical information. A similar approach is used in [24]. Ref. [7] also identifies an upper bound and a lower bound for the inventory level using mixed-integer optimization. One of the main limits of the previous approaches, however, is that the model explicitly minimizes the journey dissatisfaction levels with respect to the user and cannot be used if the operator has different goals (such as maximizing profit).

Another aspect to highlight is the role of IT technologies. While bike-sharing systems date back to 1960, this service was initially unsuccessful due to several issues, e.g., vandalism and theft [25] The IT revolution not only enabled operators to develop a better service but also to improve aspects related to strategic and operational planning [26]. Thanks to the large amount of historical data, often openly available, it is now possible to optimize not only the fleet size and location of docks but also the entire supply chain of the system, from ordering vehicles and spare parts to scheduling fleet maintenance [27]. This is expected to become a primary problem in the future. Systems where the demand exceeds the capacity

may not require rebalancing. In addition, to investigate a potential unserved demand, operators will therefore focus on optimizing inventory levels while also considering the scheduled maintenance, hence reducing costs and increasing margins [28].

2.2. Learning from Imbalanced Datasets

Data Imbalance means that an uneven distribution of classes exists within the data, and it is a serious threat for classification problems, as standard classifiers assume the data to be well-balanced. In general, class imbalance is one of the greatest challenges in machine learning and data mining research [29] and can appear in two main forms: *rare classes* or *rare cases* [30]. The problem of rare classes refers to datasets that contain different proportions of observations (or instances) per class. The concept of rare cases refers instead to the sparse distribution of examples in the feature space [30]. The two problems are closely related, as they both result in an uneven distribution of observations. However, rare cases refer to data that is sparse by nature, while a rare class might simply depend on the sampling procedure.

Different approaches have been proposed in the literature to deal with imbalanced datasets, and they can be broadly divided into two: methods working at a learner level and methods working at a sampling level [31]. Methods working at a learner level modify an existing algorithm to increase the precision of the minority class. The most common approach is to use cost-sensitive approaches, in which the learner associates the rare class with some weights to compensate for data imbalance [32]. The main limitation of these methods is that they are designed for specific learners and are hard to generalize. Methods working at a sampling level are considered more general [33] as they use resampling to artificially rebalance the dataset. Over-sampling and under-sampling are the most common resampling techniques [34]; this approach consists in creating a balanced dataset by artificially generating new observations for the rare class. In the case of over-sampling, the algorithm creates new artificial data points for the minority class. The SMOTE (synthetic minority over-sampling technique) approach is one of the most common techniques [35]. The other main option is to use under-sampling procedures, which consists of using a subsample of the majority class [36]. One particularly advanced model is balance cascading [37]: in this algorithm, the model iteratively drops observations from the majority class that are correctly classified. The argument is that these observations are redundant and might negatively affect the quality of the classifier, making it biased toward the majority class.

2.3. Discussion

In this paper, we propose a novel model that uses ML—most notably decision trees—to address the inventory problem as a classification problem. The main argument for this decision is that inventory levels are usually estimated from historical data, and ML captures historical trends better than simple averages. Similar to other approaches presented in the literature, our model will compute upper bounds and lower bounds for the inventory level. Moreover, as it is based on data, it is not limited to one specific goal, as it usually happens when using optimization.

As previously reported, the main problem in using ML for solving the inventory problem is data imbalance. In the case of the inventory problem, we deal with both *rare cases* and *rare classes.* Though existing algorithms for data imbalance demonstrate promising advantages, they also have several disadvantages, most notably over-generalization [38]. Simply stated, dock stations characterized by a high demand strongly differ from those characterized by a low demand. No model will be able to properly predict both, which means that the model will either provide poor predictions or fit the dominant class. Case-specific algorithms, tailor-made to the problem, can address the over-generalization issue [39]. Decision trees are among the most interpretable ML classifiers as they allow to understand how each feature contributes to the classification effort. Differently from standard decision trees, random forest classifiers (RFCs) and gradient tree-boosting classifiers (GTBCs) combine multiple models to make predictions, which allow them to provide better output. Inspired

by these models, we proposed an iterative resampling algorithm that leverages multiple learners. Similar to balance cascading, at each iteration the model drops those observations that belong to the majority class and are properly classified. However, instead of simply dropping some observations and repeating the training exercise on the new dataset, the proposed model drops the majority class entirely and defines a new classification problem. Therefore, the model outputs an ensemble of models, each of them having different classes, different features, and different prediction capabilities. The final prediction is a combination of all these classifiers and provides more reliable predictions compared to using a single learner.

3. Methodology

The system considered is station-based with rebalancing activity conducted during the night (i.e., low demand, static rebalancing).

The proposed method to approach the inventory problem as a ML classification problem is here described. It will be adopted for both defining the target demand levels for each station and estimating the demand for new stations. The results of our proposed method will be compared with the results of common approaches based on historical data (benchmark values, as described in the following section). Finally, the resampling technique used to avoid data imbalance is presented.

3.1. Benchmark Values from Historical Data

The benchmark values from historical data are computed in terms of a set of options, from a lower bound (LB) to an upper bound (UB), never producing a lake of bikes or docks [7,22–24]. To compute the LB and UB, once departures and arrivals in a bike-sharing station are collected (Figure 1), the cumulative curves for departures and arrivals can be derived, as well as the cumulative net curve, i.e., the difference between arrivals and departures (Figure 2).

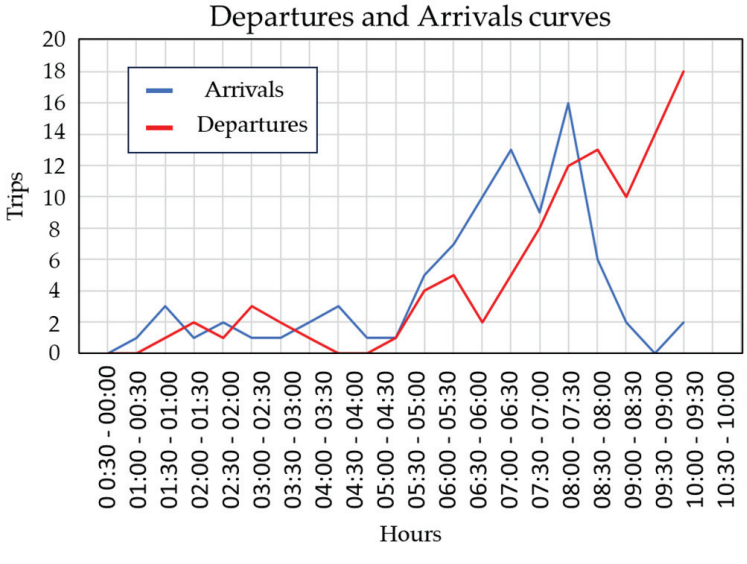


Figure 1. Hypothetical trend of bike departures and arrivals (Source: own elaboration).

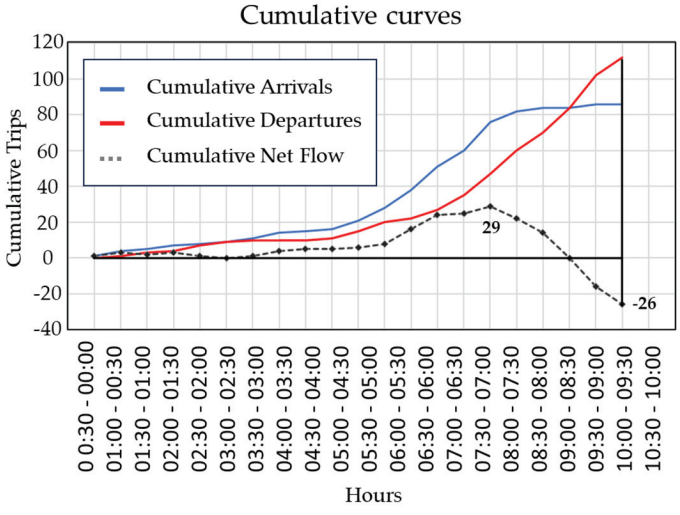


Figure 2. Hypothetical cumulative net flow (Source: own elaboration).

Considering the cumulative net curve, its highest value is the minimum number of docks required to respond to the arrival demand at the station (or the dock's lower bound—LB). Instead, its lowest value is the minimum number of bikes required to respond to the demand (bikes' LB). The upper bound (UB), for docks and bikes, can be computed as the difference between the capacity C of the station and, respectively, the LB for bikes and docks [10], as in the following equations:

$$LB_{dock} = \max(Cumulative\ net\ flow) \tag{1}$$

$$LB_{bikes} \begin{cases} min|Cumulative\ net\ flow|\ if\ Cumulative\ net\ flow < 0 \\ LB_{bikes} = 0\ if\ Cumulative\ net\ flow \geq 0 \end{cases} \tag{2}$$

$$UB_{bike} = \mathrm{C} - LB_{dock} \tag{3}$$

$$UB_{dock} = \mathrm{C} - LB_{bikes} \tag{4}$$

3.2. The Inventory Problem as a Classification Problem Using Machine Learning

An ML classifier is an algorithm that returns the probability $p(Y = k)$ that the dependent variable Y belongs to a certain class k. Without the loss of generality, in the case of a supervised classifier, we can write

$$p(Y = k) = \mathcal{H}(\boldsymbol{X}, \boldsymbol{\Theta}) \tag{5}$$

where $\mathcal{H}$ is a general nonlinear-model, $\boldsymbol{X}$ is the set of features (or independent variables) used to predict the probability $p(Y = k)$, and $\boldsymbol{\Theta}$ is a set of hyperparameters. The form of the model $\mathcal{H}$ depends on the type of classifier used (neural networks, decision trees, etc.) and the hyperparameters $\boldsymbol{\Theta}$, which explain the relationship between dependent and independent variables. To obtain the correct value of $\boldsymbol{\Theta}$, supervised models use a training set (i.e., a dataset where both dependent and independent variables are known) and compute the set of parameters $\boldsymbol{\Theta}$ that, given the training set, is more likely to reproduce the data. In the case of decision trees, Equation (5) can be rewritten as follows:

$$p(Y = k) = DT(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{\Theta}) \tag{6}$$

where $\boldsymbol{\theta}$ is the set of hyperparameters/weights that is associated with the feature vectors $\boldsymbol{X}$. Similar to linear regression or logistic regression, $\boldsymbol{\theta}$ represents the impact that each feature has on the prediction. This aspect related to interpretability makes decision trees among the simplest and most interpretable classifiers.

In order to use Equations (5) and (6) in the context of the inventory problem, it is necessary to define the dependent and independent variables. In the case of the dependent variables, our objective is to estimate the UB and LB of the demand for bikes/docks; therefore, we might want to set $Y^{UB} = UB$ and $Y^{LB} = LB$ in Equations (5) and (6). The problem is that classification problems require discrete variables while the LB/UB for bikes/docks are continuous ones. Therefore, we need to define classes for the UB and LB and transform Y^{UB}/Y^{LB} into categorical variables. To do so, we introduce the error term ε, which represents the expected precision of the model. Given ε, we can say that a certain value of the UB belongs to the class k if, and only if,

$$\begin{cases} UB \in k & if\ UB^k - \varepsilon < UB \leq UB^k + \varepsilon\ with\ UB^k - \varepsilon \geq 0 \\ UB \notin k & otherwise \end{cases} \tag{7}$$

where UB^k is the center of the class. To provide a numerical example, the first class will have a center equal to zero, therefore $UB^0 = 0$. Assuming an error term ε of 2 bikes, all target values $UB \leq 2$ will belong to Class 0. The second class will assume $UB^1 = 4$, therefore all observations $2 < UB \leq 6$ will belong to this class, and so on. Note that for $\varepsilon = 0.5$, each integer represents a separate class. The same procedure applies for the LB.

Creating independent variables is straightforward, as any feature can potentially be used within the ML classifier. In general, we argue that three features should be used in the context of the inventory problem:

$$\mathbf{X} = \left\{ \mathbf{X}_{behavioral}, \mathbf{X}_{endogenous}, \mathbf{X}_{exogenous} \right\} \tag{8}$$

The difference between endogenous and exogenous features is straightforward. Endogenous features are explained by other variables within the model (for instance, lagged variables such as the departures and arrivals collected in the station the day before), while exogenous variables are not explained by other variables within a model (e.g., location of the station).

The distinction between these two variables is not important when we want to predict the target values for an existing station, but it becomes relevant when the objective is to predict the demand for a new station, for which exogenous features might not be available. Finally, as the target levels depend on user behavior [7], some features that approximate user behavior will also help the model provide better estimates.

In this research, we have no access to behavioral features such as the value of time or individual preferences. Therefore, we use weather data to approximate $\mathbf{X}_{behavioral}$. The reason is that, in the case of bike sharing, it has been observed that user behavior is highly correlated to weather data, which allows to partially compensate for the lack of behavioral features [40].

To propose a formulation that can be deployed in practice, the classification problem proposed in Equation (6) can be rewritten as a time series and be used to make day-to-day predictions:

$$p\left(Y^d = k\right) = DT\left(\mathbf{X}^{d-1}, \mathbf{X}^{d-2}, \ldots, \mathbf{X}^{d-T}, \boldsymbol{\beta}, \boldsymbol{\Theta}\right) \tag{9}$$

where $\mathbf{X}^d$ is the set of features for a given day, T represents the time lag, i.e., how many past days are used to predict the demand for the next day, and Y^d is the prediction for day d. With respect to Equation (6), we have a new dataset every day, and the procedure must be repeated daily.

3.3. Iterative Resampling and Data Imbalance

Though traditional techniques for data imbalance provide many benefits, in the case of the inventory problem, off-the-shelf techniques might over-generalize the problem. Simply stated, they might identify a set of hyperparameters $\boldsymbol{\Theta}$ that is too general, as they assume that the same features that are important for the majority class are also important for the minority class. Therefore, this section introduces a resampling technique that generates and estimates, in an iterative manner, balanced data classes. The model focuses on a database-splitting function, and it is intuitively illustrated in Figure 3.

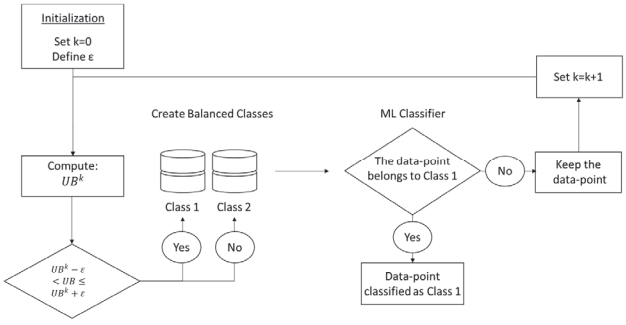


Figure 3. Illustrative example of the resampling technique (Source: own elaboration).

Starting from the entire dataset, the approach divides the data into 2 classes. Class 1 corresponds to $k = 0$, therefore $UB^0 = 0$. All the other data points are inserted in Class 2. The objective is to create two balanced classes of data. In Figure 3, we assume that two classes are sufficient. In practice, additional classes are created until all classes are the same size. However, only Class 1 is computed using the system of Equation (6), therefore UB^k and ε, while all other classes are computed to ensure balance within the data, i.e., the only criterion to create all other classes is that they must have the same number of data points. At each iteration, a decision tree classifier is used to classify the data. All data points that are classified as Class 1 are considered properly classified. All remaining observations are included in a new dataset. At this point, we set $k = k + 1$, and we repeat the operation on the new reduced dataset. The operation continues until all data points have been classified. As Figure 3 is a purposely trivial example, the procedure is depicted in Algorithms 1 and 2, where E is the tolerance error for the class imbalance, i.e., how much class imbalance is allowed in the system.

Algorithm 1: Iterative Resampling Technique

Procedure: resampling (**X**, **Y**, ε, k, E)
For UB in **Y**

$$UB^k = UB^{k-1} + 2\varepsilon$$

if $UB^k - \varepsilon < UB \leq UB^k + \varepsilon$
$UB \in Class^1$
Else
$UB \in Class^2$
Set Len = length$\left(Class^2\right)$
If length$\left(Class^1\right)$ < Len + E
Set n = Len/length$\left(Class^1\right)$
Split $Class^2$ *in* n classes
If length$\left(Class^1\right)$ > Len − E
Set n = length$\left(Class^1\right)$/Len
Split $Class^1$ *in* n classes
Return (**X**, **Class**)

Algorithm 2: Iterative ML Classifier

Procedure : $\mathcal{HIT}(\mathbf{X}, \mathbf{Y}, \boldsymbol{\Theta})$
Set ε, k, E
Set $k = 0$
While $(\mathbf{X}, \mathbf{Y})$ is not empty:
ClassLabel $=$ resampling$(X, Y, \varepsilon, k, E)$
Train ML Classifier $DT(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\Theta})$
For X, Y in$Class^1$:
remove X, Y from $\mathbf{X}, \mathbf{Y}$
Set $k = k + 1$

It should be noted that, as the lower and upper bounds are computed using the cumulative net flow and not the demand, stations with different demand levels (e.g., in the city center or in the suburbs) may still have similar inventory levels and therefore be grouped in the same class. This could create issues during the classification, as these stations may exhibit a different behavior. Therefore, the model should ideally incorporate endogenous features, as previously mentioned, to account for this phenomenon.

The resampling function needs to be used together with the learner in an iterative manner. This is illustrated in Algorithm 2. Specifically, the function determines a balanced group of classes to be forecasted at each iteration. Once a maximum number of classes is defined for each prediction group, the function generates the set of classes that have to be predicted (A) and the set to predict later (B). At each iteration, the machine learning classifier is adopted to forecast (A), and the process goes ahead on the group (B). The procedure is repeated until all points have been properly predicted. An illustrative example is presented in Figure 4.

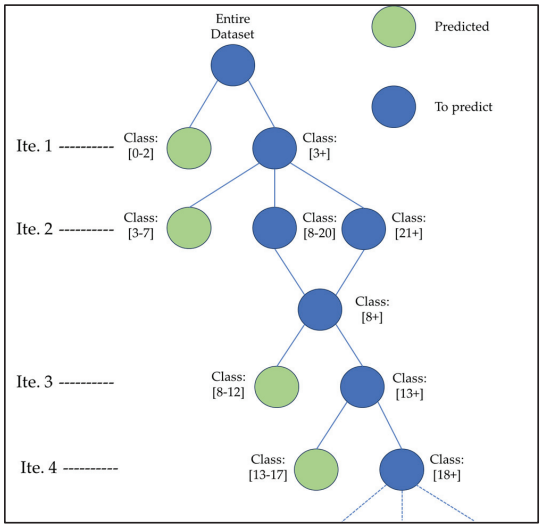


Figure 4. Example of the iterative procedure (Source: own elaboration).

The proposed resampling procedure can be considered as a hybrid model, as it works both at a learner level and at the sampling level. Specifically, at each iteration, a different model is trained. For instance, the models used in Figure 4 to predict Class (0–2) and Class (3–7) are different. At the same time, any classifier can be used, as the model leverages different training sets at each iteration and does not modify the learner. Similarly, it should be noted that the iterative resampling procedure is used on the training set, as it requires knowledge of both dependent and independent variables.

4. Numerical Results

4.1. Case Study

The methods discussed in the previous section are now tested adopting real data from the Citi Bike station-based service in New York City, US (about 900 stations and 14,500 shared bikes available). In particular, the database for the machine learning derives from over 17 million rides during 2018.

The trained models estimate the LB/UB of bikes adopting the available features reported in Table 1. Given the capacity at the station and the LB/UB of bikes, the number of docks is then calculated as in Equations (3) and (4). A correlation analysis of the features is performed to avoid redundant variables. All the ML algorithms, unless differently indicated, adopt 90% of the data as the training set and 10% for testing (randomly selected). Also, the benchmark values have been computed on the same 10% dataset, thus allowing for the comparison of the results.

Table 1. Available features aggregated as a function of the feature type (adapted from [10]).

Feature Type	Characterization	[Unit]
Exogenous	Month	
	Weekday	
	Season	
	Capacity of the station	[number of bikes]
	Nearby stations	[number]
	Station location (inside/outside Manhattan)	[binary]
Endogenous	Average number of departures/arrivals observed in the previous two months, in the same referenced period	[number]
	Number of departures/arrivals at the station in the previous day	[number]
	Departure/arrival trip duration observed at the station the previous two months	[seconds]
	Average LB (or UB) observed in the previous two months, in the same referenced period and at the same station	[number of bikes]
Behavioral *	Average temperature (current day)	[Classes from 1(Low) to 10 (High)]
	Average temperature (day before)	[Classes from 1(Low) to 10 (High)]
	Variability of the temperature (current day), as the difference between the maximum and minimum registered	[Classes from 1(Low) to 5 (High)]
	Variability of the temperature (day before), as the difference between the maximum and minimum registered	[Classes from 1(Low) to 5 (High)]
	Precipitation (current day)	[Classes 0 (Null) and from 1(Low) to 5 (High)]
	Precipitation (day before)	[Classes 0 (Null) and from 1(Low) to 5 (High)]
	Snow depth (current day)	[Classes 0 (Null) and from 1(Low) to 5 (High)]
	Snow depth (day before)	[Classes 0 (Null) and from 1(Low) to 5 (High)]
Dependent variable	Bikes LB (or UB)	[number of bikes]

* Weather data are assumed to approximate behavior, e.g., more rides when the temperature is high, less rides when the temperature is low.

In the next subsections, we first analyze the data adopted in this research; the aim is twofold: (i) to underline if imbalance exists and (ii) if different features can impact the model explanation as a function of the considered class of the dependent variable. Then, the results are reported and, specifically, the following:

- The computation of the UB and LB of the inventory problem by using different decision trees, i.e., RFC and GTBC, with and without combining them with a standard resam-

pling technique (BorderlineSMOTE [34]) or with the iterative resampling approach discussed in Section 2;
- The prediction for new stations by adopting the best classifier as a result of the first point, again combining it with a standard resampling technique (BorderlineSMOTE) or with the iterative resampling approach;
- The first results in terms of predictions in a day-to-day framework as a result of applying Equation (9).

The results are reported only in terms of bikes since docks can be calculated as the difference between the capacity at the station and the number of bikes.

4.2. Data Imbalance

Table 2 shows the percentage of observations as a function of the number of bikes for the UB and LB. Considering the LB, it appears evident that the dataset is highly imbalanced and that low values are dominant. Therefore, it is expected that traditional ML classifiers will prioritize the larger classes and that over-sampling techniques will be required. The observations for the UB computation are more balanced than for the LB, hence traditional ML should correctly classify the data.

Table 2. Distribution of the upper-bound and lower-bound observation in the dataset.

Number of Bikes	Upper-Bound Distribution [%]	Lower-Bound Distribution [%]
(0–10)	6.2	83.1
(11–20)	18.6	10.4
(21–30)	39.2	4.5
(31–40)	19.2	1.5
(41–50)	11.3	0.3
(>50)	5.5	0.2

4.3. Resampling and Feature Importance

This subsection demonstrates how different demand classes may depend on different features. We adopt a random forest classifier (RFC) which performs feature selection based on correlation analysis. Figure 5 shows the relative importance of some features along the iterations of the algorithm, with Iteration 0 being the model that predicts the majority class (low demand values), and Iteration 32 is the model that forecasts the rarest class (high demand values). Only three main features have been shown in order to point out how their importance changes along iterations.

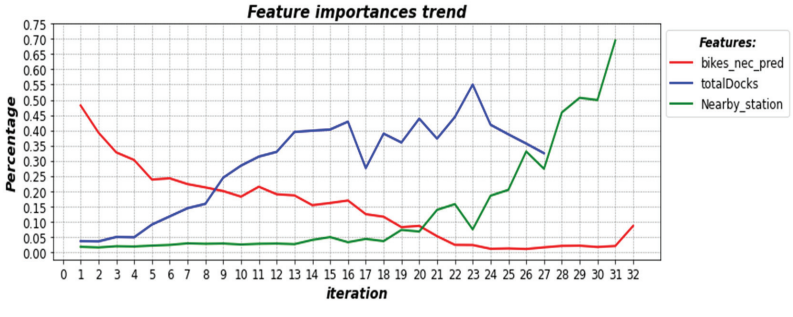


Figure 5. Relative feature importance with the number of iterations of the model.

The feature "*Average LB (or UB) observed in the previous two months*" (in the table—*bikes_nec_pred)* is the most important one in the first iterations (low demand), while it becomes irrelevant when the goal is to predict large volumes of bikes (Iteration 32). The opposite trend can be observed for other features, such as the *Capacity of the station* (in the

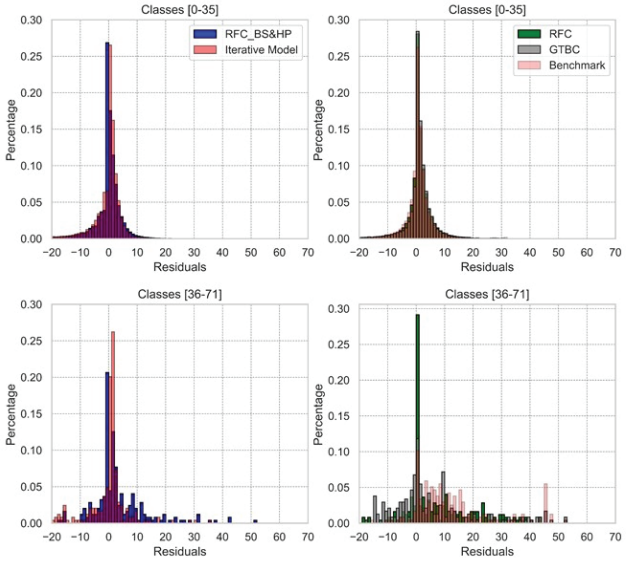
table—*totalDocks*), which is not relevant for small demand values but becomes dominant when we deal with large volumes of bikes. Note also that the features *Nearby_station* and *totalDocks* became even more correlated, and therefore at Iteration 23, the latter becomes redundant, and it is removed, allowing for the increase in the relevance of the *Nearby_station* feature.

Thus, while different features can impact different demand levels, general purpose resampling methods would not perceive this difference and therefore would not be able to use the best features during the prediction phase. This is the reason for proposing the iterative resampling approach that provides more flexibility when it comes to feature importance, allowing to capture the difference between features that are good at explaining the majority class, the rare class, or both.

4.4. Prediction of the Upper Bound and Lower Bound for Existing Stations

In this section, the UB and LB are calculated using the methods discussed in Section 2. Specifically, decision trees such as GTBC and RFC are firstly implemented with the features of Table 1.

For clarity of analysis, the results (Figures 6 and 7) are illustrated dividing the class with the low number of bikes from the class with the high number of bikes, where the division was performed according to half of the observed UB/LB. Hence, the MAE values can be compared (Figure 7), highlighting the accuracy for each class. Please note that these classes are not those used to solve the classification exercise. In that case, the classes are defined as discussed in Section 2, assuming $\varepsilon = 2$.



	[0-35]				
	Benchmark	Iterative	Borderline Smooth	Gradient Boost	Random Forest
Precision	0.2	0.2	0.22	0.19	0.19
Recall	0.19	0.21	0.22	0.18	0.17
Fscore	0.19	0.2	0.21	0.18	0.18

	[36-71]				
	Benchmark	Iterative	Borderline Smooth	Gradient Boost	Random Forest
Precision	0.28	0.27	0.23	0.18	0.32
Recall	0.12	0.32	0.16	0.11	0.27
Fscore	0.16	0.27	0.13	0.1	0.27

Figure 6. Distribution of the residuals for the LB, Classes (0–35) and (36–71), and related prediction metrics.

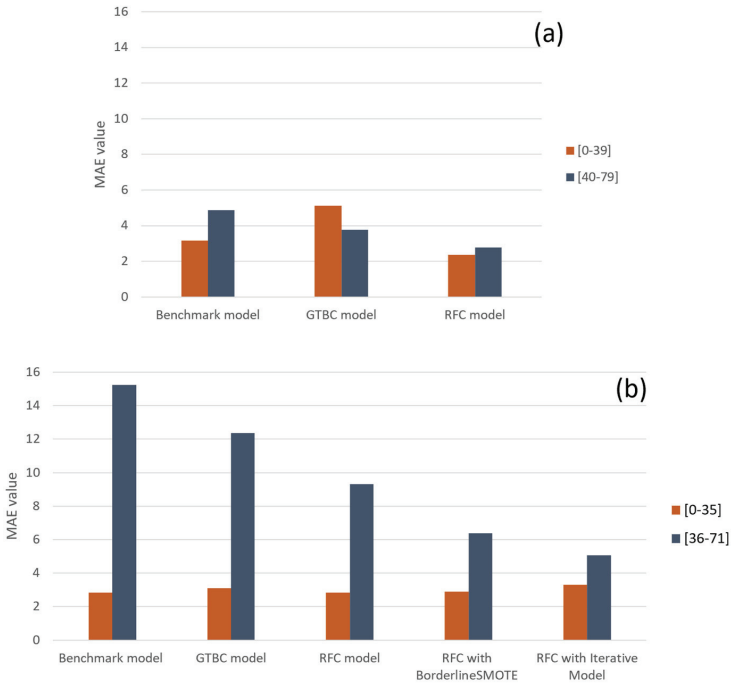


Figure 7. MAE values for the UB (**a**) and LB (**b**) with and without resampling for existing stations.

In the UB estimation case, no rebalancing technique was needed since the database was sufficiently balanced. All models, the benchmark included, returned good forecasts of bikes' UB, nevertheless a slight underestimation is observed (Figure 7a).

With the benchmark model, good predictions are obtained; however, the RFC was the best performing algorithm. Specifically, it adjusted the benchmark estimation by deriving information thanks to the features additionally adopted in the calibration. On the other hand, the GTBC was quite poor, especially with respect to the class with the highest number of bikes; thus, the RFC should be used in this case.

The situation is different for the LB (Figures 6 and 7b). In this case, the database was imbalanced (Table 2) toward the low values of bikes; thus, resampling techniques have been used. These techniques, specifically the BorderlineSMOTE and our proposed iterative model, have been combined with the RFC. The BorderlineSMOTE is the reference model for resampling, as it represents a common method for data imbalance.

As expected for the LB prediction, when looking at the benchmark, at the RFC and the GDBC, the MAE is fairly low for the classes with the highest number of bikes, while it is at least three times larger for the other ones. From the distribution of the residuals for each model (Figure 6), it emerges clearly.

Concerning data unbalancing, we can also observe that both the BorderlineSMOTE and the iterative model perform better than the other models. Nevertheless, the pro-posed iterative model clearly outperforms the BorderlineSMOTE. This is because the Borderline SMOTE shows a smaller error for the dominant class (MAE (0–35)) and a larger error for the minority one (MAE (36–71)), which is a clear indication of the BorderlineSMOTE model overfitting the dominant class.

4.5. Predicting New Stations Using Only Exogenous Variables

One of the main problems when using ML is the lack of endogenous variables. For instance, if a new station appears in the system, the endogenous features presented in

Table 1 cannot be used to predict inventory levels. Therefore, in this section, we test the same model as discussed in the previous section but with only exogenous and behavioral features. This model can be used, for instance, to predict the inventory levels when historical data are not available, as in the case of a new station.

Table 3 shows the numerical results. As in the previous case, for the LB calculation, the RFC alone and in combination with the iterative resampling and BorderlineSMOTE has been used to make predictions. As expected, the results look worse than in the previous case, especially for the rare classes (36–71). When looking at the RFC, the MAE is fairly low for (0–35), while it is almost 10 times larger for (36–71), showing that data imbalance becomes more relevant when only exogenous variables are available. Concerning the other models, in this case, the BorderlineSMOTE performs better than the proposed iterative model for the dominant class, while it performs worse in terms of rare classes. This is related to the generalization problem and shows, as for the previous test, the tendency of the model to overfit the dominant class. With or without endogenous variables, the proposed iterative procedure achieves similar results for the minority and majority classes, showing that the model is less sensitive to overfitting. To remove possible collinearity issues and assess how the features affect the model, the approach was tested using different correlation cuts (see Table 4).

Table 3. Results for the LB (with and without resampling) for new stations.

Lower-Bound Models	MAE (0–35)	MAE (36–71)
RFC	4.85	37.83
RFC with BorderlineSMOTE	6.68	9.07
RFC with Iterative Model	8.41	8.36

Table 4. Results for the LB (with and without resampling) for new stations.

	RFC with BorderlineSMOTE		RFC with Iterative Model	
Correlation Cut	MAE (0–35)	MAE (36–71)	MAE (0–35)	MAE (36–71)
0.1	6.66	10.67	8.18	8.19
0.3	6.68	9.07	8.41	8.36
0.5	6.73	8.94	8.31	8.43
0.7	6.66	9.40	8.16	8.51
1	6.66	8.81	8.19	8.44
1.2	10.45	21.39	11.43	8.17

Multicollinearity occurs when multiple features utilized by the ML classifier are strongly correlated. When features are correlated, they are unable to individually provide independent predictions for the dependent variable. Instead, they jointly explain a portion of the variance, thereby diminishing their individual statistical significance. In this section, multicollinearity is accounted for by performing hierarchical clustering on the Spearman's rank-order correlations, picking a threshold, and keeping a single feature from each cluster. As hierarchical clustering computes the information loss associated with aggregating two features, high thresholds will translate into larger clusters and higher information loss. This threshold is called a 'correlation cut', and it is one of the parameters of the model that we used to avoid overfitting. As discussed at the beginning of this section, during each iteration, we use feature selection to select which features should be used and which features should be excluded to avoid overfitting. A high correlation cut translates into a model with less features. Intuitively, more features (i.e., a low correlation cut) implies more overfitting while less features (i.e., high correlation cut) lead to poor model performance. The results confirm that the BorderlineSMOTE tends to overfit the dominant class with respect to the minority class and, in general, model performance is heavily influenced by the adopted correlation cut. The proposed iterative model, which makes predictions on an ensemble of classifiers and weights, provides more balanced predictions and an error that

is similar for the majority and minority classes. It can also be observed that, for all models, the error is maximum when the *correlation cut* = 1.2. This is reasonable, as when too many features are grouped together, the model is not able to sufficiently generalize from the data.

4.6. Prediction Based on a Day-to-Day Approach

Finally, we test the model using the day-to-day approach as described in Equation (8). The results are depicted in Table 5. Note that, in this case, the dominant class and the minority class of the lower bound have different values than in the previous experiment (MAE (0–39) and MAE (40–79) instead of MAE (0–35) and MAE (36–71) in Tables 3 and 4). The reason is that, in this experiment, we use a different dataset. More specifically, this experiment uses data from 2019, while the previous one focused on the number of rides in 2018. The definition of the dominant class in the two experiments is the same. However, the interval is different, which also reflects an increase in the demand for bike-sharing services in 2019 with respect to 2018.

Table 5. Results for the LB (best classifier with iterative resampling model) previsions for new stations.

Lower-Bound Models	MAE (0–39)	MAE (40–79)
RFC with Iterative Model	3.27	4.79

It should be noted that the results are shown only for the iterative model; this is because firstly the comparison between resampling approaches was already presented in the previous subsections. Secondly, the BorderlineSMOTE was extremely time-consuming and not applicable in practice for a day-to-day framework. Therefore, Table 4 only validates what was described before, showing that the proposed formulation is a good method to compute inventory levels and that the iterative approach also performs well in the case of a day-to-day framework.

5. Conclusions

The inventory problem is a challenge for bike-sharing operators. The problem, which consists in estimating the total number of bikes necessary at each bike-sharing station, is complex for two reasons. First, for traditional station-based systems, it is necessary to estimate both the number of bikes as well as the number of bicycles, which makes the problem more complex. Second, the demand for bicycles is sparse, meaning that many stations are empty while a few have very high demand values. While researchers agree that the inventory problem is a key issue, this information is usually obtained from historical data. Therefore, in this paper, we proposed using machine learning (ML) as a more accurate way of extracting this information from historical data. Specifically, we formulate the inventory problem as a classification problem that can be solved using any state-of-the-art classifier ML model. We also developed an iterative resampling technique to deal with the problem of the sparsity of the demand, which is a main problem when using ML classifiers.

The model is tested using real-world data from Citi Bike, the bike-sharing system that is currently in service in New York, US. The model provides estimates for the inventory problem in terms of the upper bound and lower bound of bikes. The results suggest that the proposed approach is robust in terms of results and can be applied in several circumstances, including opening new bike-sharing stations and day-to-day operations. The current research has two main limitations. First, it has been tested on a station-based system. Second, it has been adopted for the solution of the static inventory problem, i.e., estimating during the nighttime the optimal inventory levels for the morning. Future research will therefore focus on testing with different data, different operational settings, and in the case of real-time problems. A relevant future research direction is also to use clustering to identify similar stations. While in this research we focused primarily on resampling, it appears obvious that it is irrelevant to compute the inventory levels for stations that tend to naturally rebalance themselves, therefore requiring no action from

the operator. As multiple stations have a lower bound of zero, it would be relevant to use clustering to identify these stations. This would allow to remove these stations from the dataset and remove or at least substantially reduce data imbalance.

Author Contributions: Conceptualization, G.C. (Guido Cantelmo); methodology, G.C. (Guido Cantelmo), M.N. and C.A.; formal analysis, G.C. (Giovanni Ceccarelli); data curation, G.C. (Giovanni Ceccarelli); writing—original draft preparation, G.C. (Giovanni Ceccarelli) and G.C. (Guido Cantelmo); writing—review and editing, G.C. (Giovanni Ceccarelli), G.C. (Guido Cantelmo) and M.N.; supervision, G.C. (Guido Cantelmo), M.N. and C.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the authors.

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

References

1. Loaiza-Monsalve, D.; Riascos, A.P. Human Mobility in Bike-Sharing Systems: Structure of Local and Non-Local Dynamics. *PLoS ONE* **2019**, *14*, e0213106. [CrossRef]
2. Lahoorpoor, B.; Faroqi, H.; Sadeghi-Niaraki, A.; Choi, S.-M. Spatial Cluster-Based Model for Static Rebalancing Bike Sharing Problem. *Sustainability* **2019**, *11*, 3205. [CrossRef]
3. Fricker, C.; Gast, N.; Mohamed, H. Mean Field Analysis for Inhomogeneous Bike Sharing Systems. *Discret. Math. Theor. Comput. Sci.* **2012**. Available online: https://dmtcs.episciences.org/3006/pdf (accessed on 23 June 2023). [CrossRef]
4. Cruz, F.; Subramanian, A.; Bruck, B.P.; Iori, M. A Heuristic Algorithm for a Single Vehicle Static Bike Sharing Rebalancing Problem. *Comput. Oper. Res.* **2017**, *79*, 19–33. [CrossRef]
5. Regue, R.; Recker, W. Proactive Vehicle Routing with Inferred Demand to Solve the Bikesharing Rebalancing Problem. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *72*, 192–209. [CrossRef]
6. Legros, B. Dynamic Repositioning Strategy in a Bike-Sharing System; How to Prioritize and How to Rebalance a Bike Station. *Eur. J. Oper. Res.* **2019**, *272*, 740–753. [CrossRef]
7. Datner, S.; Raviv, T.; Tzur, M.; Chemla, D. Setting Inventory Levels in a Bike Sharing Network. *Transp. Sci.* **2019**, *53*, 62–76. [CrossRef]
8. Ashqar, H.I.; Elhenawy, M.; Almannaa, M.H.; Ghanem, A.; Rakha, H.A.; House, L. Modeling Bike Availability in a Bike-Sharing System Using Machine Learning. 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; pp. 374–378.
9. Ruffieux, S.; Spycher, N.; Mugellini, E.; Khaled, O.A. Real-Time Usage Forecasting for Bike-Sharing Systems: A Study on Random Forest and Convolutional Neural Network Applicability. In Proceedings of the 2017 Intelligent Systems Conference (IntelliSys), London, UK, 7–8 September 2017; pp. 622–631.
10. Ceccarelli, G.; Cantelmo, G.; Nigro, M.; Antoniou, C. Machine Learning from Imbalanced Data-Sets: An Application to the Bike-Sharing Inventory Problem. In Proceedings of the 2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Heraklion, Greece, 16–17 June 2021; pp. 1–6.
11. Laporte, G.; Meunier, F.; Calvo, R.W. Shared Mobility Systems. *4OR* **2015**, *13*, 341–360. [CrossRef]
12. Dell'Amico, M.; Hadjicostantinou, E.; Iori, M.; Novellani, S. The Bike Sharing Rebalancing Problem: Mathematical Formulations and Benchmark Instances. *Omega* **2014**, *45*, 7–19. [CrossRef]
13. Santos, G.G.D.; Correia, G.H.D.A. Finding the Relevance of Staff-Based Vehicle Relocations in One-Way Carsharing Systems through the Use of a Simulation-Based Optimization Tool. *J. Intell. Transp. Syst.* **2019**, *23*, 583–604. [CrossRef]
14. Pal, A.; Zhang, Y. Free-Floating Bike Sharing: Solving Real-Life Large-Scale Static Rebalancing Problems. *Transp. Res. Part C Emerg. Technol.* **2017**, *80*, 92–116. [CrossRef]
15. Chemla, D.; Meunier, F.; Calvo, R.W. Bike Sharing Systems: Solving the Static Rebalancing Problem. *Discret. Optim.* **2013**, *10*, 120–146. [CrossRef]
16. Erdoğan, G.; Battarra, M.; Calvo, R.W. An Exact Algorithm for the Static Rebalancing Problem Arising in Bicycle Sharing Systems. *Eur. J. Oper. Res.* **2015**, *245*, 667–679. [CrossRef]
17. Kloimüllner, C.; Papazek, P.; Hu, B.; Raidl, G.R. Balancing Bicycle Sharing Systems: An Approach for the Dynamic Case. In *Evolutionary Computation in Combinatorial Optimisation*; Lecture Notes in Computer, Science; Blum, C., Ochoa, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 73–84.
18. Chen, P.; Hsieh, H.; Su, K.; Sigalingging, X.K.; Chen, Y.; Leu, J. Predicting Station Level Demand in a Bike-Sharing System Using Recurrent Neural Networks. *IET Intell. Transp. Syst.* **2020**, *14*, 554–561. [CrossRef]
19. Wang, B.; Vu, H.L.; Kim, I.; Cai, C. Short-Term Traffic Flow Prediction in Bike-Sharing Networks. *J. Intell. Transp. Syst.* **2022**, *26*, 461–475. [CrossRef]

20. Xu, C.; Ji, J.; Liu, P. The Station-Free Sharing Bike Demand Forecasting with a Deep Learning Approach and Large-Scale Datasets. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 47–60. [CrossRef]
21. Nair, R.; Miller-Hooks, E. Fleet Management for Vehicle Sharing Operations. *Transp. Sci.* **2011**, *45*, 524–540. [CrossRef]
22. Schuijbroek, J.; Hampshire, R.; van Hoeve, W.-J. Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. *Eur. J. Oper. Res.* **2017**, *257*, 992–1004. [CrossRef]
23. O'Mahony, E.; Shmoys, D.B. Data Analysis and Optimization for (Citi) Bike Sharing. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, Austin, TX, USA, 25–30 January 2015; AAAI Press: Washington, DC, USA, 2015; pp. 687–694.
24. Rudloff, C.; Lackner, B. Modeling Demand for Bikesharing Systems: Neighboring Stations as Source for Demand and Reason for Structural Breaks. *Transp. Res. Rec.* **2014**, *2430*, 1–11. [CrossRef]
25. Ploeger, J.; Oldenziel, R. The sociotechnical roots of smart mobility: Bike sharing since 1965. *J. Transp. Hist.* **2020**, *41*, 134–159. [CrossRef]
26. Moran, M.E.; Laa, B.; Emberger, G. Six scooter operators, six maps: Spatial coverage and regulation of micromobility in Vienna, Austria. *Case Stud. Transp. Policy* **2020**, *8*, 658–671. [CrossRef]
27. Li, L.; Liu, Y.; Song, Y. Factors affecting bike-sharing behaviour in Beijing: Price, traffic congestion, and supply chain. *Ann. Oper. Res.* **2019**, 1–16. [CrossRef]
28. Jin, Y.; Ruiz, C.; Liao, H. A simulation framework for optimizing bike rebalancing and maintenance in large-scale bike-sharing systems. *Simul. Model. Pract. Theory* **2022**, *115*, 102422. [CrossRef]
29. Jamali, I.; Bazmara, M.; Jafari, S. Feature Selection in Imbalance Data Sets. *Int. J. Comput. Sci. Issues* **2013**, *9*, 42.
30. Orriols-Puig, A.; Bernadó-Mansilla, E. Evolutionary Rule-Based Systems for Imbalanced Data Sets. *Soft Comput.* **2009**, *13*, 213–225. [CrossRef]
31. Krawczyk, B. Learning from Imbalanced Data: Open Challenges and Future Directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
32. Zhou, Z.-H.; Liu, X.-Y. On Multi-Class Cost-Sensitive Learning. *Comput. Intell.* **2010**, *26*, 232–257. [CrossRef]
33. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
34. Ganganwar, V. An Overview of Classification Algorithms for Imbalanced Datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
35. Wang, J.; Xu, M.; Wang, H.; Zhang, J. Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding. In Proceedings of the 2006 8th international Conference on Signal Processing, Guilin, China, 16–20 November 2006; Volume 3.
36. Dal Pozzolo, A.; Caelen, O.; Johnson, R.A.; Bontempi, G. Calibrating Probability with Undersampling for Unbalanced Classification. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 159–166.
37. Liu, X.-Y.; Wu, J.; Zhou, Z.-H. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* **2009**, *39*, 539–550. [CrossRef]
38. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
39. Chen, H.; Li, C.; Yang, W.; Liu, J.; An, X.; Zhao, Y. Deep Balanced Cascade Forest: A Novel Fault Diagnosis Method for Data Imbalance. *ISA Trans.* **2021**, *126*, 428–439. [CrossRef] [PubMed]
40. Cantelmo, G.; Kucharski, R.; Antoniou, C. Low-Dimensional Model for Bike-Sharing Demand Forecasting That Explicitly Accounts for Weather Data. *Transp. Res. Rec.* **2020**, *2674*, 132–144. [CrossRef]

algorithms

MDPI

Article

A Plant Disease Classification Algorithm Based on Attention MobileNet V2

Huan Wang [1], Shi Qiu [1,*], Huping Ye [2,3] and Xiaohan Liao [2,3,4]

[1] Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China; wanghuan@opt.ac.cn
[2] State Key Laboratory of Resources and Environment Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; yehp@igsnrr.ac.cn (H.Y.); liaoxh@igsnrr.ac.cn (X.L.)
[3] Key Laboratory of Low Altitude Geographic Information and Air Route, Civil Aviation Administration of China, Beijing 100101, China
[4] The Research Center for UAV Applications and Regulation, Chinese Academy of Sciences, Beijing 100101, China
* Correspondence: qiushi@opt.ac.cn

Abstract: Plant growth is inevitably affected by diseases, and one effective method of disease detection is through the observation of leaf changes. To solve the problem of disease detection in complex backgrounds, where the distinction between plant diseases is hindered by large intra-class differences and small inter-class differences, a complete plant-disease recognition process is proposed. The process was tested through experiments and research into traditional and deep features. In the face of difficulties related to plant-disease classification in complex backgrounds, the advantages of strong interpretability of traditional features and great robustness of deep features are fully utilized, and include the following components: (1) The OSTU algorithm based on the naive Bayes model is proposed to focus on where leaves are located and remove interference from complex backgrounds. (2) A multi-dimensional feature model is introduced in an interpretable manner from the perspective of traditional features to obtain leaf characteristics. (3) A MobileNet V2 network with a dual attention mechanism is proposed to establish a model that operates in both spatial and channel dimensions at the network level to facilitate plant-disease recognition. In the Plant Village open database test, the results demonstrated an average SEN of 94%, greater than other algorithms by 12.6%.

Keywords: plant disease; classification; MobileNet V2; attention

Citation: Wang, H.; Qiu, S.; Ye, H.; Liao, X. A Plant Disease Classification Algorithm Based on Attention MobileNet V2. *Algorithms* **2023**, *16*, 442. https://doi.org/10.3390/a16090442

Academic Editors: Gloria Cerasela Crisan, Ha Duy Long and Elena Nechita

Received: 24 August 2023
Revised: 10 September 2023
Accepted: 12 September 2023
Published: 13 September 2023

1. Introduction

With the proliferation of artificial-intelligence technology, the development of smart agriculture has gained momentum. In the area of plant-disease recognition, researchers have conducted much research, which can be divided into two approaches: classification and clustering.

Classification perspective: Based on the analysis of inter-class differences, plant-disease classification can be mainly divided into traditional-feature and deep-feature aspects. Representative algorithms based on traditional features include: Al-Hiary et al. [1] analyzed the typical characteristics of plants and proposed a fast classification algorithm for plants; Kulkarni et al. [2] constructed a classifier founded on texture features extracted from plant images; Arivazhagan et al. [3] classified plant health status based on texture features; Hossain et al. [4] achieved plant-disease classification through leaf color information analysis; Singh et al. [5] implemented swift plant-disease detection from the algorithmic level based on image segmentation and soft computing techniques; Kaur et al. [6] implemented plant-disease detection based on gradient and texture features; Nanehkaran et al. [7] formulated a visual model for disease analysis and correlation assessment; Pujari et al. [8]

extracted plant image features based on SVM and ANN; Brahimi et al. [9] focused on the salient area of plants, established a saliency map, and achieved plant-disease classification; Mahmoud et al. [10] established a disease image representation using inverse coding technology; and Sandesh et al. [11] constructed an Adaboost-based model for disease prediction from the perspective of color. Representative algorithms based on deep feature extraction mainly include: Hang et al. [12] proposed a CNN-based method for plant-disease analysis; Atila et al. [13] devised an EfficientNet deep learning model to mine image depth features; Sardogan et al. [14] conducted research based on CNN with LVQ algorithm; Deepa et al. [15] enhanced images and established an interactive model to facilitate disease classification; ALTAN et al. [16] constructed capsule networks to measure the efficacy in plant-disease classification; Pal et al. [17] established the semantic relationship between images and diseases in AgriDet; and Liang et al. [18] introduced a deep-learning network for plant-disease classification and severity assessment.

Clustering perspective: Models are built by an analysis of intra-class differences. Representative algorithms based on traditional features include: Yu et al. [19] constructed a K-means model to analyze intra-class differences and achieve clustering; Padol et al. [20] established an SVM to cluster different disease images; Rani et al. [21] enhanced clustering accuracy by adding SVM on top of the K-means algorithm; Trivedi et al. [22] established a model from the perspective of a color histogram for image analysis; Faithpraise et al. [23] established a K-means model for disease classification from a clustering perspective; Tamilselvi et al. [24] used unsupervised machine-learning algorithms to cluster based on color features; and Hasan et al. [25] proposed an extended kernel-density-estimation approach to analyze disease morphology. On the other hand, representative algorithms based on deep feature extraction mainly include: Yadhav et al. [26] obtained clustering features based on the CNN model with optimized activation functions; Bhimavarapu et al. [27] fused PSO and CNN algorithms to extract multi-dimensional features; Hatuwal et al. [28] experimentally demonstrated the capabilities of random forest, KNN, SVM, and CNN for clustering; Pareek et al. [29] established a 1D-CNN model for clustering based on image segmentation; Mukti et al. [30] achieved plant-disease detection based on multiple iterations of ResNet; Li et al. [31] analyzed plant diseases through the construction of the model ensemble with inception module and cluster algorithm; Türkoğlu et al. [32] used deep networks to extract disease image features and analyze differences between classes; and Ramesh et al. [33] constructed a model from the perspective of image and machine learning to achieve disease classification.

In summary, plant-disease classification algorithms based on images face the following issues: (1) Traditional algorithms are based on visual features that can be easily affected by natural factors, such as lighting and angles, with limited performance improvement. (2) Deep-learning algorithms based on neural conduction processes exhibit the features of strong robustness and positive effects. However, the classification effect remains to be improved in the face of complex background interference.

Through an analysis of the image features of plant diseases, traditional features with deep features are integrated to propose a comprehensive plant-disease feature classification algorithm, which involves the following components: (1) The OSTU algorithm based on the naive Bayes model is proposed to eliminate background interference and focus on the area where leaves are located. (2) From the perspective of traditional features, a multi-scale and multi-directional Gabor feature extraction model is proposed to obtain interpretable features. (3) Based on the advantages of MobileNet V2, spatial attention and channel attention mechanisms are proposed for plant-disease classification.

The remainder of this paper is organized as follows: Section 2 introduces the constructed database, including the multilevel feature extraction algorithm in Section 2.1 and the MobileNet algorithm based on dual attention in Section 2.2. Section 3 presents the experimental results and analysis, which verify the effectiveness of the proposed algorithm. Section 4 summarizes the innovations introduced in this paper and outlines potential avenues for future research.

2. Materials and Methods

The experimental data are sourced from the Plant Village public database provided by the University of Pennsylvania. It includes a total of 61 categories, classified by "species-disease-degree". The categories consist of 10 species, 27 diseases (24 of which are classified as general or severe), and 10 health classifications, as shown in Table 1. The dataset comprises 31,718 pictures in the training set and 4514 pictures in the test set, as shown in Figure 1. We can see that there are certain similarities within classes and certain differences between classes. It is crucial to choose effective features.

Table 1. Data classification.

Apple	Healthy		**Sreawberry**	Healthy	
	Scab	General		Scorch	General
		Serious			Serious
	Cedar Rust	General	**Tomato**	Bacterial Spot Bacteria	General
		Serious			Serious
Cherry	Healthy			Early Blight Fungus	General
	Powdery Mildew	General			Serious
		Serious		Late Blight Water Mold	General
Corn	Healthy				Serious
	Cercospora Zeaemaydis Techon and Daniels	General		Leaf Mold Fungus	General
		Serious			Serious
	Puccinia Polvsora	General		Target Spot Bacteria	General
		Serious			Serious
	Corn Curvularia Leaf Spot Fungus	General		Septoria Leaf Spot Fungus	General
		Serious			Serious
	Maize dwarf mosaic virus			Spider Mite Damage	General
Grape	Healthy				Serious
	Black Rot Fungus	General		YLCV Virus	General
		Serious			Serious
	Black Measles Fungus	General		Tomv	
		Serious	**Pepper**	Healthy	
	Leaf Blight Fungus	General		Scab	General
		Serious			Serious
Citrus	Healthy		**Potato**	Healthy	
	Greening June	General		Early Blight Fungus	General
		Serious			Serious
Peach	Healthy			Late Blight Fungus	General
	Bacterial Spot	General			Serious
		Serious	**Pepper**	Scab	General
Pepper	Healthy				Serious

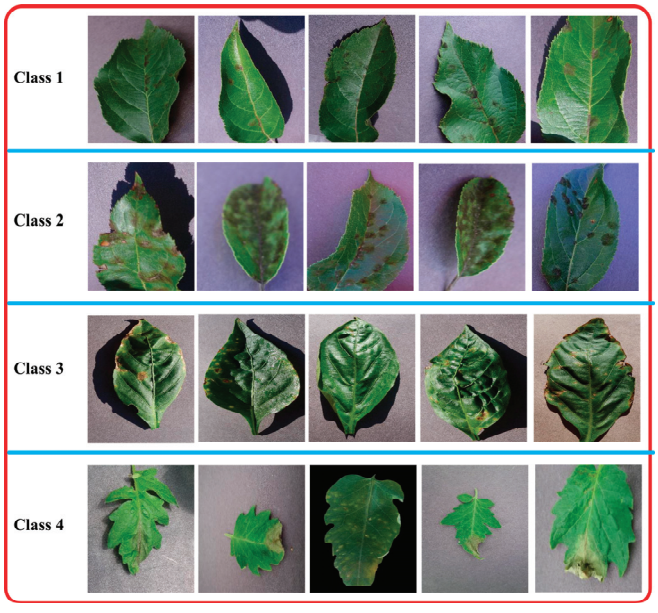


Figure 1. Experiment database.

The proposed plant disease classification algorithm based on an attention mechanism is shown in Figure 2. The algorithm comprises the following steps: (1) The OSTU algorithm based on the weighted Naive Bayes model is constructed to focus on the area where leaves are located and remove the influence of complex backgrounds. (2) Interpretable traditional features are adopted and extracted from multi-scale and multi-directional Gabor filters. (3) The extracted Gabor features are fed into a dual attention network for plant disease classification.

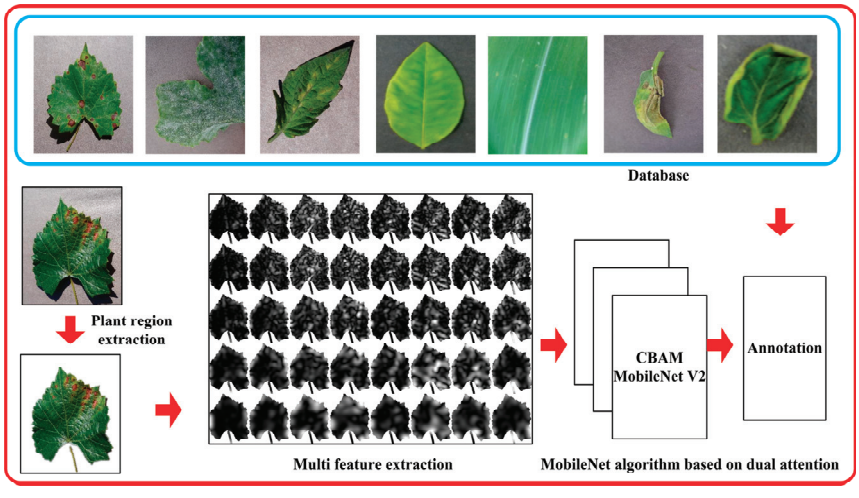


Figure 2. Algorithm flow chart.

2.1. Multilevel Feature Extraction Algorithm

The OTSU algorithm [34] achieves image segmentation through the calculation of the image gray features to determine the threshold. The principle is to maximize the inter-class

variance between the objects and the background of the image. It serves as an automatic optimization algorithm for image segmentation.

The total number of pixels in the image is notated as N, and the pixel gray level as G. f_i represents the number of pixels at gray level i. The threshold t is selected to divide the analysis into two categories: C_0 and C_1. Then, the corresponding probability of the two categories can be expressed as:

$$\begin{aligned} w_0 &= P(C_0) = \sum_{i=1}^{t} P_i = w(t) \qquad P_i = f_i/N \\ w_1 &= 1 - w(t) \end{aligned} \tag{1}$$

The inter-class variance of each pixel $\sigma_B^2(t)$ is selected as the evaluation index:

$$\sigma_B^2(t) = \frac{[\mu_r w(t) - \mu(t)]^2}{w(t)[1 - w(t)]} \tag{2}$$

$$\mu_r = \mu(L) = \sum_{i=1}^{L} iP_i, \qquad w(t) = \sum_{i=1}^{t} P_i \tag{3}$$

Then, the optimal threshold T is determined.

$$\sigma_B^2(T) = \max_{1 \le t \le l} \left\{ \sigma_B^2(t) \right\} \tag{4}$$

The traditional OTSU algorithm considers the neighborhood information. However, in cases where there is minimal distribution difference between background features and target pixels, the two peaks may not be clearly defined, resulting in a poor segmentation effect. To improve the traditional OTSU image processing method, the weighted Naive Bayes algorithm is introduced to refine the segmentation effect.

The naive Bayes algorithm is a commonly utilized data classification algorithm in machine learning algorithm research [35]. Thanks to its strong theoretical support, it boasts high classification efficiency and has been continuously studied and applied across different fields. Firstly, the Bayes principle is introduced:

$$P(c|X) = \frac{P(X|c)P(C)}{P(X)} \tag{5}$$

where $P(c \mid X)$ represents the posterior probability that X belongs to the category c. $P(c)$ and $P(X)$ denote the prior probabilities of category c, and conditional X. $P(X \mid c)$ represents the posterior probability that category c belongs to the condition X.

Suppose the dataset comprises m attribute variables denoted as A, and the category variables are $C = \{c_1, c_2, \ldots c_n\}$. The Naive Bayes model is obtained

$$c(x)_{NB} = \arg\max_i P(c_k) \prod_{i=1}^{m} P(x_i|c_k), \qquad 1 \le k \le n \tag{6}$$

where $P(c_k)$ is the prior probability when category c takes the value of k; $P(x_i \mid c_i)$ represents the posterior probability that category c_i belongs to the condition x_i.

The traditional naive Bayes model presupposes that different attributes are independent of one another, which is difficult to achieve in practice. In cases where there is a correlation between certain attributes, it greatly reduces the classification efficiency of the model, resulting in inaccurate experimental results. Therefore, attribute weighting is employed to retain the high classification accuracy of the traditional naive Bayes algorithm. The approach also alleviates the negative impact caused by the special condition of attribute

independence, which improves and enhances the efficiency of the traditional algorithm to a large extent. The corresponding formula is

$$c(x)_P = \arg\max_i P(c_k)\prod_{i=1}^{m} P(x_i|c_k)^{w(i)}, \quad 1 \le k \le n \tag{7}$$

where w_i represents the weight value of the class attribute A_i, exerting control over the segmentation effect. The key to the improved classification algorithm lies in the precise determination of the corresponding weight value of each attribute to yield superior results.

Through the above algorithms, the segmentation problem is transformed into a problem in probability theory calculations. The segmentation image is obtained by isolating the gray features in the image data and training with the Naive Bayes model. The corresponding algorithm process is illustrated in Figure 3.

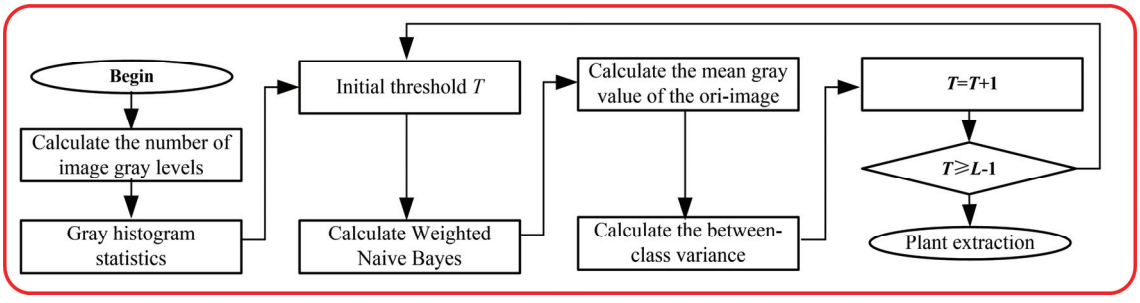


Figure 3. OSTU algorithm based on naive Bayes model.

The input image gray map is notated as G, with a total of N pixels, and the gray levels $i = 0, 1 \ldots L-1$. The corresponding gray histogram is represented as $H = \{h_0, h_1, \ldots h_{L-1}\}$. The threshold T is calculated to divide the gray image into G_b (background region) and G_f (foreground region). The corresponding probability distributions are

$$P(G_b) = \sum_{i=0}^{T} P(i) \tag{8}$$

$$P\left(G_f\right) = \sum_{i=T+1}^{L-1} P(i) \tag{9}$$

The average occurrence probability of each gray level is given by:

$$M(G_b) = \frac{\sum_{i=0}^{T} i \times w_i \times P(i)}{P(G_b)} \tag{10}$$

$$M\left(G_f\right) = \frac{\sum_{i=k+1}^{L-1} i \times w_i \times P(i)}{P\left(G_f\right)} \tag{11}$$

The threshold T to divide the original gray image into two categories is calculated by:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} \tag{12}$$

$$\begin{aligned}\sigma_B^2 &= P(G_b)(M(G_b) - M_G)^2 \\ &+P\Big(G_f\Big)\Big(M\Big(G_f\Big) - M_G\Big)^2\end{aligned} \tag{13}$$

$$\sigma_G^2 = \sum_{i=0}^{L-1}(i - M_G)^2 P(i) \tag{14}$$

The distance between classes of the two parts is calculated, and the distance is proportional to the segmentation effect. Therefore, the corresponding optimal threshold is:

$$T = \max_{0 \le i \le L-1} \sigma_B^2 \tag{15}$$

In order to analyze plant features, Gabor filtering [36] was introduced for further feature extraction. The Gabor filter, a kind of wavelet transform, exhibits excellent characteristics in the time and frequency domain. The Gabor function can be used to construct filters with different scales and directions. Since plant disease images are two-dimensional, research was conducted at the two-dimensional Gabor level. Its corresponding complex expression is as follows:

$$g(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right)\exp\left[i\left(2\pi\frac{X}{\lambda} + \varphi\right)\right] \tag{16}$$

$$\begin{aligned}X &= x\cos\theta + y\sin\theta \\ Y &= -x\sin\theta + y\cos\theta\end{aligned} \tag{17}$$

$$\begin{cases} g_{re} = \exp\left(-\frac{X^2+\gamma^2Y^2}{2\sigma^2}\right)\cos\left(2\pi\frac{X}{\lambda} + \varphi\right) \\ g_{im} = \exp\left(-\frac{X^2+\gamma^2Y^2}{2\sigma^2}\right)\sin\left(2\pi\frac{X}{\lambda} + \varphi\right) \end{cases} \tag{18}$$

where θ is the filter direction, λ is the filter wavelength, φ is the phase translation, γ is the spatial aspect ratio, σ is the standard deviation of the Gaussian factor, and b is the bandwidth. g_{re} represents the real part, and g_{im} represents the imaginary part.

$$\sigma = \frac{\lambda}{\pi}\sqrt{\frac{\ln 2}{2}}\left(\frac{2^b + 1}{2^b - 1}\right) \tag{19}$$

When the curve of the elliptic Gaussian envelope modulated by the complex sine wave of the Gabor function falls within the range of $(\mu - 3\sigma, \mu + 3\sigma)$, the area contained accounts for about 99.7% of the total area.

When designing a deep network, the receptive field must cover the entire relevant image region and be large enough to capture the context information for each pixel axis. Currently, the mainstream algorithms stack either large convolution kernels in shallow or small convolution kernels. However, increasing the receptive field will lead to a rise in training parameters and the computational cost.

If the standard convolution layer contains K $m \times m$ convolution kernels and c input features, then the corresponding number of parameter training is $(m \times m \times c + 1) \times K$. The proposed Gabor convolutional layer structure only requires updating of 4 parameters in each iteration, and the corresponding training parameter is $(4 \times c + 1) \times K$. Therefore, the Gabor convolution kernel is advantageous for the design of more compact networks.

In the Gabor convolution kernel, the parameters of each Gabor filter need to be optimized. The gradient descent algorithm is used to optimize filtering parameters through

back propagation according to the objective function. The reverse derivation process is as follows:

$$\begin{aligned}
\frac{\partial g_{re}}{\partial \lambda} &= \frac{\partial \pi X}{\lambda^2} g_{im} \\
\frac{\partial g_{re}}{\partial \theta} &= \frac{g_{re} XY}{\sigma^2}\left(\gamma^2 - 1\right) - \frac{2\pi Y}{\lambda} g_{im} \\
\frac{\partial g_{re}}{\partial \varphi} &= -g_{im} \\
\frac{\partial g_{re}}{\partial \sigma} &= \frac{X^2 + \gamma^2 Y^2}{\sigma^3} g_{re} \\
\frac{\partial g_{re}}{\partial \gamma} &= -\frac{\gamma^2 Y^2}{\sigma^2} g_{re}
\end{aligned} \tag{20}$$

To further enhance the feature map expression, Gabor filter weighting is adopted to generate Gabor filter with U directions and V scales. The direction is weighted by learning the weight vector W. The modulation process is as follows:

$$C_{i,u}^{v} = C[Wg(u, v)] \tag{21}$$

where u and v represent orientation and scale indexes. Since the Gabor filter contains multiple directions, the corresponding output feature updating process follows the back propagation mechanism:

$$\delta = \frac{\partial L}{\partial C_u^n} = \sum_{u=1}^{U} \frac{\partial L}{\partial C_u^n}[Wg(u, v)] \tag{22}$$

$$C_u^{n+1} = C_u^n - \eta\delta \tag{23}$$

where L is the loss function, η is the learning rate, and C_u^n is the result of the nth iteration. This makes the model more compact and robust to changes in direction and scale.

2.2. MobileNet Algorithm Based on Dual Attention

The main features of MobileNet V2 include: (1) the adoption of depth separable convolution in place of ordinary convolution to reduce model computation and parameter requirements; (2) the introduction of reverse residual structure to increase the number of network layers and enhance feature expressiveness; (3) replacement of nonlinear structures with linear Bottleneck structures to minimize the loss of low-dimensional feature information.

Based on the low power consumption characteristics of MobileNet V2 [37], MobileNet V2 has been selected as the main backbone network and improved through adjustments in width factor, attention module and multi-scale feature fusion. The network block diagram is shown in Figure 4.

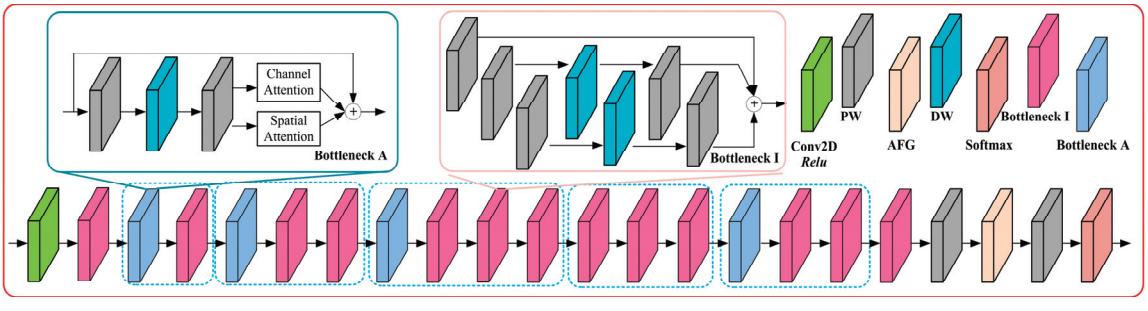


Figure 4. Attention-based MobileNet framework.

The depth-separable convolution replaces the standard convolution with fewer parameters and computation [38]. The computation ratio of the depth-separable convolution with the standard convolution is given by:

$$H = \frac{D_f^2 D_k^2 M + D_f^2 MN}{D_f^2 D_k^2 MN} = \frac{1}{N} + \frac{1}{D_k^2} \tag{24}$$

where D_f represents the length of the input feature map and the number of channels, D_k is the length of depthwise convolution (DW) convolution kernel, and M is the number of pointwise convolution channels.

In feature extraction, the usual convolution kernel size is 3×3. The computation amount and parameter quantity of depth-separable convolution are approximately $1/9$ of conventional convolution. MobileNet V2 incorporates the concept of ResNet and proposes the reverse residual structure. There are two typical methods: In the first method, PW convolution is used to increase the dimension, and DW convolution is used to extract the features from each channel, and PW convolution is then employed to reduce the feature dimensionality. When the step size is 1, a residual connection is established, while a series connection is established when the step size is 2; S The second method of the reverse residual structure first increases and then reduces dimensionality, which allows the network to accommodate smaller input and output dimensions, so as to reduce the computational load and parameters. At the same time, the residual connection can improve gradient propagation efficiency, with a deeper network layer.

An important parameter in the network is the width factor, which adjusts the number of convolution kernels in each module of the network to α times the original one, and the corresponding calculation load is:

$$T = \alpha D_f^2 D_k^2 M + \alpha^2 D_f^2 MN \tag{25}$$

By adjusting α, the computational burden of the model is greatly reduced.

CBAM is an attention mechanism module that integrates channel and space [39]. It is embedded in a convolutional neural network for end-to-end training. The final channel attention is illustrated in Figure 5a.

$$M_c(F) = \sigma\left(W_1 W_0 F_{avg} + W_1 W_0 F_{max}\right) \tag{26}$$

where F is the input feature map, σ is the nonlinear activation function, W_i is the weight of layer i, and F_{avg} and F_{max} are the results of input F after the average pooling and kernel maximum pooling, respectively.

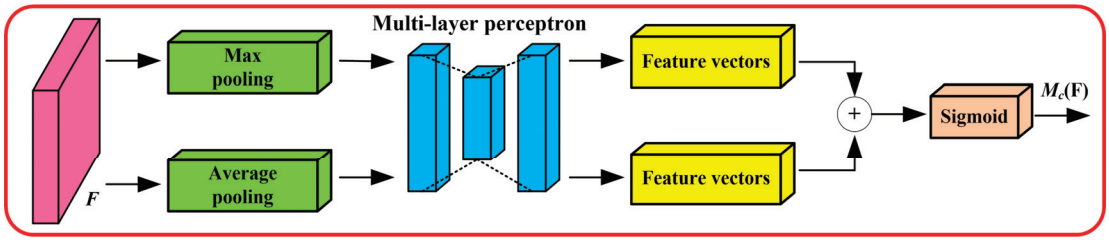
The spatial attention mechanism is shown in Figure 5b: the channel direction is averaged, and the kernel is maximally pooled to generate a feature description of size $2 \times H \times W$, and the feature vector is obtained and activated by a convolution operation. The corresponding spatial attention mapping model is as follows:

$$M_s(F) = \sigma\left(f\left(f_c(F_{avg} + F_{\max})\right)\right) \tag{27}$$

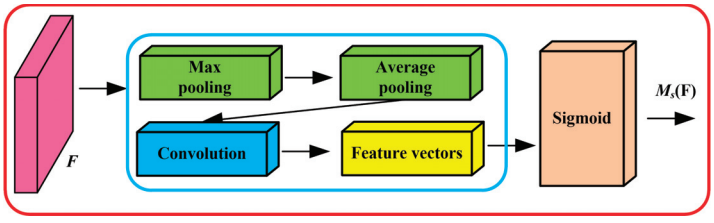
where f is the convolution operation, and f_c is the join operation. The complete calculation process of the CBAM module is

$$\begin{aligned} FA &= M_s(F')F' \\ F' &= M_c(F)F \end{aligned} \tag{28}$$

The CBAM module is integrated consecutively with the reverse residual block of MobileNet V2, which enables the module to focus on important features and suppress unnecessary ones in channel and spatial dimensions.



(**a**) Channel attention mechanism



(**b**) Spatial attention mechanism

Figure 5. Attention mechanism module.

Inception uses multiple convolution checks of varying sizes to extract features from feature maps, which increases the adaptability of the network to different scales. The structure of Inception V1 is shown in Figure 6, which enriches features at spatial scales and proves beneficial for subsequent classification.

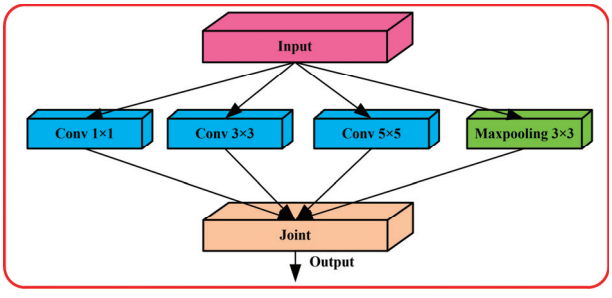


Figure 6. Inception V1 module.

Given that the MobileNet V2 network uses depth-separable convolution, in order to give full play to the advantages of MobileNet and Inception, the Inception module step is set to 2 to remove the linear structure of residual short-form, as shown in Figure 7. Feature extraction is carried out through three parallel branches. Considering that stitching increases channel count in the output feature map, along with the total network parameters, an addition-based merging approach is chosen to reduce the overall model parameters.

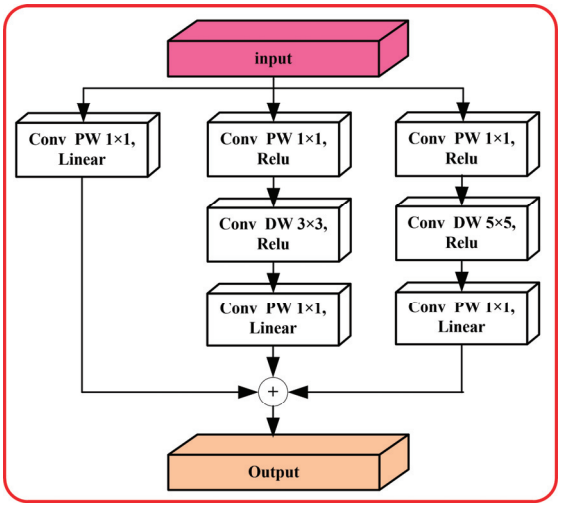


Figure 7. Scale feature fusion module.

3. Experimental Results and Analysis

3.1. Feature Extraction Algorithm

The result of leaf extraction is shown in Figure 8. To evaluate the algorithm performance, the following index [40] is introduced:

$$\begin{cases} AOM = \frac{R_s \cap R_g}{R_s \cup R_g} \\ AVM = \frac{R_s - R_g}{R_s} \\ AUM = \frac{R_g - R_s}{R_g} \\ CM = \frac{1}{3}\{AOM + (1 - AVM) + (1 - AUM)\} \end{cases} \tag{29}$$

where the area overlap measure (*AOM*), the area over segmentation measure (*AVM*), the area under segmentation measure (*AUM*), and the combination measure (*CM*) are used to evaluate the algorithm's performance. R_S represents the result of manual leaf labeling and serves as the gold standard. R_g denotes the result of the algorithmic labeling. Values of *AOM* and *CM* are proportional to the segmentation results, while *AVM* and *AUM* values are inversely proportional to the segmentation results.

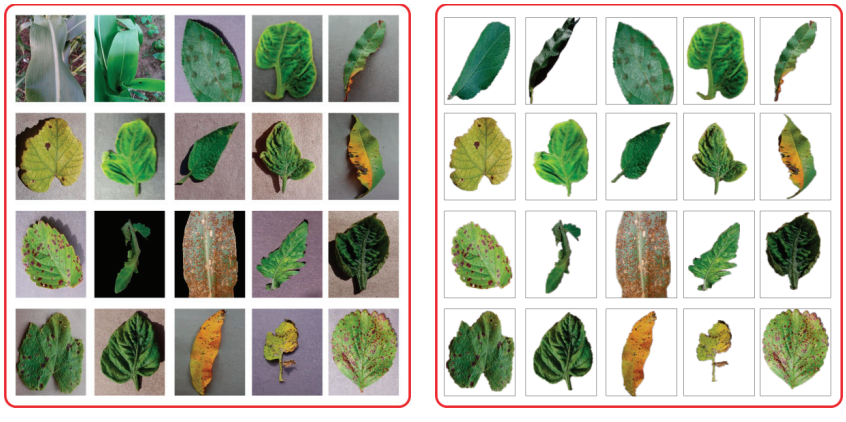
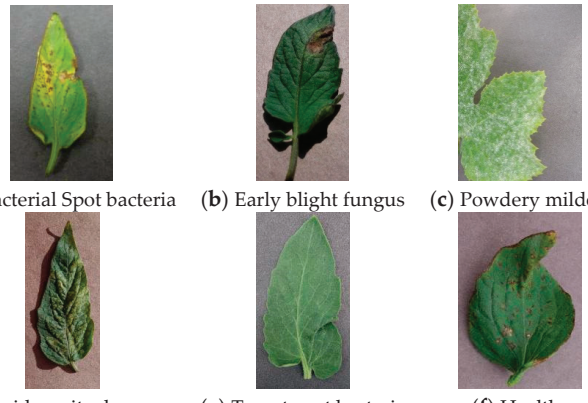


Figure 8. The effect of our object extraction.

Typical tomato leaf diseases, as shown in Figure 9, were selected for the study. The results of the algorithm comparison are summarized in Table 2. The algorithm with a fixed threshold requires human-computer interaction for threshold selection, which exhibits low adaptability and presents different colors in the presence of the scab, resulting in the inability to accurately segment tomato leaves with a single threshold. On the other hand, OSTU [34] achieves threshold segmentation by calculating the gap between classes and setting the threshold, which is self-adaptive. However, due to the uniqueness of the threshold setting, the segmentation performance is somewhat limited. The GSO [41] algorithm searches for local optimal clustering, which realizes target recognition even under complex backgrounds, but it has some limitations in considering inter-class differences. The proposed algorithm fuses the OSTU algorithm with the attribute-weighted Bayes algorithm. This hybrid approach considers inter-class differences and intra-class similarities at a local level and exhibits a favorable effect on shadow suppression.



(**a**) Bacterial Spot bacteria (**b**) Early blight fungus (**c**) Powdery mildew

(**d**) Spider mite damage (**e**) Target spot bacteria (**f**) Healthy

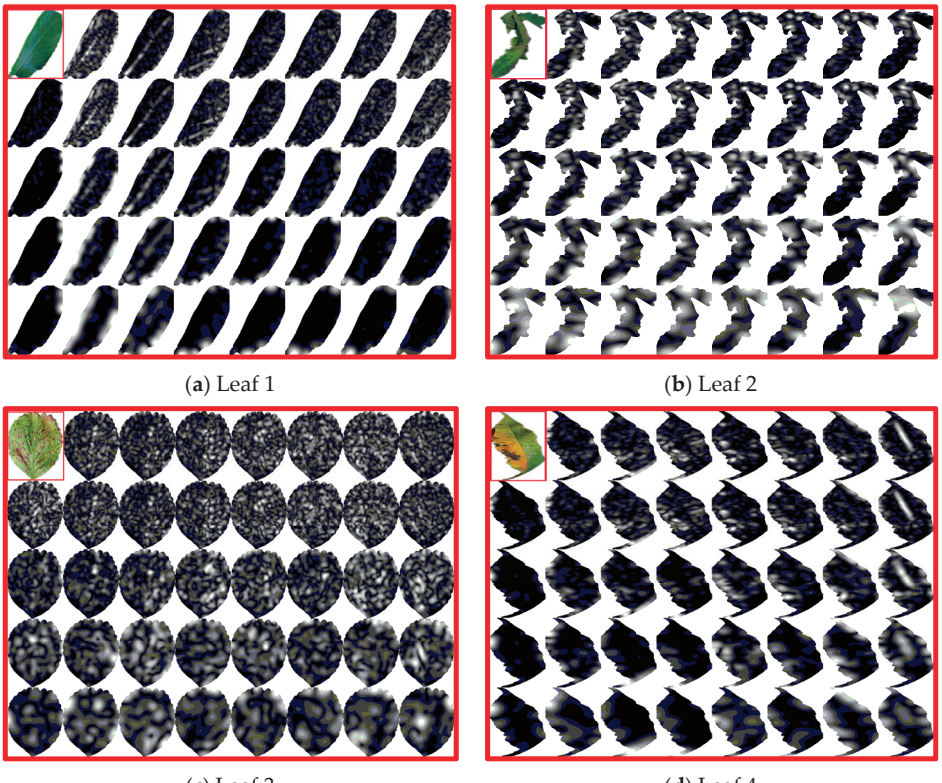
Figure 9. Typical tomato diseases.

To show the effect of the algorithm, an analysis is conducted on the effect of Gabor filtering. As shown in Figure 10, a variety of conditions of leaves are observed: Figure 10a shows a smooth leaf surface; Figure 10b displays leaves damaged as a whole; Figure 10c illustrates leaves with a large number of spots on the surface; and Figure 10d exhibits leaves with yellowing surfaces. After Gabor filtering, Leaf 1 demonstrates rich texture features at low dimensions and angles, appearing relatively rich and smooth as dimensions and angles increase. Leaf 2 exhibits a robust response in high dimensions and angles, with high pixel values in the displayed image. Leaf 3 presents more textures after Gabor filtering. Overall, the response of Leaf 4 is not strong, but the lesion area exhibits a robust response in high dimensions and angles. Through the above analysis, the Gabor-filtered image is input into the depth learning module for disease classification.

To reflect the effect of feature extraction through Gabor filtering, the study takes the original image and Gabor-filtered image as input, uses a deep network for training, and compares their convergence performance, as shown in Figure 11. The results clearly demonstrate that the features extracted by Gabor are more representative, with the fast convergence speed of the algorithm, which brings it closer to the target function. This is because Gabor extracts multi-angle features, analyzes image characteristics, focuses on targets, and achieves efficient representation.

Table 2. (1) The algorithm extracts the performance of tomato leaves with bacterial spot bacteria. (2) The algorithm extracts the performance of tomato leaves with early blight fungus. (3) The algorithm extracts the performance of tomato leaves with powdery mildew. (4) The algorithm extracts the performance of tomato leaves with spider mite damage. (5) The algorithm extracts the performance of tomato leaves with target spot bacteria. (6) The algorithm extracts the performance of healthy tomato leaves.

		(1)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.71	0.41	0.34	0.65
OSTU [34]	0.76	0.35	0.33	0.70
GSO [41]	0.82	0.34	0.31	0.72
Ours	0.85	0.31	0.29	0.75
		(2)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.74	0.33	0.35	0.69
OSTU [34]	0.81	0.31	0.32	0.73
GSO [41]	0.85	0.27	0.29	0.76
Ours	0.87	0.26	0.27	0.78
		(3)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.75	0.31	0.33	0.70
OSTU [34]	0.78	0.27	0.31	0.73
GSO [41]	0.84	0.24	0.28	0.77
Ours	0.88	0.23	0.24	0.80
		(4)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.74	0.34	0.27	0.71
OSTU [34]	0.81	0.35	0.25	0.74
GSO [41]	0.86	0.24	0.22	0.8
Ours	0.91	0.21	0.19	0.84
		(5)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.79	0.31	0.25	0.74
OSTU [34]	0.87	0.28	0.23	0.79
GSO [41]	0.91	0.23	0.19	0.83
Ours	0.93	0.18	0.17	0.86
		(6)		
Algorithm	***AOM***	***AVM***	***AUM***	***CM***
T	0.86	0.26	0.23	0.79
OSTU [34]	0.89	0.23	0.22	0.81
GSO [41]	0.92	0.17	0.18	0.86
Ours	0.95	0.15	0.16	0.88



(**a**) Leaf 1 (**b**) Leaf 2

(**c**) Leaf 3 (**d**) Leaf 4

Figure 10. The effect of Gabor filtering.

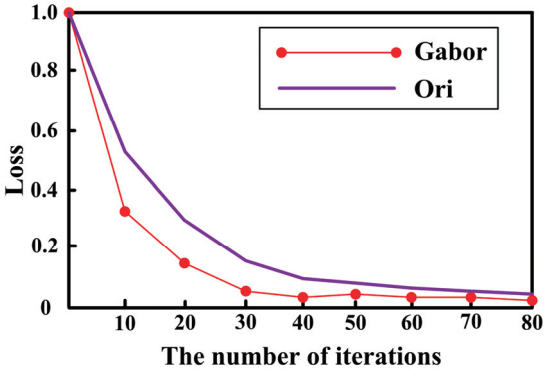


Figure 11. Iteration curve.

3.2. Comparison of Classification Algorithms

The following index are introduced for performance evaluation,

$$\begin{cases} SEN = \frac{TP}{TP+FN} \\ SPE = \frac{TN}{TN+FP} \\ ACC = \frac{TP+TN}{TP+FP+TN+FN} \\ FPF = 1 - ACC \end{cases} \quad (30)$$

where *SEN* reflects the detection performance for real objects. *SPE* reflects the detection performance for false objects. *ACC* reflects the ratio of correct test results to all samples in the test results, and *FPF* reflects the ratio of false test results diagnosed as true objects.

The comparison results of algorithms are shown in Tables 3 and 4. It can be seen that the algorithm's results without leaf area extraction are inferior to those with leaf area extraction, indicating that focusing on the area where leaves are located and reducing the influence of surrounding environments yields a positive effect. This verifies the good effect of the proposed leaf segmentation algorithm on subsequent disease clustering. The algorithm based on texture features proposed in Ref. [2] exhibits disease detection capabilities, albeit with slightly lower performance in terms of indicators. Ref. [23] conducts an in-depth analysis of the image characteristics and constructs a K-means model to improve the classification accuracy. Ref. [8] adds ANN based on an SVM classifier to introduce the classification task into multi-dimensional space, realize classification, and achieve certain results. Based on the current mainstream RESNET, Ref. [30] requires 50 iterations to achieve disease detection, but too many network parameters cause large consumption of computing resources. Ref. [37] constructs the traditional MobileNet V2 algorithm and refines the convolution calculation method to effectively boost the calculation speed and performance. Although SEN reaches an impressive 91%, this process is inconsistent with the human cognitive process. On the basis of MobileNet V2, the proposed algorithm adds an attention mechanism module in line with the process of human visual perception. The modification induces a further enhancement in algorithm performance, with superior results achieved.

Table 3. Algorithm results without leaf region extraction.

Algorithm	*SEN*	*SPE*	*ACC*	*FPF*
Texture [2]	0.65	0.31	0.72	0.28
K-means [32]	0.72	0.29	0.76	0.24
SVM + ANN [8]	0.75	0.25	0.80	0.20
ResNet [30]	0.81	0.20	0.82	0.18
MobileNet V2 [37]	0.84	0.18	0.84	0.16
ICNN [42]	0.85	0.15	0.86	0.14
GAN [43]	0.86	0.12	0.89	0.11
Ours	0.86	0.10	0.91	0.09

Table 4. Algorithm results with leaf region extraction.

Algorithm	*SEN*	*SPE*	*ACC*	*FPF*
Texture [2]	0.71	0.26	0.79	0.21
K-means [32]	0.79	0.24	0.85	0.15
SVM + ANN [8]	0.81	0.21	0.87	0.13
ResNet [30]	0.85	0.15	0.91	0.09
MobileNet V2 [37]	0.91	0.13	0.96	0.04
ICNN [42]	0.91	0.11	0.97	0.03
GAN [43]	0.92	0.10	0.96	0.04
Ours	0.94	0.08	0.98	0.02

Table 3 illustrates the limited effect of the traditional algorithm [2,8,23], but the algorithm based on depth network exhibits commendable performance. Therefore, deep learning algorithms [30,37,42,43] are compared, as shown in Figure 12. The ResNet algorithm [30] involves many parameters, resulting in slower convergence in the training process. In contrast, the MobileNet V2 algorithm [37] introduces a fast convolution change to reduce parameters, along with faster convergence. Improved CNN (ICNN) [42] based on color features focuses on the area where leaves are located and achieves satisfactory results. Generative adversarial networks (GAN) [43] further improve the effect by deeply mining intra-class and inter-class features in small sample situations.

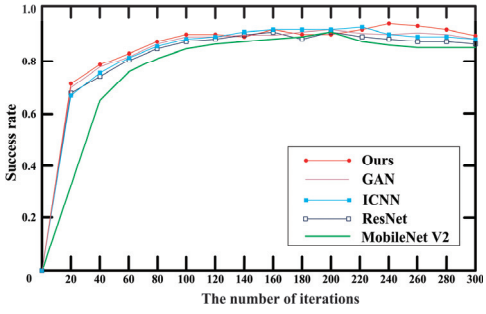


Figure 12. Algorithm success rate curve.

On the basis of MobileNet V2, the proposed algorithm integrates the Inception module, which not only ensures swift convergence but also introduces visual attention, which is in line with the principle of human visual perception, alongside further improved performance.

Classified images of representative tomato diseases are selected: scab, early blight, powdery mildew, starred spider, spotted disease, and health to verify the performance of different algorithms, as shown in Figure 13. ResNet extracts local and global features for accurate image classification. MobileNet V2 demonstrates a certain classification performance in low-parametric network construction. VGG, through multi-layer filter convolution, can fully explore image features and realize classification. It can be seen from the figure that for single disease recognition, the accuracy of scab and early blight identification is low due to their striking similarity in spot shape, size and texture. As a result, misclassifications are more likely to occur in these cases. On the contrary, Health exhibits the highest accuracy. Based on MobileNet V2, the proposed algorithm adds the attention mechanism module to extract features across different scales, with superior results compared to other algorithms.

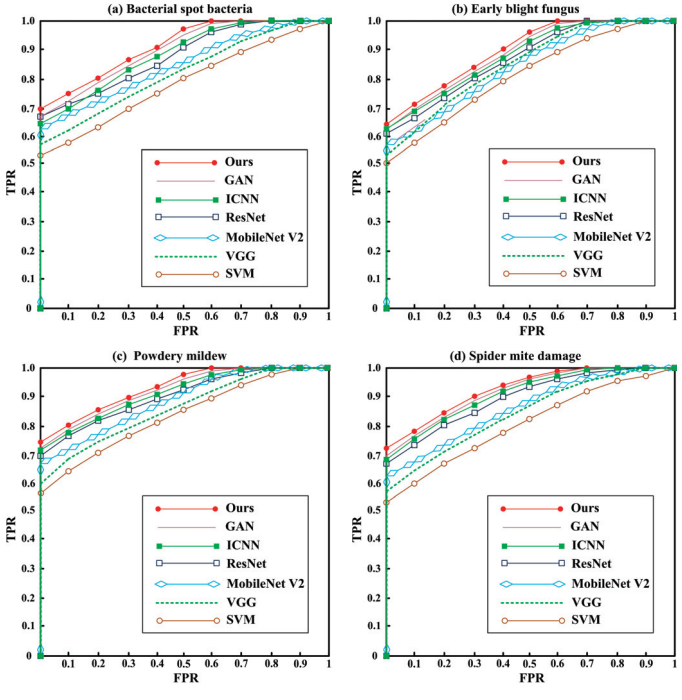


Figure 13. *Cont.*

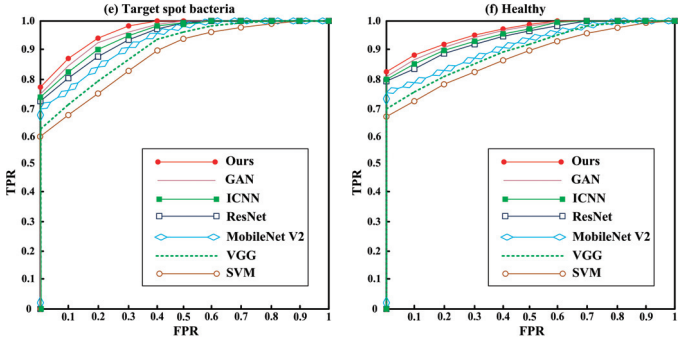


Figure 13. ROC curve.

4. Conclusions

Plant diseases impose a serious impact on plant growth, making it of great significance to identify diseases through artificial intelligence. Since leaves are the direct manifestation of plant diseases, the research focuses on leaf features. To maximize the potential of traditional features and deep features, a comprehensive Plant Disease classification algorithm is proposed. (1) To solve the difficult classification of leaf diseases in complex backgrounds, the OSTU algorithm based on the Naive Bayes model is proposed to focus on the area where leaves are located and reduce background interference. (2) From the perspective of feature interpretability, a multi-dimensional feature model based on traditional features is constructed to fully explore leaf features. (3) From the perspective of deep learning, a MobileNet framework based on dual attention is established to achieve swift disease recognition. The algorithm underwent rigorous testing on the Plant Village open database, and the results showed that the algorithm could achieve plant disease classification.

Despite these achievements, there are also some problems in the research: The experimental dataset is limited and does not cover most diseases. Therefore, a larger dataset will be constructed to further integrate traditional features and deep features. Further studies will be conducted on interpretable fusion networks to promote research on plant disease prediction.

Author Contributions: Conceptualization, H.W. and S.Q.; methodology, H.Y. and X.L.; software, H.W. and S.Q.; validation, H.W.; formal analysis, All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Light of West China (No. XAB2022YN10) and Shaanxi key research and development plan (No. 2018ZDXM-SF-093) and Shaanxi Province key industrial innovation chain (No. S2022-YF-ZDCXL-ZDLGY-0093 and 2023-ZDLGY-45).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Al-Hiary, H.; Bani-Ahmad, S.; Reyalat, M.; Braik, M.; Alrahamneh, Z. Fast and accurate detection and classification of plant diseases. *Int. J. Comput. Appl.* **2011**, *17*, 31–38. [CrossRef]
2. Kulkarni, A.H.; Patil, A. Applying image processing technique to detect plant diseases. *Int. J. Mod. Eng. Res.* **2012**, *2*, 3661–3664.
3. Arivazhagan, S.; Shebiah, R.N.; Ananthi, S.; Varthini, S.V. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agric. Eng. Int. CIGR J.* **2013**, *15*, 211–217.
4. Hossain, E.; Hossain, M.F.; Rahaman, M.A. A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–6.
5. Singh, V.; Misra, A.K. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **2017**, *4*, 41–49. [CrossRef]

6. Kaur, R.; Singla, S. Classification of plant leaf diseases using gradient and texture feature. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing, Thai Nguyen, Vietnam, 12–13 December 2016; pp. 1–7.
7. Nanehkaran, Y.A.; Zhang, D.; Chen, J.; Tian, Y.; Al-Nabhan, N. Recognition of plant leaf diseases based on computer vision. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1–18. [CrossRef]
8. Pujari, D.; Yakkundimath, R.; Byadgi, A.S. SVM and ANN based classification of plant diseases using feature reduction technique. *IJIMAI* **2016**, *3*, 6–14. [CrossRef]
9. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: Detection and saliency map visualization. In *Human and Machine Learning*; Springer: Cham, Switzerland, 2018; pp. 93–117.
10. Mahmoud MA, B.; Guo, P.; Wang, K. Pseudoinverse learning autoencoder with DCGAN for plant diseases classification. *Multimed. Tools Appl.* **2020**, *79*, 26245–26263. [CrossRef]
11. Sandesh Kumar, C.; Sharma, V.K.; Yadav, A.K.; Singh, A. Perception of plant diseases in color images through adaboost. In *Innovations in Computational Intelligence and Computer Vision*; Springer: Singapore, 2021; pp. 506–511.
12. Hang, J.; Zhang, D.; Chen, P.; Zhang, J.; Wang, B. Classification of plant leaf diseases based on improved convolutional neural network. *Sensors* **2019**, *19*, 4161. [CrossRef]
13. Atila, Ü.; Uçar, M.; Akyol, K.; Uçar, E. Plant leaf disease classification using EfficientNet deep learning model. *Ecol. Inform.* **2021**, *61*, 101182. [CrossRef]
14. Sardogan, M.; Tuncer, A.; Ozen, Y. Plant leaf disease detection and classification based on CNN with LVQ algorithm. In Proceedings of the 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 20–23 September 2018; pp. 382–385.
15. Deepa, N.R.; Nagarajan, N. Kuan noise filter with Hough transformation based reweighted linear program boost classification for plant leaf disease detection. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 5979–5992. [CrossRef]
16. Altan, G. Performance evaluation of capsule networks for classification of plant leaf diseases. *Int. J. Appl. Math. Electron. Comput.* **2020**, *8*, 57–63. [CrossRef]
17. Pal, A.; Kumar, V. AgriDet: Plant leaf disease severity classification using agriculture detection framework. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105754. [CrossRef]
18. Liang, Q.; Xiang, S.; Hu, Y.; Coppola, G.; Zhang, D.; Sun, W. PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Comput. Electron. Agric.* **2019**, *157*, 518–529. [CrossRef]
19. Yu, H.; Liu, J.; Chen, C.; Heidari, A.A.; Zhang, Q.; Chen, H.; Mafarja, M.; Turabieh, H. Corn leaf diseases diagnosis based on K-means clustering and deep learning. *IEEE Access* **2021**, *9*, 143824–143835. [CrossRef]
20. Padol, P.B.; Yadav, A.A. SVM classifier based grape leaf disease detection. In Proceedings of the 2016 Conference on Advances in Signal Processing (CASP), Pune, India, 9–11 June 2016; pp. 175–179.
21. Rani, F.P.; Kumar, S.N.; Fred, A.L.; Dyson, C.; Suresh, V.; Jeba, P.S. K-means clustering and SVM for plant leaf disease detection and classification. In Proceedings of the 2019 International Conference on Recent Advances in Energy-Efficient Computing and Communication (ICRAECC), Nagercoil, India, 7–20 March 2019; pp. 1–4.
22. Trivedi, V.K.; Shukla, P.K.; Pandey, A. Automatic segmentation of plant leaves disease using min-max hue histogram and k-mean clustering. *Multimed. Tools Appl.* **2022**, *81*, 20201–20228. [CrossRef]
23. Faithpraise, F.; Birch, P.; Young, R.; Obu, J.; Faithpraise, B.; Chatwin, C. Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. *Int. J. Adv. Biotechnol. Res.* **2013**, *4*, 189–199.
24. Tamilselvi, P.; Kumar, K.A. Unsupervised machine learning for clustering the infected leaves based on the leaf-colours. In Proceedings of the 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai, India, 23–24 March 2017; pp. 106–110.
25. Hasan, R.I.; Yusuf, S.M.; Mohd Rahim, M.S.; Alzubaidi, L. Automatic clustering and classification of coffee leaf diseases based on an extended kernel density estimation approach. *Plants* **2023**, *12*, 1603. [CrossRef]
26. Yadhav, S.Y.; Senthilkumar, T.; Jayanthy, S.; Kovilpillai, J.J.A. Plant disease detection and classification using cnn model with optimized activation function. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 564–569.
27. Bhimavarapu, U. Prediction and classification of rice leaves using the improved PSO clustering and improved CNN. *Multimed. Tools Appl.* **2023**, *82*, 21701–21714. [CrossRef]
28. Hatuwal, B.K.; Shakya, A.; Joshi, B. Plant leaf disease recognition using random Forest, KNN, SVM and CNN. *Polibits* **2020**, *62*, 13–19.
29. Pareek, P.K.; Ramya, I.M.; Jagadeesh, B.N.; LeenaShruthi, H.M. Clustering based segmentation with 1D-CNN model for grape fruit disease detection. In Proceedings of the 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 24–25 February 2023; pp. 1–7.
30. Mukti, I.Z.; Biswas, D. Transfer learning based plant diseases detection using ResNet50. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 20–22 December 2019; pp. 1–6.
31. Li, M.; Cheng, S.; Cui, J.; Li, C.; Li, Z.; Zhou, C.; Lv, C. High-performance plant pest and disease detection based on model ensemble with inception module and cluster algorithm. *Plants* **2023**, *12*, 200. [CrossRef]

32. Muammer, T.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1636–1651.
33. Ramesh, S.; Hebbar, R.; Niveditha, M.; Pooja, R.; Shashank, N.; Vinod, P.V. Plant disease detection using machine learning. In Proceedings of the 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 25–28 April 2018; pp. 41–45.
34. Hoang, N.D. Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding. *Adv. Civ. Eng.* **2018**, *2018*, 3924120. [CrossRef]
35. Vembandasamy, K.; Sasipriya, R.; Deepa, E. Heart diseases detection using Naive Bayes algorithm. *Int. J. Innov. Sci. Eng. Technol.* **2015**, *2*, 441–444.
36. Yuan, Y.; Wang, L.N.; Zhong, G.; Gao, W.; Jiao, W.; Dong, J.; Shen, B.; Xia, D.; Xiang, W. Adaptive Gabor convolutional networks. *Pattern Recognit.* **2022**, *124*, 108495. [CrossRef]
37. Srinivasu, P.N.; SivaSai, J.G.; Ijaz, M.F.; Bhoi, A.K.; Kim, W.; Kang, J.J. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors* **2021**, *21*, 2852. [CrossRef]
38. Dang, L.; Pang, P.; Lee, J. Depth-wise separable convolution neural network with residual connection for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 3408. [CrossRef]
39. Fu, H.; Song, G.; Wang, Y. Improved YOLOv4 marine target detection combined with CBAM. *Symmetry* **2021**, *13*, 623. [CrossRef]
40. Qiu, S.; Jin, Y.; Feng, S.; Zhou, T.; Li, Y. Dwarfism computer-aided diagnosis algorithm based on multimodal pyradiomics. *Inf. Fusion* **2022**, *80*, 137–145. [CrossRef]
41. Selvanambi, R.; Natarajan, J.; Karuppiah, M.; Islam, S.H.; Hassan, M.M.; Fortino, G. Lung cancer prediction using higher-order recurrent neural network based on glowworm swarm optimization. *Neural Comput. Appl.* **2020**, *32*, 4373–4386. [CrossRef]
42. Kaya, Y.; Gürsoy, E. A novel multi-head CNN design to identify plant diseases using the fusion of RGB images. *Ecol. Inform.* **2023**, *75*, 101998. [CrossRef]
43. Lamba, S.; Saini, P.; Kaur, J.; Kukreja, V. Optimized classification model for plant diseases using generative adversarial networks. *Innov. Syst. Softw. Eng.* **2023**, *19*, 103–115. [CrossRef]

algorithms

MDPI

Article

Predicting the Gap in the Day-Ahead and Real-Time Market Prices Leveraging Exogenous Weather Data

Nika Nizharadze [1,†], Arash Farokhi Soofi [1,2,†] and Saeed Manshadi [1,*]

[1] Department of Electrical and Computer Engineering, Diego State University, San Diego, CA 92182, USA; nnizharadze@sdsu.edu (N.N.); afarokhisoofi7161@sdsu.edu (A.F.S.)
[2] Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA
* Correspondence: smanshadi@sdsu.edu
† These authors contributed equally to this work.

Abstract: Predicting the price gap between the day-ahead Market (DAM) and the real-time Market (RTM) plays a vital role in the convergence bidding mechanism of Independent System Operators (ISOs) in wholesale electricity markets. This paper presents a model to predict the values of the price gap between the DAM and RTM using statistical machine learning algorithms and deep neural networks. In this paper, we seek to answer these questions: What will be the impact of predicting the DAM and RTM price gap directly on the prediction performance of learning methods? How can exogenous weather data affect the price gap prediction? In this paper, several exogenous features are collected, and the impacts of these features are examined to capture the best relations between the features and the target variable. An ensemble learning algorithm, namely the Random Forest (RF), is used to select the most important features. A Long Short-Term Memory (LSTM) network is used to capture long-term dependencies in predicting direct gap values between the markets stated. Moreover, the advantages of directly predicting the gap price rather than subtracting the price predictions of the DAM and RTM are shown. The presented results are based on the California Independent System Operator (CAISO)'s electricity market data for two years. The results show that direct gap prediction using exogenous weather features decreases the error of learning methods by 46%. Therefore, the presented method mitigates the prediction error of the price gap between the DAM and RTM. Thus, the convergence bidders can increase their profit, and the ISOs can tune their mechanism accordingly.

Keywords: electricity market; real-time market; day-ahead market; locational marginal pricing; long short-term memory (LSTM); multivariate time series forecasting

Citation: Nizharadze, N.; Farokhi Soofi, A.; Manshadi, S. Predicting the Gap in the Day-Ahead and Real-Time Market Prices Leveraging Exogenous Weather Data. *Algorithms* **2023**, *16*, 508. https://doi.org/10.3390/a16110508

Academic Editors: Gloria Cerasela Crisan, Ha Duy Long and Elena Nechita

Received: 19 September 2023
Revised: 27 October 2023
Accepted: 29 October 2023
Published: 4 November 2023

1. Introduction

One major concern in the design of a two-settlement electricity market is the gap in the clearing prices across the DAM and RTM. The DAM is a financial market that schedules the supply and demand before the operating day, while the RTM is a physical market that settles based on the served demand and provided supply. Based on the concept of locational marginal pricing (LMP) [1], the ISOs determine DAM and RTM prices daily using generation units' offers and locational demands. The difference between the locational marginal pricing (LMP) values of the DAM and RTM is an indicator of the surplus or shortage of electricity in the electric grid compared to the predicted values. Wholesale electricity market prices are volatile due to fuel-cost alterations; weather-sensitive generation units, such as solar generation [2] and Distributed Energy Resources (DERs) [3]; weather-related demands [4]; and planned and forced outages [5]. Multiple settlements create more stable prices and lessen the RTM's vulnerability to price surges [6]. It is shown in [7] that with two-settlement electricity markets, generation units have incentives to enter into real-time contracts, which will reduce real-time electricity prices, which in turn will increase social

welfare. Consequently, all market participants will benefit from such a settlement. Nevertheless, there will be a gap between the day-ahead and real-time settlement. The increase in the penetration level of renewable energy resources exacerbated the volatility of energy supply and prices within the RTM [8]. Thus, predicting the price gap between the DAM and RTM has become more complicated. Predicting such a gap plays an integral role in establishing the operating schedules and adjusting the bidding strategies of the market participants, i.e., physical and virtual market participants within the market [9]. This is particularly important for convergence bidders who are interested in a hedge against the price gap across the two markets [10,11]. The day-ahead market (DAM) is a financial market where participants submit their bids for the following 24 h, whereas the real-time market (RTM) is a physical market in which buyers and sellers submit their bids during the day, usually for a 5-min interval. The RTM balances out the differences between DAM purchases and the actual real-time demand and generation of electricity. In this paper, the focus is on predicting the gap between the cleared prices within the two markets. The gap value can provide significant insights about the supply and demand of the electricity market, which could be valuable information for ISOs, market designers, and physical and virtual market participants to help them enhance the efficiency of the market and reduce their risks.

The prediction of day-ahead hourly electricity prices by leveraging an integrated machine learning model is proposed in [12]. In this article, the authors employed Bayesian clustering by dynamics to cluster the data set into several subsets, and Support Vector Machines (SVM) were used to fit the training data into each subset. The error metrics of the integrated model are significantly improved compared to that of the single SVM network. In [13], the authors proposed Auto-Regressive Integrated Moving Average (ARIMA) models to predict next-day prices for Spanish and Californian markets. In [14], a Random Forest regression is leveraged to predict DAM prices. The proposed approach outperformed the ARIMA model. However, this paper does not consider the impacts of exogenous features such as temperature and solar irradiance to predict prices. Moreover, the price gap between the DAM and RTM is not predicted in the literature.

Even though statistical models perform well at identifying patterns and indicators that will influence the price of electricity, they struggle to predict prices accurately in the presence of spikes, which is particularly important for predicting the gap price across a two-settlement market [15]. The electricity market will be cleared based on the net demand, which in turn depends on many characteristics such as weather, temperature, wind speed, and precipitation. Thus, the LMP tends to fluctuate over an operational horizon. In [16], the authors compared the ARIMA model with Artificial Neural Networks (ANNs) to forecast an electricity price. To handle the complexity of the electricity market, ANNs are used in [17]. The increase in the number of computation layers increases the feature abstraction capability of the networks, which makes them better at identifying non-linear trends [18]. An ensemble of CNN-LSTM and an ARMA model is utilized for financial time series data in [19]. In [20], a Temporal Convolutional Neural Network (TCNN) model is utilized for the analysis of financial time series data, specifically focusing on applications in Forex markets. This approach is contrasted with Recurrent Neural Networks and other deep learning models, as well as some of the top-performing Machine Learning methods, to demonstrate its effectiveness in handling financial data. The ARMA model captures the linear dependencies of features and target variables, while CNN-LSTM models the nonlinear spatial connections in data features between adjacent time intervals and also accounts for long-term time-based patterns in the data. The ensemble of CNN-LSTM and the ARMA model achieved a 0.8837 MAE score for The European Union Emission Trading System (EU ETS) dataset. In [21], the LSTM network and a variation of the deep Recurrent Neural Network (RNN) are used to forecast electricity load, and the outputs of the models are compared to those of statistical models. The electricity consumption of the past 10 days is used to predict the electricity consumption of the next day. The LSTM-based network significantly outperformed the Seasonal-ARIMA and Support Vector Regression (SVR)

models. A similar model is used in [22] to predict electricity load, but in this case, in addition to the historical load data, weather datasets are also utilized. However, no change in the model performance is observed when weather-related features are removed and only the time lags are used as inputs. In [23], the LSTM network is used to predict the next 24 h of electricity prices for Australian and Singaporean markets. The mean absolute percentage error (MAPE) was used to evaluate the model, and up to a 47.3% improvement was observed compared to a multi-layer ANN.

According to [24], the prediction of real-time LMP is even more challenging, and most of the approaches adopted from previous studies generate an MAPE of around 10–20%. In [25], a homogeneous Markov chain representation of the RTM LMP is used to predict the RTM LMP for the prediction horizon of 6–8 h. Future prices are computed based on state transition matrices using the Monte Carlo method. Although the mean average error (MAE) metric of the model was 11.75 USD/MWh, it has a huge computation burden. In [26], the authors proposed a deep LSTM (D-LSTM) network to estimate short- and medium-term demand as well as the LMP. The D-LSTM network turned out to have a flat trend without the validation set. However, once the network was tuned, it outperformed the nonlinear auto-regressive network with exogenous variables (NARX) and Extreme Learning Machine (ELM) models in terms of accuracy. In [27], the researchers used the Generative Adversarial Network (GAN)-based video prediction approach on market data from ISO NE to predict RTM LMPs. The market data images are created from the historical data, and by the concatenation of these images, a video stream is created. Consequently, the prediction of the next frame is used to predict the next-hour RTM LMP. The proposed method achieved approximately an 11% MAPE score. However, weather data sets are not utilized to enhance the prediction model of the price spikes. The enhanced convolutional neural networks are also used in [28] to predict electricity load and prices. Here, feature selection is carried out using the Random Forest model, and the extracted features are passed to the convolutional layer, which later is filtered using the max pooling layer. The showcased work resulted in smaller error measurements than the SVR using NYISO market data.

Leveraging the LSTM network to predict the gap between RTM and DAM prices using weather features brings the following question to mind: Can we improve the prediction of the price gap across the DAM and RTM by leveraging exogenous information (e.g., weather data, including solar irradiance)?

The contributions of this paper are summarized as follows:

1. Syncing the exogenous information on weather data with the electricity market information, i.e., prices and demand data, to create an extensive dataset. The significance of leveraging the external dataset is illustrated, and the importance of features is also demonstrated.
2. Both the DAM and RTM are analyzed for price prediction. A realistic set of assumptions is made regarding the availability of features for both the RTM and DAM once the prices are predicted 24–36 h in advance for the following market operation day upon the clearing of the market.
3. The ensemble learning method, namely the Random Forest (RF), is used to calculate the probability distribution of the predicted market prices for the DAM and RTM, as well as the gap.
4. An LSTM architecture is deployed to enhance predictions given the complexity of predicting values for the time series dataset. The proposed model is compared with other statistical machine learning methods, which demonstrate significant improvements.

The rest of the paper is organized as follows. The learning methods used and the LSTM network are discussed in Section 2: Learning Methods. The metrics deployed to evaluate the performance of learning methods are presented in Section 3. The data collection and data cleansing procedures are detailed in Section 4: Data. The performance of the proposed method is evaluated in Section 5. The paper is concluded in Section 6.

2. Learning Algorithms and Methodologies

In this section, the methods that are leveraged to examine the direct price gap values are introduced. The described learning algorithms are utilized to predict price gaps between the RTM and DAM as well as to rank features based on their importance. In addition, these methods are leveraged to construct probability distributions for the DAM and RTM price predictions.

2.1. Least Absolute Shrinkage and Selection Operator (LASSO)

The objective of the linear regression model is to find a relationship between two variables by fitting a linear equation to observed data points. The most common way to find a fitted line is to use the least-squares method, in which the model finds a fitted line by minimizing the sum of squared residuals; however, shrinking or setting some coefficients to 0 can increase the accuracy of the mentioned model. In the LASSO model, an L_1 regularization term is added to the cost function to address the above-mentioned issue. L_1 regularization is a technique that modifies the objective function of a model by adding a penalty based on the absolute values of the coefficients, leading to simpler and sparser models [29]. The penalty term, λ, controls the amount of regularization. LASSO is a good method to eliminate irrelevant variables and only consider related variables to compute the output of the model. The cost function, J, of the LASSO method is presented in (1). Here, m represents the size of the dataset, while g denotes the model.

$$J_{\text{LASSO}}(\theta) = \frac{1}{2}\sum_{i=1}^{m}(g_{\text{LASSO}}(x_i) - y_i)^2 + \lambda\|\theta\|_1 \tag{1}$$

In this paper, the LASSO method is utilized to predict the DAM price, the RTM price, and the price gap between the DAM and RTM directly. LASSO can set some coefficients to zero, so it can perform variable selection. On the other hand, LASSO has difficulties handling correlated features. One of the correlated features will have a high coefficient, while the rest will be nearly zero. However, this one feature is selected randomly. In addition, the LASSO algorithm can only learn linear mappings; thus, due to the nature of non-linearity in the existing dataset, it may not be the best family of functions in the hypothesis space.

2.2. Support Vector Regression (SVR)

The SVR method is a non-linear learning algorithm. One of the most common versions of SVR regression is ϵ-SV regression. The goal of ϵ-SV regression is to find a function that has the most ϵ divergence for all the data points. The algorithm accepts errors only within the range of ϵ, as presented in (2)–(5).

$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2}w_2^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{2}$$

$$\text{subject to} \quad w^T\phi(x_i) + b - y_i \leq \epsilon + \xi_i \tag{3}$$

$$y_i - w^T\phi(x_i) - b \leq \epsilon + \xi_i^* \tag{4}$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, m \tag{5}$$

Here, the constant, $C > 0$, balances the flatness of a function and the amount up to which deviations larger than ϵ are tolerated. $\phi(x_i)$ maps x_i into a higher-dimensional space, where w and b are coefficients. ξ and ξ^* represent the distance from the actual values to the margin of the ϵ-tube with support vectors. Errors outside the margin are penalized linearly.

A predictor, g, of the SVR with m-training examples is presented in (6).

$$g_{\text{SVR}}(x) = \sum_{i=1}^{m}(-\alpha_i + \alpha_i^*)K(x_i, x) + b. \tag{6}$$

The SVR with an RBF kernel is a non-linear algorithm, and it enables choosing the acceptable error of the model. The hyper-parameter ϵ controls the maximum acceptable error for the model. Thus, it is expected that the SVR with an RBF kernel predicts prices better than the LASSO algorithm.

2.3. Random Forest Algorithm

The Random Forest is an ensemble learning algorithm. It combines multiple weak models to build a strong predictor by taking advantage of methods called bagging and decision trees. The goal of the decision tree algorithm is to build a tree-like structure from the existing data points, where each leaf will only contain labels from the same class. The algorithm will split the dataset into roughly two halves until the leaves are pure. To find the best split that will keep the tree compact, the impurity function is minimized. In the case of regression tasks, usually the squared loss, as given in (7), is used as an impurity function, while classification problems employ the Gini impurity, as presented in (8).

$$L(D) = \frac{1}{|D|}\sum_{(x,y)\in D}(y - \bar{y}_D)^2 \tag{7a}$$

$$\text{where } \bar{y}_D = \frac{1}{|D|}\sum_{(x,y)\in D} y \tag{7b}$$

Given a dataset, $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, with c distinct categories, where D_k is all the inputs with the label k, the squared loss impurity outputs the average squared difference of the actual value and the average prediction, while the Gini impurity measures the homogeneity of the classes.

$$G(D) = \sum_{k=1}^{c} p_k(1 - p_k) \tag{8a}$$

$$\text{where } p_k = \frac{|D_k|}{|D|} \tag{8b}$$

Decision trees learn the exact patterns in the training set, so they do not generalize well enough, so they are prone to overfitting. The Random Forest uses bagging to decrease the high variance caused by decision trees. Bagging generates datasets $D_1, \ldots, D_m$ from the existing data points, D. The created datasets are the replicated datasets, each consisting of k features drawn at random but with replacements from the original dataset [30]. The new datasets are equal in size to the original dataset and have approximately the same probability distribution.

The Random Forest consists of large number of decision trees, $h(x, D_m)$, from $D_1, \ldots, D_m$, where D_m is an independent, identically distributed vector [31]. In the case of classification tasks, the majority vote acquired from all the decision trees will be the prediction, and for regression purposes, the average of all the predictors will be the output. Moreover, the RF algorithm has only two hyper-parameters, m and k. Based on empirical evidence, a good choice for k is $k = \sqrt{d}$, where d is the total number of features in the dataset, and increasing the size of m will only benefit the model.

The RF algorithm performs feature selection and generates uncorrelated decision trees by choosing a random set of features to build each decision tree. In addition, by averaging the results from each decision tree that the model builds, it also averages variance as well. Consequently, the RF can balance the bias-variance trade-off well. Thus, in this paper, the RF algorithm is utilized to select the most important features.

2.4. Long Short-Term Memory (LSTM) Networks

Neural networks try to model the behavior of the human brain. They consist of artificially created neurons and a set of edges that connect those neurons. Furthermore, each neuron has its associated activation function, which models neuron impulses. The RNN is a special type of neural network, where the input is a sequence. An RNN is very powerful because it not only uses the input to predict the output but it also utilizes the information from previously observed timestamps. All RNNs form a sequence of connected units that represent the state of the network at a timestamp, t. A single module takes data from the previous unit, $h_t - 1$, and input for that timestamp, x_t, then uses the *tanh* function to compute the output for a timestamp, t. According to [32], a finite-sized RNN can compute any function that exists. However, RNNs suffer either from exploding or vanishing gradients when back-propagating through time. To update the weights, the neural network computes partial derivatives of the loss function of the current layer at each timestamp. Consequently, when the gradients are very small, either learning happens at a very slow rate or not at all. To overcome this issue with RNNs, the LSTM network is proposed, as suggested in [33]. The LSTM network is a special kind of RNN architecture. Instead of only using the tanh function in a unit, the LSTM network utilizes three gate units: a forget gate, an input gate, and an output gate. The forget gate is responsible for keeping only the relevant information, as given in (9a). It takes an input at timestamp x_t and the data from the previous hidden layer, $h_t - 1$. Then, the sigmoid function is applied to those inputs, and as a result, the output of the forget gate is somewhere between 0 and 1. The output closer to 0 will be forgotten, and the output with a numeric value of 1 will be kept for further calculations. Furthermore, the input gate decides how the memory cell will be updated, as shown in (9b). First, the candidate value is computed using (9c), then the result is scaled by the output of the input gate to decide by how much the cell state will be updated, as shown in (9d). Finally, the LSTM network employs an output gate, which is a filtered version of the cell state. First, the cell state is normalized using the tanh function, then the sigmoid layer that is presented in (9e) is utilized to decide which parts of the memory will be output, as presented in (9f). The outputs of the hidden state, h_t, and the prediction, y_t, is the same; however, the notation h_t are used as a hidden state input at timestamp $t + 1$. The structure of the LSTM network for a single unit is given in Figure 1.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{9a}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{9b}$$

$$\tilde{c_t} = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{9c}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c_t} \tag{9d}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{9e}$$

$$h_t = \tanh(c_t) \otimes o_t \tag{9f}$$

The LSTM architecture is better suited for time series problems compared to the other mentioned algorithms. The LSTM model will learn the previously observed sequence before predicting the output, whereas the mentioned models treat each row in the dataset as an individual training sample.

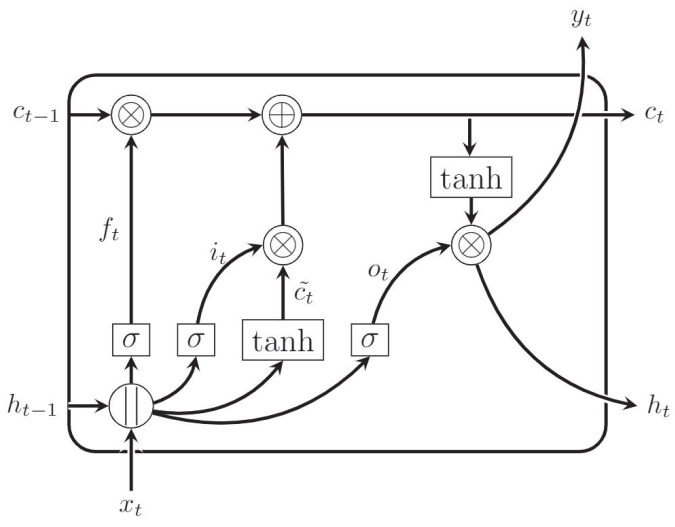


Figure 1. The LSTM architecture of a single unit.

3. Prediction Performance Evaluation

The metrics introduced in Table 1 are presented to measure the performance of the learning algorithms presented in the previous section. Here, y_t is the actual value at time t, while $\hat{y}_t$ is the predicted value for the same timestamp. The maximum and minimum values of all the actual values are represented by $y_{\max}$ and $y_{\min}$. In addition, s is the number of samples in the testing dataset. The mean absolute error (MAE) measures the average magnitude of the errors between the predictions and actual values. Similarly, the RMSE also expresses the average model prediction error. However, it is measured by taking the square root of the average of the squared differences between the actual and predicted values. Both of these metrics measure prediction errors, and they can range from 0 to ∞. Consequently, lower values characterize a better-performing model. The error metric nRMSE outlined represents the normalized RMSE value. In this case, the normalization is carried out by dividing the RMSE score by the difference between the maximum and minimum values of the actual values. Furthermore, the metric max error represents the evaluation of the worst-case scenario and measures the maximum error in the predicted value of the samples.

Table 1. Prediction errors for Station A while using temporal data for various learning algorithms.

MAE	$\left[\frac{1}{s}\sum_{t=1}^{s}\lvert y_t - \hat{y}_t\rvert\right]$
RMSE	$\sqrt{\frac{1}{s}\sum_{t=1}^{s}(y_t - \hat{y}_t)^2}$
nRMSE	$\sqrt{\frac{1}{s}\sum_{t=1}^{s}(y_t - \hat{y}_t)^2}\Big/[y_{\max} - y_{\min}]$
Max Error	$max(\lvert y_t - \hat{y}_t\rvert)\forall t \in \{0, s\}$

4. Data Preparation

4.1. Data Collection

Three distinct datasets are collected and merged to form an extensive dataset for studying the price gap across the DAM and RTM. The first dataset is the one with information from the electricity market. The California Independent System Operator (CAISO) provides an Open-Access Same-time Information System (OASIS) Application Programming Interface (API), which produces reports for the energy market and power grid information

in real-time. To demo the results of this paper, the *MURRAY6N015* node, located in San Diego, CA, is chosen, and its reports for the energy market and power grid information in real-time are leveraged. The LMP, the LMP congestion component, the LMP energy component, and the LMP loss component are collected for the DAM and RTM. Furthermore, a seven-day-ahead load forecast as well as load forecasts for the next two days are acquired using the CAISO API. The period of collection for the dataset is two years, starting from the 1 January 2017.

The second one has meteorological data. For the historical hourly weather dataset, the Meteostat API is used. Meteostat collects hourly weather measurements from more than 5000 weather stations around the world. In addition, it offers comprehensive historical datasets that combine their measurements with the NOAA's Global Historical Climatology Network's dataset. The weather data are obtained from the San Diego International Airport weather station, which is the closest weather station to the node of interest. The collected data include information about temperature, dew point, humidity, wind speed, wind direction, weather condition, sea level pressure, wind gust, cloud layers, and weather forecasts for the next 3 and 6 days. Weather conditions directly and indirectly influence both the demand and supply of electricity, which in turn affects the price. Solar power generation is directly influenced by the amount of sunlight, while the production of electricity from wind turbines depends on wind speeds. If wind speeds are predicted to be low, wind-generated electricity might be reduced, potentially leading to higher prices. On the other hand, weather conditions may influence the demand for electricity. Extreme temperatures, both hot and cold, increase the demand for electricity.

The third dataset concerns renewable energy availability. The mesonet API is utilized to acquire a dataset for solar irradiation. This API offers quality-controlled, surface-based environmental data such as Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), solar zenith angle, cloud type, and precipitable water. The GHI is the total amount of terrestrial irradiance received from above by a surface horizontal to the ground. The DNI means the radiation that comes in a straight line directly from the sun and is absorbed by a unit perpendicular to the rays. Furthermore, the DHI is the radiation that does not arrive on a direct path from the sun, and it is equally absorbed by the particles in the atmosphere. It should be noted that the historical weather data (i.e., the second dataset), the forecast weather data (i.e., the third dataset), and the forecast demand data (i.e., the first dataset) are utilized.

4.2. Data Cleansing and Pre-Processing

The collected datasets are merged based on the date and the hour of the day. Only the data from the time span of 1 January 2017, 00:00 to 30 December 2018, 23:00 are utilized. Data cleansing techniques are applied to ensure the quality of the data. Duplicate rows are dropped, categorical variables are converted to numerical representations, and every measurement is converted to a floating-point value. In addition, the missing values are substituted with a global constant. Mean imputation is performed to address cases in which the measurements from the weather dataset are not available at a particular timestamp. Mean imputation replaces all missing values with a mean value calculated across the whole dataset. After data cleansing, since the data for some hours are missing from the official API, 16,566 h worth of data are available. The dataset is arbitrarily split into two parts. The earlier 90% of the data are used for training, and the later 10% are utilized for testing purposes. Then, the input data are normalized using a Min–Max scaler. As a result, each feature is converted into a $\{0, 1\}$ range. The Random Forest model is employed to select features. The data collected are extensive and combine three different datasets. Consequently, it is important to showcase which features contribute to prediction and which are insignificant. Moreover, feature selection ensures that features that do not affect the prediction are removed and do not introduce extra noise into the system.

5. Simulation Results

5.1. Feature Importance

LASSO, SVR, and Random Forest algorithms can not inherently capture temporal dependencies for sequential data; that is why day-ahead prices for a previous 48 h time horizon are added as features. Therefore, 48 new columns are created that contain the delayed values of the DAM LMP. Similarly, lagged real-time and gap values are added to the existing dataset; however, for the RTM, the most recent prices that are available are at $t - 12$ h. Consequently, only those features that are realistically available for the RTM are taken into consideration.

In this section, the RF algorithm is procured to select the most important features. Since the RF method employs decision trees, it can be leveraged for feature selection. The RF naturally ranks by how well each decision tree improves the purity of the node. The Gini index of decision tree algorithms is leveraged to procure feature importance values. For example, the greatest decrease in impurity happens at the root of the tree, while the least decrease in impurity happens at the leaves of the tree. Consequently, pruning the tree below a particular node creates a subset of the most relevant features. In comparison to PCA, the above-described algorithm captures the non-linear dependencies of the features, while PCA only captures linear relationships between features. Table 2 presents the selected features, ranked by their importance for predictions of the price gap between the two markets. The RF algorithm renders 204 features useful for gap prediction. Note that the 13 most relevant features are shown in Table 2. The right column of Table 2 represents the importance coefficient. The importance coefficient is scaled so that the sum of all the importance coefficients is 100. It is interesting that the external features that are collected demonstrated a significant effect on predicting the price gap between the two markets. For example, the solar zenith angle has an importance coefficient of 1.0, while the DHI contributes to the prediction with a 0.39 importance score.

Table 2. Selected feature importance for gap predictions.

Feature	Importance Coefficient
GAP LMP price 24 h before	3.32
DAM LMP price 24 h before	1.9
RTM LMP price 24 h before	1.3
Solar Zenith Angle	1.0
Demand Forecast Day-Ahead	0.78
Relative humidity	0.67
Perceptible Water	0.61
Cloud Layer	0.54
Dew point	0.53
Wind Speed	0.47
Wind Direction	0.46
Demand Forecast 2 Days Ahead	0.46
DHI	0.39

5.2. Hyper-Parameter Tuning

To perform day-ahead, real-time gap predictions, the hyper-parameters of each learning method presented in Section 2, are optimized. Hyper-parameters control the learning process, and they have to be optimized so that the predefined loss function is minimized for a given dataset. A grid search with nested cross-validation is used to tune hyper-parameters. A grid search is a brute force algorithm that calculates the output for all subsets of predefined parameters and picks the best estimator. The performance of the estimators

is evaluated using time-series nested K-Fold cross-validation, where $k = 5$. The time-series nested cross-validation divides the existing dataset into k inner loops, and each inner loop is split into a training subset and a validation set. Then, the parameters that minimize the error on the validation set are chosen. The outer loop splits the dataset into multiple different training and test sets, and the error on each split is averaged to compute a robust estimate of the model's error. K-fold cross-validation helps mitigate the risk of overfitting and provides a more reliable assessment of how well the model is expected to perform on unseen data.

The Lasso algorithm presented in (1) has only one hyper-parameter: λ. To find the optimal value for λ, a set of arbitrarily chosen values, {0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.001, 0.002, 0.003, 0.004, 0.005, 0.01}, is examined. The highest accuracy or minimal loss is acquired using a grid search when the hyper-parameter λ is 0.0003.

The SVR model is optimized for four different hyper-parameters, including C; ϵ; the kernel function, K; and the kernel coefficient, γ. The optimal value found using the grid search for the constant C is $C = 1000$, while the margin of the tube $\epsilon = 0.001$ turned out to give the most accurate estimator. In addition, different kernel functions, including linear, sigmoid, and RBFs, are tested, and the most accurate results are obtained using an RBF with $\gamma = 0.1$.

Similarly, the Random Forest algorithm is also tuned for hyper-parameters. Generally, an increase in the number of trees in the forest can only benefit the algorithm. However, this increment also introduces significant overhead in computation time, so only forests with 50 and 100 trees were tested, and 100 trees turned out to give more accurate results.

In addition, the maximum number of features considered when looking for the best split turned out to be equal to the total number of features. Moreover, different maximum depths of the trees are passed to the grid search, and the optimal value is found when the nodes are expanded until all the leaves are pure. Finally, the algorithm is tuned for the methods of sampling the data points, and sampling with replacement turned out to be the optimal option.

The LSTM network is optimized for the number of units, loss function, optimizer, and lookback period, which represent several previous timestamps that are considered for a prediction at each time unit. The following set of values, $\{10, 20, 50, 100\}$, is examined for the number of units, while the MAE and MSE are tested for the loss function, and Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), and Adaptive Moment Optimization (ADAM) are utilized for optimizer choices. Note that RMSProp is a gradient-based optimization technique that uses the moving average of squared gradients to normalize the gradient, while ADAM is a combination of RMSProp and SGD. Moreover, for lookback options, a day, a week, and a month are tested, and for epoch numbers, the following set of values are examined: $\{10, 20, 50, 100\}$. It turned out that 100 LSTM cells with a loss function of the MSE and with an ADAM optimizer resulted in the most accurate results. In addition, the optimal lookback period is a day, and the best number of epochs is 100. Note that here, the different learning methods presented in Section 2 are utilized to predict the electricity prices for the DAM and RTM and the direct gap between DAM and RTM prices.

5.3. Analysis of Probability Distributions

The Random Forest algorithm described in Section 2 is used to calculate the probability distribution of the predicted electricity prices for the DAM and RTM. The outputs of the 100 regression trees are used to approximate the probability distribution for both markets. The spread of predictions from the individual decision trees showcases the uncertainty and variance of the predictions. To represent the results, the 15 October 2018 is chosen for the test, and the hours of interest are 8 a.m. and 5 p.m.

As shown in Figure 2, the prices for the DAM at 8am range from 18 USD/MWh to 44 USD/MWh, while the prices for the same market at 5 p.m. range from 30 USD/MWh to 62 USD/MWh. For this case study, the RTM prices tend to be in a lower range. The price

prediction for the RTM at 8 p.m. is in the range of −3 USD/MWh to 44 USD/MWh, while the price prediction at 5 p.m. ranges from 17 USD/MWh to 60 USD/MWh. The electricity prices tend to be much higher at 5 p.m. compared to those at 8 a.m.

Direct predictions by leveraging the RF algorithm would render the most promising probability distribution on the price gap. Figures 3 and 4 present the importance of direct gap prediction in comparison to the difference in predicted prices of the DAM and RTM. In direct gap prediction, the target value for the model is the gap price between the DAM and the RTM. However, to calculate the difference between the predicted DAM and RTM, two distinct models are developed to predict DAM and RTM prices, and then the predictions are subtracted. The time and date are the same as in the case study described above. However, in this case, the actual electricity gap price is also displayed to underline the significance of direct gap prediction. The ground truth gap price for the mentioned date at 8 a.m. is 17.4 USD/MWh.

It can be observed from Figure 3 that the probability of the gap procured by the direct gap prediction case is 17.4 USD/MWh higher than the probability acquired by subtracting the day-ahead and real-time price predictions. Figure 4 shows the probability distributions for 5 p.m. The direct gap prediction has a higher chance of being more accurate. The actual gap price for 5 p.m. is 49 USD/MWh. On the one hand, the price range acquired by subtracting the DAM and RTM price predictions is from −60 USD/MWh to 30 USD/MWh; in this case, it would be virtually impossible to correctly predict the actual gap by calculating the difference in predictions for the mentioned markets. On the other hand, the range for direct gap predictions includes the ground truth gap value. Even though, in this case, the probability of accurately predicting the actual gap using the direct gap prediction is not very high, it is still the better choice between those two methods. Furthermore, the direction of the gap difference is more solid toward a positive range in the direct gap prediction than in the difference calculation, which has a smaller expected value.

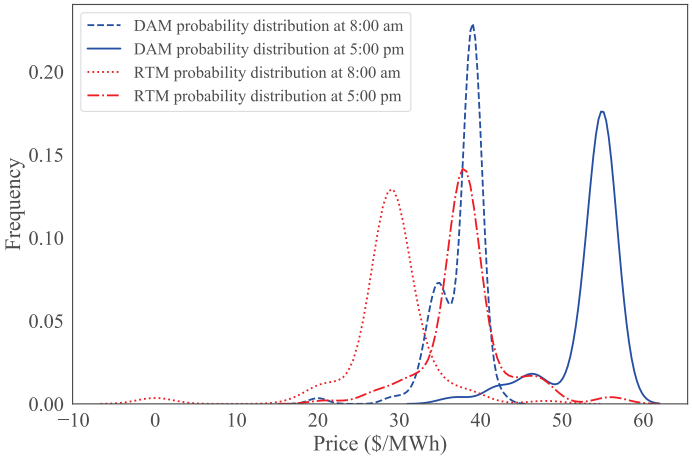


Figure 2. The probability distribution of the DAM and RTM price predictions for a specific date and time, procured by the RF algorithm.

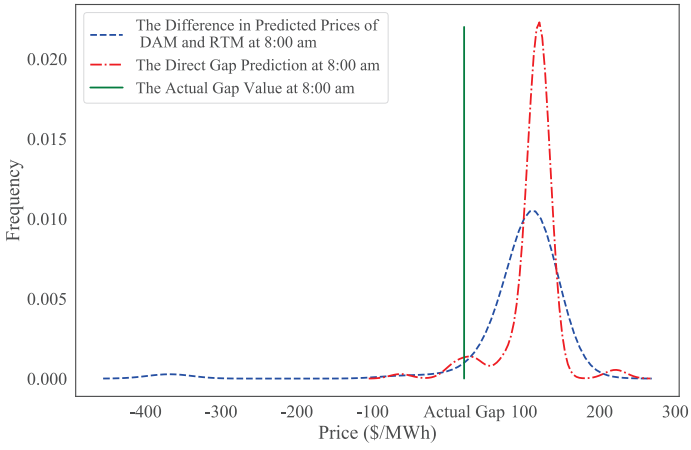


Figure 3. A comparison between the probability distribution of direct gap predictions and the difference between separately predicted prices for the DAM and RTM at 8 a.m., procured by LSTM network.

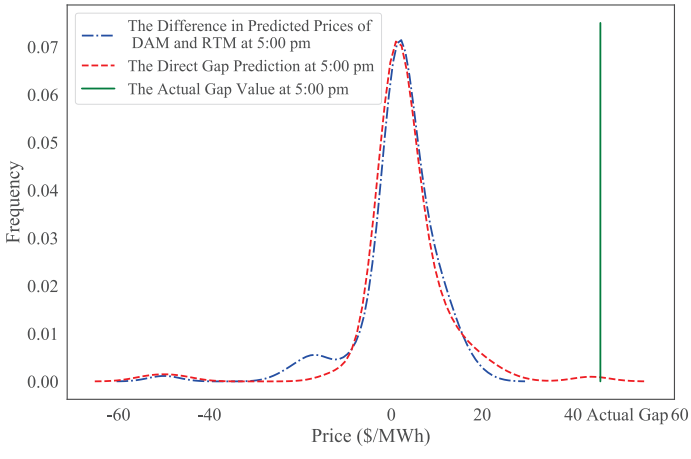


Figure 4. A comparison between the probability distribution of direct gap predictions and the difference between separately predicted prices for the DAM and RTM at 5 pm, procured by LSTM network.

5.4. Performance Evaluation

To understand the quantitative insights for the DAM, RTM, and gap prices, the descriptive statistics of the entire data set are presented in Table 3, where all the values are in USD/MWh. It is worth pointing out that the standard deviation for the DAM was almost half of that for the RTM, which means that the values tend to be closer to the mean in the case of the DAM and prices do not fluctuate as much as the price fluctuations in the case of the RTM. The 25th percentile of all gap prices is less than -0.16, which means that almost a quarter of the direct gap prices are negative.

Table 3. Statistical analysis of DAM, RTM, and gap prices of the data set.

Statistics	Gap	DAM	RTM
Mean	3.1	39.1	36.0
Standard Deviation	87.7	45.6	84.5
Min	−1488.6	−61.6	−262.3
Max	2276.2	2374.4	1545.4
25th Percentile	−0.16	25.0	19.6
50th Percentile	6.6	33.4	27.1
75th Percentile	17.5	46.5	36.7

In Tables 4 and 5, an evaluation of the DAM prices and RTM prices procured by the learning methods presented in Section 2 is illustrated, where the unit for the MAE, RMSE, and max error is USD/MWh. All algorithms performed significantly better at predicting DAM prices than predicting RTM electricity prices. For instance, the LSTM network, which is the best-performing algorithm, had an MAE and RMSE of 4.9 and 7.1, respectively, while, for the RTM, the same algorithm resulted in a MAE of 21.2 and an RMSE of 48. Consequently, the complexity of predicting the gap between these markets is dependent on the accuracy of the prediction of the RTM.

Table 4. Prediction errors for the DAM.

Error Measure	LASSO	SVR	RF	LSTM
MAE	9.6	11.7	5.1	4.9
RMSE	13.3	33.7	7.9	7.1
nRMSE [%]	7.4	39.8	4.4	4.2
Max Error	95.2	122.7	62	40

Table 5. Prediction errors for the RTM.

Error Measure	LASSO	SVR	RF	LSTM
MAE	18.9	21.4	26.4	21.2
RMSE	59	54	71.9	48
nRMSE [%]	5.1	5.0	6.2	4.4
Max Error	1064	1060	1058	1040

LASSO failed to capture any spikes in the price change, and the maximum error between the prediction and the actual gap values was 1054.8 USD/MWh. The poor performance of the LASSO algorithm can be explained by the fact that LASSO is a linear algorithm and leverages a linear function for prediction, while gap prediction should be a non-linear mapping based on empirical evidence.

The most promising results were procured using the RF algorithm and the LSTM network. The Random Forest algorithm had an MAE score of 24.5 USD/MWh and an RMSE score of 67.5 USD/MWh when predicting direct gap prices. Even though these metrics are slightly worse than the above-described algorithms, it can be observed from Figure 5 that the RF algorithm is not as good as the LSTM network. While the RF does a good job of predicting correct values when predicting positive gap prices, it suffers from a notable error in capturing a big negative price spike in gap values. The LSTM network had the best performance in terms of error metrics as well as an empirical evaluation based on the plot provided in Figure 5. Furthermore, Figure 6 illustrates the relative error of the price gap predicted by the LSTM and RF methods. Table 6 shows that the LSTM network

renders the lowest MAE, nRMSE, and Max Error values among all the learning methods used to predict the gap prices. In addition, the LSTM network also outperformed all the methods in predicting the DAM and RTM electricity prices individually.

Table 6. Prediction errors for the gap.

Error Measure	LASSO	SVR	RF	LSTM
MAE	19.6	28.2	24.5	17.1
RMSE	58.9	80.4	67.5	56.9
nRMSE [%]	4.98	6.1	5.7	4.8
Max Error	1054.8	1051	1048	1046

5.5. Importance of Exogenous Weather Information

To illustrate the importance of the collected exogenous features, the learning methods described in Section 2 are leveraged without exogenous weather features to predict gap prices, and the results are compared to the predicted gap prices procured by those methods using collected features. Adding exogenous features such as weather conditions and solar irradiance significantly improved the accuracy of the price gap prediction for all the learning algorithms. The error metrics of all the algorithms without exogenous weather features are presented in Table 7. All the algorithms without exogenous weather features had worse error metrics than those with exogenous weather features, as illustrated by comparing Tables 6 and 7.

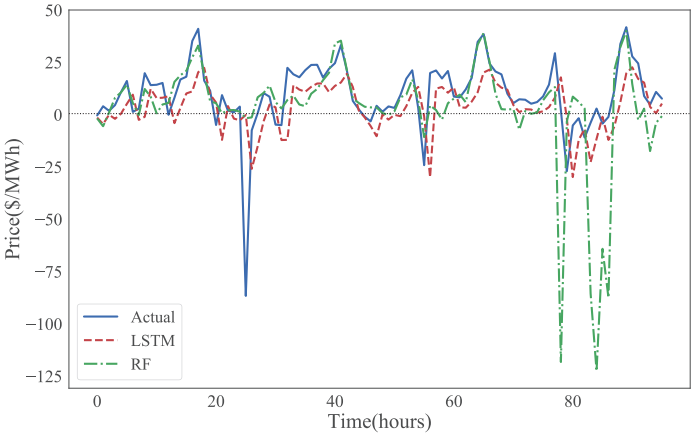


Figure 5. A comparison of predicting the gap using LSTM and RF algorithms for the next 96 h to the actual values of the gap.

Table 7. Prediction errors for the gap without exogenous features.

Error Measure	LASSO	SVR	RF	LSTM
MAE	19.7	44.8	24.7	31.8
RMSE	64	81.6	75	62.15
nRMSE [%]	5.4	6.9	6.3	5.2
Max Error	1053.7	1069	1058	1052

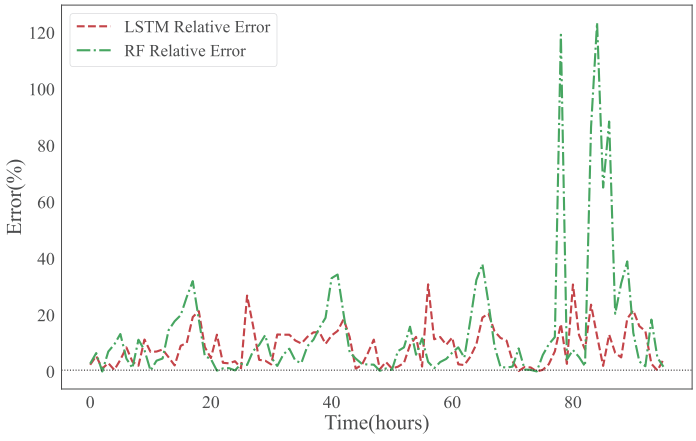


Figure 6. A comparison of the relative error of direct gap predictions procured by LSTM and RF algorithms for the next 96 h.

6. Conclusions

This paper proposed a model to predict the price gap between the RTM and the DAM. To this end, several machine learning algorithms and neural networks are leveraged to obtain the price gap across the DAM and the RTM. To improve the accuracy of the price gap prediction, exogenous weather data, e.g., solar irradiance, is added to the training data of the learning methods and the LSTM. To enable the integration of exogenous weather information, three distinct datasets are collected, matched, and synchronized. It is shown that consideration of related exogenous weather information will outweigh the importance of algorithm selection. Furthermore, this paper investigates the benefits of learning algorithms for direct gap prediction compared to the subtraction of price predictions. To fully achieve this goal, several learning methods are tested to evaluate the performance of the learning algorithms for direct gap prediction compared to the subtraction of price predictions, and it is shown that the prediction error will be lower with a direction price gap prediction. There is no single algorithm that will deliver the best performance all the time. The Random Forest algorithm did a better job of predicting positive gaps as well as the probability distribution of the price gap, while the overall prediction error for the LSTM network was lower. Thus, for future work, it is recommended to consider a combination of the Random Forest algorithm and the LSTM network to predict the price gap. While the former does predict the sign of the gap relatively well, the latter will be able to determine the value of the gap given the sign.

Author Contributions: Conceptualization, S.M. and N.N.; methodology, N.N. and A.F.S.; software, N.N. and A.F.S.; validation, S.M., N.N. and A.F.S.; formal analysis, N.N. and A.F.S.; data curation, N.N.; writing—original draft preparation, N.N. and A.F.S.; writing—review and editing, A.F.S. and S.M.; supervision, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to access permission restrictions.

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

References

1. Stoft, S. *Power System Economics: Designing Markets for Electricity*; IEEE Press: Piscataway, NJ, USA, 2002; Volume 468.
2. Woo, C.K.; Moore, J.; Schneiderman, B.; Ho, T.; Olson, A.; Alagappan, L.; Chawla, K.; Toyama, N.; Zarnikau, J. Merit-order effects of renewable energy and price divergence in California's day-ahead and real-time electricity markets. *Energy Policy* **2016**, *92*, 299–312. [CrossRef]

3. Khoshjahan, M.; Baembitov, R.; Kezunovic, M. Impacts of weather-related outages on DER participation in the wholesale market energy and ancillary services. In Proceedings of the CIGRE Grid of the Future Symposium, Providence, RI, USA, 17–20 October 2021.
4. Li, Y.; Flynn, P. Electricity deregulation, spot price patterns and demand-side management. *Energy* **2006**, *31*, 908–922. [CrossRef]
5. Benini, M.; Marracci, M.; Pelacchi, P.; Venturini, A. Day-ahead market price volatility analysis in deregulated electricity markets. In Proceedings of the IEEE Power Engineering Society Summer Meeting, Chicago, IL, USA, 21–25 July 2002; Volume 3, pp. 1354–1359.
6. Veit, D.J.; Weidlich, A.; Yao, J.; Oren, S.S. Simulating the dynamics in two-settlement electricity markets via an agent-based approach. *Int. J. Manag. Sci. Eng. Manag.* **2006**, *1*, 83–97. [CrossRef]
7. Yao, J.; Adler, I.; Oren, S.S. Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network. *Oper. Res.* **2008**, *56*, 34–47. [CrossRef]
8. Bayani, R.; Soofi, A.F.; Manshadi, S.D. Strategic Competition of Electric Vehicle Charging Stations in a Regulated Retail Electricity Market. *arXiv* **2021**, arXiv:2111.11592.
9. Mehdipourpicha, H.; Bo, R. Optimal Bidding Strategy for Physical Market Participants With Virtual Bidding Capability in Day-Ahead Electricity Markets. *IEEE Access* **2021**, *9*, 85392–85402. [CrossRef]
10. Kohansal, M.; Sadeghi-Mobarakeh, A.; Manshadi, S.D.; Mohsenian-Rad, H. Strategic Convergence Bidding in Nodal Electricity Markets: Optimal Bid Selection and Market Implications. *IEEE Trans. Power Syst.* **2020**, *36*, 891–901. [CrossRef]
11. Kohansal, M.; Samani, E.; Mohsenian-Rad, H. Understanding the Structural Characteristics of Convergence Bidding in Nodal Electricity Markets. *IEEE Trans. Ind. Informatics* **2020**, *17*, 124–134. [CrossRef]
12. Fan, S.; Liao, J.R.; Kaneko, K.; Chen, L. An integrated machine learning model for day-ahead electricity price forecasting. In Proceedings of the 2006 IEEE PES Power Systems Conference and Exposition, PSCE 2006, Atlanta, GA, USA, 29 October–1 November 2006; pp. 1643–1649. [CrossRef]
13. Contreras, J.; Espínola, R.; Nogales, F.J.; Conejo, A.J. ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [CrossRef]
14. Sadeghi-Mobarakeh, A.; Kohansal, M.; Papalexakis, E.E.; Mohsenian-Rad, H. Data mining based on random forest model to predict the California ISO day-ahead market prices. In Proceedings of the 2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2017, Washington, DC, USA, 23–26 April 2017. [CrossRef]
15. Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [CrossRef]
16. Gao, G.; Lo, K.; Fan, F. Comparison of ARIMA and ANN models used in electricity price forecasting for power market. *Energy Power Eng.* **2017**, *9*, 120–126. [CrossRef]
17. Adebiyi, A.A.; Adewumi, A.O.; Ayo, C.K. Comparison of ARIMA and artificial neural networks models for stock price prediction. *J. Appl. Math.* **2014**, *2014*, 614342. [CrossRef]
18. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]
19. He, K.; Yang, Q.; Ji, L.; Pan, J.; Zou, Y. Financial Time Series Forecasting with the Deep Learning Ensemble Model. *Mathematics* **2023**, *11*, 1054. [CrossRef]
20. Baldo, A.; Cuzzocrea, A.; Fadda, E.; Bringas, P.G. Financial forecasting via deep-learning and machine-learning tools over two-dimensional objects transformed from time series. In Proceedings of the Hybrid Artificial Intelligent Systems: 16th International Conference, HAIS 2021, Bilbao, Spain, 22–24 September 2021; pp. 550–563.
21. Zheng, J.; Xu, C.; Zhang, Z.; Li, X. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems, CISS 2017, Baltimore, MD, USA, 22–24 March 2017; pp. 1–6. [CrossRef]
22. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies* **2018**, *11*, 1636. [CrossRef]
23. Jiang, L.; Hu, G. Day-Ahead Price Forecasting for Electricity Market using Long-Short Term Memory Recurrent Neural Network. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018, Singapore, 18–21 November 2018; pp. 949–954. [CrossRef]
24. Aggarwal, S.K.; Saini, L.M.; Kumar, A. Electricity price forecasting in deregulated markets: A review and evaluation. *Int. J. Electr. Power Energy Syst.* **2009**, *31*, 13–22. [CrossRef]
25. Ji, Y.; Kim, J.; Thomas, R.J.; Tong, L. Forecasting real-time locational marginal price: A state space approach. In Proceedings of the Conference Record—2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 379–383. [CrossRef]
26. Mujeeb, S.; Javaid, N.; Ilahi, M.; Wadud, Z.; Ishmanov, F.; Afzal, M.K. Deep long short-term memory: A new price and load forecasting scheme for big data in smart cities. *Sustainability* **2019**, *11*, 987. [CrossRef]
27. Zhang, Z.; Wu, M. Predicting real-time locational marginal prices: A GAN-based video prediction approach. *arXiv* **2020**, arXiv:2003.09527.

28. Zahid, M.; Ahmed, F.; Javaid, N.; Abbasi, R.A.; Kazmi, H.S.Z.; Javaid, A.; Bilal, M.; Akbar, M.; Ilahi, M. Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. *Electronics* **2019**, *8*, 122. [CrossRef]
29. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* **1996**, *58*, 267–288. [CrossRef]
30. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
32. Siegelmann, H.T.; Sontag, E.D. Turing computability with neural nets. *Appl. Math. Lett.* **1991**, *4*, 77–80. [CrossRef]
33. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

algorithms

MDPI

Article

Encouraging Eco-Innovative Urban Development

Victor Alves [1,2,3], Florentino Fdez-Riverola [2,3], Jorge Ribeiro [1], José Neves [4,5] and Henrique Vicente [4,6,*]

[1] ADiT-LAB, Instituto Politécnico de Viana do Castelo, Rua da Escola Industrial e Comercial de Nun'Alvares, 4900-347 Viana do Castelo, Portugal; vitoralves@estg.ipvc.pt (V.A.); jribeiro@estg.ipvc.pt (J.R.)
[2] CINBIO, Department of Computer Science, ESEI—Escuela Superior de Ingeniería Informática, Universidade de Vigo, 32004 Ourense, Spain; riverola@uvigo.es
[3] SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), SERGAS-UVIGO, 36213 Vigo, Spain
[4] Centro Algoritmi/LASI, Universidade do Minho, Campus de Gualtar, Rua da Universidade, 4710-057 Braga, Portugal; jneves@di.uminho.pt
[5] Instituto Universitário de Ciências da Saúde, CESPU, Rua José António Vidal, 81, 4760-409 Famalicão, Portugal
[6] Departamento de Química e Bioquímica, Escola de Ciências e Tecnologia & REQUIMTE/LAQV, Universidade de Évora, Rua Romão Ramalho, 59, 7000-671 Évora, Portugal
* Correspondence: hvicente@uevora.pt

Abstract: This article explores the intertwining connections among artificial intelligence, machine learning, digital transformation, and computational sustainability, detailing how these elements jointly empower citizens within a smart city framework. As technological advancement accelerates, smart cities harness these innovations to improve residents' quality of life. Artificial intelligence and machine learning act as data analysis powerhouses, making urban living more personalized, efficient, and automated, and are pivotal in managing complex urban infrastructures, anticipating societal requirements, and averting potential crises. Digital transformation transforms city operations by weaving digital technology into every facet of urban life, enhancing value delivery to citizens. Computational sustainability, a fundamental goal for smart cities, harnesses artificial intelligence, machine learning, and digital resources to forge more environmentally responsible cities, minimize ecological impact, and nurture sustainable development. The synergy of these technologies empowers residents to make well-informed choices, actively engage in their communities, and adopt sustainable lifestyles. This discussion illuminates the mechanisms and implications of these interconnections for future urban existence, ultimately focusing on empowering citizens in smart cities.

Keywords: smart cities; artificial intelligence; machine learning; digital transformation; computational sustainability; logic programming; the laws of thermodynamics; entropy

Citation: Alves, V.; Fdez-Riverola, F.; Ribeiro, J.; Neves, J.; Vicente, H. Encouraging Eco-Innovative Urban Development. *Algorithms* **2024**, *17*, 192. https://doi.org/10.3390/a17050192

Academic Editors: Gloria Cerasela Crisan and Elena Nechita

Received: 11 April 2024
Revised: 26 April 2024
Accepted: 28 April 2024
Published: 1 May 2024

1. Introduction

In the evolving domain of urban innovation, smart cities (SCs) symbolize a transformative agenda, as described by Batty et al. [1], and further refined by Allam and Newman [2]. Here, the incorporation of technology into urban spaces acts as a propellant for sustainable, efficient living. Anthopoulos [3] underscores this strategy's focus on interconnectivity and sophisticated technology to improve residents' lives. This discourse delves into the synergistic bond between artificial intelligence (AI), machine learning (ML), digital transformation (DT), and computational sustainability (CS)—the four critical tenets shaping the SC concept and enhancing citizen empowerment, resonating with views from Caragliu et al. [4], Angelidou [5], and Bibri and Krogstie [6]. Kitchin [7] observes that the digital age has propelled technology forward, prompting cities worldwide to adopt AI and ML as vital components of urban ingenuity, a sentiment shared by Amović et al. [8]. These tools are proficient at handling large datasets, pivotal for bolstering urban operations, automating processes, and customizing services, as suggested by Komninos et al. [9]. AI and ML also

allow urban areas to anticipate societal necessities, oversee intricate systems, and promptly respond to crises, expanding upon Neirotti et al. [10]. Digital transformation extends beyond the scope outlined by Chourabi et al. [11], not merely digitizing public amenities but revolutionizing urban governance and service provision, promoting inclusivity, efficacy, and a citizen-focused approach. Wirtz et al. [12] argue this shift can diminish bureaucratic barriers, enhance transparency, and spur economic growth, advancing the conversation initiated by Batty [13]. CS, a cornerstone of the SC concept emphasized by Albino et al. [14], is crucial amidst pressing environmental challenges. AI, ML, and digital advancements aid in resource optimization, waste reduction, carbon footprint shrinking, and the pursuit of sustainable growth. These initiatives are key in monitoring environmental indicators, endorsing clean energy, refining waste management, and encouraging residents towards sustainable living. At the nexus of AI, ML, DT, and CS lies citizen empowerment in SCs. Harnessing these technological strides, citizens gain the means to make well-informed decisions, actively participate in their communities, and pursue sustainable practices. These technologies afford access to real-time transport data, digital platforms for civic engagement, and bolster an informed, proactive, and resilient populace. This analysis endeavors to unravel the intricate interplay among these intertwined elements and their collective impact on citizen empowerment within SCs. It strives to shed light on these technologies' role in enhancing urban life, navigate the path for cities seeking smart evolution, and underscore pivotal considerations for the future of urban living, a concept emphasized by Meijer and Bolíver [15].

The Synergy between Sustainable Development, Green Technology, Corporate Social Responsibility, and Innovation

For a smart city model to be sustainable, it must incorporate environmental considerations (related to CS) alongside innovative practices (related to DT), powered by data-driven decisions (related to AI and/or ML) [16]. Indeed, the intersection of AI, ML, DT, CS, and SC represents a collaborative framework essential for advancing urban innovation [17]. The logical interconnection of these concepts fosters a harmonious blend of technology and sustainability, which is crucial to the development and functioning of SCs [18]. AI and ML serve as the brain of SCs, equipping urban systems with the capability to process vast amounts of data, enabling adaptive and predictive operations [18]. This intelligence is not just about the automation of tasks but extends to the comprehension of patterns and behaviors within an urban context, allowing for the efficient allocation of resources and better decision-making. DT is the circulatory system of this organism, vital for conveying the benefits of technology to every corner of the urban fabric [19]. It facilitates the transition from traditional practices to digitally enabled governance and service delivery, ensuring that efficiency, transparency, and inclusivity are embedded in the city's operations. It is the pathway through which AI and ML solutions reach the hands of citizens and administrators. CS acts as the lungs, ensuring that the city's growth is not at the expense of its environment [20]. Rieder et al. [21] outline how the above-mentioned technological advancements are leveraged to enhance the quality of life for residents, promote efficient and automated urban living, and foster sustainable development. In alignment with the findings of Ortega-Fernández et al. [22], the core argument of the Rieder's study revolves around the symbiotic relationship between technology and urban development, highlighting how AI and ML serve as foundational elements in processing large datasets, automating processes, and personalizing services to meet the dynamic needs of urban populations [21]. Several authors highlight DT as a powerful force that integrates digital technology across all facets of urban life, thereby enhancing value delivery to citizens. These studies underscore the significance of CS in realizing the goal of fostering environmentally responsible and sustainable cities through the efficient utilization of digital resources [23–25].

By utilizing AI and ML, cities can optimize energy consumption, reduce waste, and promote sustainable practices, making the city not only smarter but also greener. SCs, as the tangible manifestation of these combined efforts, embody the holistic body that benefits

from the synergies of its parts. In SCs, citizen empowerment is paramount, the inhabitants are not mere bystanders but active participants, equipped with real-time data and platforms for civic engagement provided through the DT processes. However, if these concepts were to operate in silos without a logical interconnection, there is a risk of creating a semblance of interdependence that is arbitrary and not genuinely integrated. This disjunction can lead to inefficiencies, underutilization of data, and a failure to meet sustainable development goals. Without AI and ML's predictive analytics, DT might only result in superficial changes without deep systemic transformation. Without DT, the advancements of AI and ML would remain inaccessible to the broader population. Without a focus on CS, technological progress could lead to unsustainable practices that harm the environment and society in the long term. Therefore, it is imperative that AI, ML, DT, CS, and SCs not only coexist but are deeply intertwined, each reinforcing the other to create a robust, responsive, and sustainable urban ecosystem. The synergy among these components is what allows for the intelligent evolution of cities, fostering environments where technology serves the people and the planet in a balanced and thoughtful manner. SCs thrive on this interconnectedness; it is the bedrock of their innovation, ensuring that technological advancements equate to an improved quality of life for all citizens.

2. Exploring the Influence of Entropy in Knowledge Expression and Logical Discourse

The study introduces a novel methodology for evaluating entropic efficiency in problem-solving scenarios. It builds on the concept that entropy ranges between 0 and 1, with lower values indicating order and higher ones reflecting disorder [26–28]. Situated in the realm of Knowledge Representation and Reasoning (KRR) within AI, the focus is on structuring knowledge for computational interpretation and developing algorithms for intelligent decision-making using such knowledge [29]. KRR employs formal languages like First-Order Logic, Description Logics, and frame-based systems for organizing information, which reasoning algorithms then utilize to answer queries and tackle complex challenges. The methodology draws an analogy between KRR and thermodynamics, likening the process of energy degradation to the way usable energy decreases over time, an idea echoing the First and Second Laws of Thermodynamics. The First Law dictates energy conservation within an isolated system, implying energy transformation rather than loss. The Second Law introduces entropy as a measure of systemic order, delineating the natural trend towards disorder [30,31]. In KRR, the entropic state signifies the quantum of energy that diminishes yet never vanishes entirely [27,32]. This is characterized, viz.

- Exergy, reflecting the portion of energy that can be harnessed;
- Vagueness, denoting the potential energy that could have been exploited; and
- Anergy, indicating the potential of energy that remains unutilized.

KRR methodologies, especially in Model Theory [33,34] and Proof Theory [35,36], intertwine with Logic Programming (LP) principles. The paper leverages a Proof Theoretical framework to extend the LP paradigms for problem resolution. It develops a Logic Program with a well-defined set of clauses or archetypes to exemplify the application of these theories (Program 1) [36,37], viz.

Program 1. *The quintessential instance of a logical entity.*

$$
\begin{aligned}
&\{\\
&\quad \neg\ p \leftarrow not\ p,\ not\ exception_p\\
&\quad p \leftarrow p_1, \ldots, p_n,\ not\ q_1, \ldots,\ not\ q_m\\
&\quad ?\ (p_1, \ldots, p_n,\ not\ q_1, \ldots,\ not\ q_m)\ (n,\ m \geq 0)\\
&\quad exception_{p_1}, \ldots, exception_{p_j}\ (0 \leq j \leq k)\ being\ k\ an\ integer\ number\\
&\}
\end{aligned}
$$

This approach integrates foundational ground literals and assertive propositions, along with negation-as-failure—a principle asserting that a proposition is deemed false if it cannot be proven true due to the absence of explicit evidence [36]. Within this structure, each program comprises a set of abducibles, which are hypotheses or assumptions used

as exceptions to the standard extensions of the predicates shaping the program's logic framework [33,38], viz.

$$\text{exception}_{p_1}, \ldots, \text{exception}_{p_j} \ (0 \leq j \leq k) \text{ being k an integer number}$$

This data captures crucial details, insights, or specific elements that are indispensable. Meanwhile, certain types of clauses serve as integrity constraints or invariants which supply the necessary context for understanding the universe of discourse, viz.

$$? \ (p_1, \ldots, p_n, \text{not } q_1, \ldots, \text{not } q_m) \ (n, m \geq 0)$$

The structuring of knowledge for computational interpretation and algorithm development for intelligent decision-making benefits from integrating Computational Collective Intelligence with diverse disciplines such as Knowledge Representation, Thermodynamics, and Mathematical Logic [39,40]. This integrated approach demonstrates remarkable flexibility and effectiveness across different scenarios, making it universally applicable to any case study [41,42]. The core of this approach is its interdisciplinary nature, which leverages the principles of thermodynamics as a metaphorical lens for AI performance and constraints [43]. This approach is not only novel but also highly adaptable, making it suitable for a variety of case studies [44]. For instance, the integration of KRR with thermodynamic concepts allows for a dynamic assessment of AI systems, focusing on energy efficiency and entropy, the key factors in determining system performance and sustainability [44]. Several other case studies can be referenced, particularly those involving, namely:

Complex Data Environments—In these cases, the approach's emphasis on energy efficiency (borrowed from thermodynamics) can guide the structuring of AI systems to handle and process large datasets more efficiently [45]. This is particularly relevant in fields like big data analytics and cloud computing, where managing computational resources effectively is crucial [46].

Decision-Making Systems—The incorporation of mathematical logic into the problem-solving framework enhances AI's decision-making capabilities. In case studies focused on autonomous vehicles or financial systems, where precision and reliability are paramount, the rigorous logical frameworks ensure that the AI's decisions are both sound and verifiable [46].

Dynamic and Evolving Systems—This approach's adaptability makes it ideal for applications in environments that are not static but require continuous learning and adaptation. Case studies in robotics or adaptive learning systems can benefit from this approach, as it supports the development of AI that can evolve and respond to changing conditions without human intervention [47].

Interdisciplinary Integration—The ability to integrate various disciplines ensures that the approach can be applied in a broad range of case studies, from healthcare to environmental science. This flexibility is essential for developing holistic AI solutions that consider multiple aspects of a problem, such as ethical considerations, sustainability, and technical feasibility [48].

3. The Role of Thermodynamics in Data Procurement and Judgement

In the rapidly advancing domain of data science, the groundbreaking method of applying thermodynamic concepts to data collection and analysis presents an innovative perspective on data comprehension and application. This approach interestingly draws comparisons between the principles of thermodynamics, particularly regarding energy and work, and the practices of gathering and analyzing data. While non-traditional, the belief is that such a comparison will pique the reader's interest. Additionally, the incorporation of AI, ML, and DT within this framework promotes sustainability and enhances citizen empowerment in a SC context. The application of thermodynamics to data collection and analysis signifies an extraordinary convergence of distinct disciplines, providing a

novel angle that could further strengthen the role of AI, ML, and DT in forging sustainable initiatives, thus reinforcing citizen engagement in the smart city infrastructure.

Enhancing Citizen Agency in Smart Urban Environments

Artificial intelligence, machine learning, digital transformation, and computational sustainability are pivotal in advancing smart city initiatives. These technologies greatly improve citizens' ability to interact with and impact their urban environments. AI and ML are instrumental in gathering and analyzing urban data, empowering people to navigate the complexities of city life. AI decodes complex patterns, from traffic circulation to air quality, enabling informed decisions. ML enhances this with predictive models that forecast urban developments, promoting a forward-thinking community. DT and CS are transformative, making vast data sets actionable through digital platforms, allowing citizens to access real-time updates and partake in civic engagement, from urban planning to energy use. This integration turns citizens from mere spectators into active contributors, revitalizing democratic engagement in cities. It lifts citizen involvement, leveraging data for civic engagement and empowerment. Indeed, the goal is to evolve the urban experience into a collaborative creation by its residents, thanks to AI, ML, DT, and CS. This interplay is reshaping urban life, creating a milieu for an informed, involved citizenry. The vision of an informed, proactive urban community is materializing as these tools lay the groundwork for an interactive, responsive urban existence. The influence of these technologies on citizen involvement in SCs invites further exploration, especially through entropic methods in KRR, which could further enhance the empowerment process. This leads to a critical inquiry:

> *How might entropic methodologies in KRR intensify citizen empowerment within the Smart City architype?*

Offering a clear-cut response to this question is complex, as it hinges on the particularities of the urban setting and the diversity of its population. Nevertheless, several possibilities can be considered. For instance, the pertinence of each answer may need to be adjusted to fit the specific scenario, suggesting that reactions should be customized to reflect the subtleties of the inquiry at hand.

4. Methodology

This section briefly summarizes the study design, data collection procedure, instruments employed, sample characteristics, and data analysis methods. It also touches upon the ethical considerations observed during the research.

4.1. Study Design

Technological advancements act as a driving force in various domains such as urban management, innovation, job creation, industry growth, and environmental sustainability, among others. However, there remains public apprehension regarding the role of the connections among AI, ML, DT, and CS, in the empowerment of citizens within a smart city framework. To tackle this challenge, evaluating the understanding and acceptance of these technologies is essential, which entails active involvement from the population. Therefore, this study aims to evaluate the perception of the Portuguese population regarding the role of the connections among of these technologies in a smart city framework. With this goal in consideration, a questionnaire was developed and distributed in Portugal to a cross-section of individuals, incorporating male and female genders and diverse ages. Addressing five key topics (artificial intelligence and machine learning awareness and usage, digital transformation perception and use, citizen empowerment and perception, and correlation perception), the questionnaire was structured to facilitate the application of the methodology proposed in [49] for transforming non-numeric information into numeric data.

4.2. Data Collection

The choice of a questionnaire survey method arose from a thorough examination of available techniques, with the decision bolstered by its simplicity and adaptability. Although questionnaire surveys may lack depth and context, they provide efficiency, standardization, and anonymity.

The questionnaire devised for this study was divided into two segments. The first segment aimed to gather sociodemographic information, encompassing details like age, gender, and educational background. The second segment delved into a series of statements exploring the core topics under investigation (i.e., artificial intelligence and machine learning awareness and usage (AI and ML—4); digital transformation perception and use statements (DT and US—5); citizen empowerment and perception statements (CE and PS—4); and correlation perception statements (CPS—4), for which participants were prompted to select the option(s) that align with their opinions on each statement. Furthermore, they were also requested to indicate the progression tendency of his/her answer, i.e., an increasing tendency (strongly disagree → strongly agree) or the opposing (strongly agree → strongly disagree) as shown in Figure 1.

PART II

For each statement tick with × the option(s) that best reflects your opinion. Please also indicate the progression tendency of your answer, i.e., an increasing tendency (Strongly Disagree → Strongly Agree) or the opposing (Strongly Agree → Strongly Disagree).

	Strongly Agree	Agree	Disagree	Strongly Disagree	Increasing Tendency	Decreasing Tendency
Artificial Intelligence and Machine Learning Awareness and Usage – 4 Items						
S1. I am familiar with the terms "Artificial Intelligence" and "Machine Learning".	☐	☒	☒	☐	☒	☐
S2. I have utilized tools or applications that incorporate artificial intelligence and/or machine learning.	☐	☒	☐	☐	☒	☐
S3. I use artificial intelligence and/or machine learning tools or applications, such as voice assistants or recommendation systems.	☐	☐	☒	☒	☐	☒
S4. I believe the artificial intelligence and/or machine learning applications I use have contributed to making my life easier or more efficient.	☐	☐	☐	☐	☐	☐
Digital Transformation Perception and Use Statements – 5 Items						
S5. I am familiar with the term "Digital Transformation".	☐	☐	☐	☒	☐	☒
S6. I can name several tools or services I use that have undergone digital transformation.	☐	☐	☒	☐	☒	☐
S7. I regularly use digital tools or platforms in my daily life.	☐	☐	☒	☐	☒	☐
S8. I am aware of the impact of digital tools/platforms on my life.	☒	☐	☐	☐	☒	☐
S9. I believe digital transformation has significantly impacted the way I interact with institutions.	☒	☐	☐	☐	☒	☐
Citizen Empowerment and Perception Statements – 4 Items						
S10. I am confident in my understanding of "Citizen Empowerment".	☐	☐	☐	☒	☒	☐
S11. I feel empowered as a citizen in my daily life.	☐	☐	☐	☐	☐	☐
S12. I can name specific instances or tools that have contributed to my feeling of empowerment.	☐	☒	☐	☐	☒	☐
S13. I am satisfied with my degree of empowerment.	☐	☐	☒	☐	☒	☐
Correlation Perception Statements – 4 Items						
S14. I believe there is a connection between the use of artificial intelligence, machine learning, and digital transformation and my level of empowerment as a citizen.	☐	☐	☒	☐	☒	☐
S15. I can share specific examples where I feel artificial intelligence, machine learning, digital transformation, and computational sustainability have contributed to my empowerment.	☐	☐	☐	☒	☒	☐
S16. I see potential benefits with the increased integration of artificial intelligence, machine learning, digital transformation, and computational sustainability in everyday life.	☐	☒	☒	☐	☐	☒
S17. I believe in the future role of artificial intelligence, machine learning, digital transformation, and computational sustainability in citizen empowerment.	☐	☐	☒	☐	☒	☐

Figure 1. The preferences expressed by participant one in response to the second part of the questionnaire.

Each core topic is crafted to gauge aspects related to the role of the connections among artificial intelligence, machine learning, digital transformation, and computational sustainability, in the empowerment of citizens within a smart city framework. The statements associated with each of the topics mentioned earlier can be found in Figure 1. The primary goal of the AI and ML—4 topic is to comprehend and scrutinize various critical elements regarding the participant's awareness of AI and ML. It focuses on how people perceive and utilize these technologies, as well as associated tools and services. The statements included in this topic endeavors to assess awareness, gauge usage levels, identify emerging trends, evaluate public perception, understand the overall impact, inform strategic decision-making, and guide the development of policies and regulatory frameworks. Regarding the topic DT and US—5, the expectation is that the researcher team will be able to extract significant understanding of the public's grasp on and opinions about DT, particularly its effects on daily life and interactions with different entities. This knowledge is likely to be instrumental in formulating strategies for communication and education related to DT. These strategies will aim to improve public awareness of DT, assess the degree of its implementation and user engagement, identify the impact as perceived by users, and measure the quality of interactions between individuals and organizations. Concerning the topic CE and PS—4, this set of statements is primarily designed to explore the public's sense of empowerment as individuals within society. It aims to determine the contributing factors to this sentiment, identify potential areas for enhancement, and collate data that may aid in the creation of more effective strategies to boost citizen empowerment. This includes examining the concept of "empowerment" from diverse perspectives, how individuals encounter it in their daily existence, pinpointing tools or instances that intensify this sensation, and assessing the level of empowerment individuals perceive across different facets of their life. Finally, the set of statements included in the topic CPS—4 aims to delve into the public's understanding of the relationship between cutting-edge technologies such as AI, ML, DT, and CS, and the empowerment of citizens. It seeks to explore the potential advantages and limitations, uncover opportunities, and gauge expectations for the future. The questionnaire is structured to gauge the perceived linkage between technology and empowerment, pinpoint moments where empowerment occurs, assess the perceived pros and cons, identify tools that could facilitate empowerment, and understand the anticipations for the future, just to name a few.

Unlike the descriptive nature of the responses in the first segment of the questionnaire, the subsequent segment uses a four-level Likert scale (i.e., strongly agree (4), agree (3), disagree (2), and strongly disagree (1)).

The questionnaires were administered monthly for a period of 6 months, spanning from January 2023 to June 2023. Each participant received a hard copy of the questionnaire in person. All 73 distributed questionnaires were returned, resulting in a 100% return rate. The questionnaire was answered anonymously, and all participants agreed to participate over a period of 6 months by completing the questionnaire monthly. The participants received a secret personal code when they first answered the questionnaire, enabling researchers to identify responses from the same participant across multiple instances.

4.3. Participants

The study comprised an opportunity sample of 73 participants who completed the questionnaire during the study period. The age of the participants ranged from 18 to 65 years (with a mean age of 39.6 years), with 53.4% being women and 45.6% men.

4.4. Qualitative Data Processing

The information obtained in the second segment of the questionnaire uses a four-level Likert scale (i.e., strongly agree (4), agree (3), disagree (2), and strongly disagree (1)). However, since the tendency of progression of the participant's response was also asked, the Likert scale can be expanded to consider seven levels:

Strongly Agree (4), Agree (3), Disagree (2), Strongly Disagree (1), Disagree (2), Agree (3), Strongly Agree (4)

The expanded Likert scale should be read either from left to middle, indicating a progression from strongly agree (4) to strongly disagree (1), or from middle to right, indicating a progression from strongly disagree (1) to strongly agree (4). The first reading suggests a shift towards a more negative perspective or a disagreement with the statements presented, whereas the second suggests a shift towards a more positive perspective or an agreement with the statements.

Following the methodological framework introduced in [49], the non-numeric information was transformed into numerical information. In accordance with this methodology, the z responses associated with each theme are visualized in a circle with a radius of $1/\sqrt{\pi}$. Within the circle, z sections are delineated, with a mark on the axis indicating each response option, as described in Section 5.

4.5. Ethical Aspects

The research was conducted in accordance with existing legal norms and ethical standards. All participants were informed about the research objectives and voluntarily agreed to take part by filling out the questionnaire.

5. Case Study

The role of the connections among AI, ML, DT, and CS in the empowerment of citizens within a smart city framework were examined at the individual level. Thus, Table 1 presents the responses of participant one to the second segment of the questionnaire during the study period, taking into account the expanded Likert scale. For example, for the AI and ML—4 topic at month 0 the answer to S1 was Disagree (2)—Agree (3), indicating a decrease in entropy, since there is an increasing tendency in his/her opinion. For S2, the answer was Agree (3), a fact that speaks for itself. For S3, the answer was Disagree (2)—Strongly Disagree (1), indicating an increase in entropy, since there is a decreasing tendency in his/her opinion. Finally, for S4 no options were marked, corresponding to a vague situation. In this case, although the values of the different forms of energy (i.e., exergy, vagueness, and anergy) are unknown, it is known that the bandwidth is the interval [0, 1].

Table 1. The answers of participant one to the topics artificial intelligence and machine learning awareness and usage (AI and ML—4), digital transformation perception and use statements (DT and US—5), citizen empowerment and perception statements (CE and PS—4), and correlation perception statements (CPS—4), over a six-month period.

Month	Topic	Statements	Expanded Likert Scale 7 Items *							
			4	**3**	**2**	**1**	**2**	**3**	**4**	**Vagueness**
0	AI and ML—4	S1					×	×		
		S2						×		
		S3			×	×				
		S4								×
	DT and US—5	S5				×				
		S6					×			
		S7					×			
		S8							×	
		S9							×	
	CE and PS—4	S10				×				
		S11								×
		S12						×		
		S13					×			

Table 1. *Cont.*

Month	Topic	Statements	Expanded Likert Scale 7 Items *							
			4	**3**	**2**	**1**	**2**	**3**	**4**	**Vagueness**
0	CPS—4	S14					×			
		S15				×				
		S15		×	×					
		S17					×			
1	AI and ML—4	S1					×			
		S2						×		
		S3				×	×			
		S4					×			
	DT and US—5	S5					×			
		S6						×		
		S7					×	×		
		S8						×		
		S9							×	
	CE and PS—4	S10				×				
		S11				×				
		S12						×		
		S13				×				
	CPS—4	S14					×	×		
		S15			×					
		S15					×	×		
		S17						×		
2	AI and ML—4	S1						×		
		S2							×	
		S3						×		
		S4				×				
	DT and US—5	S5					×			
		S6						×		
		S7						×		
		S8							×	
		S9							×	
	CE and PS—4	S10					×			
		S11					×			
		S12					×			
		S13				×				
	CPS—4	S14						×		
		S15			×					
		S15					×	×		
		S17						×		
3	AI and ML—4	S1						×		
		S2							×	
		S3						×	×	
		S4			×					
	DT and US—5	S5		×	×					
		S6					×	×		
		S7					×			
		S8						×		
		S9						×		
	CE and PS—4	S10					×			
		S11					×	×		
		S12				×				
		S13			×					

Table 1. *Cont.*

Month	Topic	Statements	Expanded Likert Scale 7 Items *							
			4	**3**	**2**	**1**	**2**	**3**	**4**	**Vagueness**
3	CPS—4	S14						×		
		S15		×	×					
		S15							×	
		S17							×	
4	AI and ML—4	S1						×	×	
		S2							×	
		S3							×	
		S4		×	×					
	DT and US—5	S5			×					
		S6						×	×	
		S7						×		
		S8						×		
		S9							×	
	CE and PS—4	S10					×	×		
		S11						×		
		S12					×			
		S13			×					
	CPS—4	S14						×	×	
		S15		×	×					
		S15							×	
		S17						×		
5	AI and ML—4	S1							×	
		S2							×	
		S3						×	×	
		S4		×						
	DT and US—5	S5			×					
		S6							×	
		S7						×	×	
		S8							×	
		S9							×	
	CE and PS—4	S10						×		
		S11						×		
		S12						×		
		S13		×	×					
	CPS—4	S14						×		
		S15		×						
		S15						×	×	
		S17						×		

* (1) Strongly Disagree, (2) Disagree, (3) Agree, (4) Strongly Agree.

The shapes in Figure 2 represent the visual interpretation of participant one's answers to the topics AI and ML—4, DT and US—5, CE and PS—4, and CPS—4, at month 0, for both the Best-Case Scenario (BCS) and the Worst-Case Scenario (WCS). In Figure 2, the dark areas symbolize exergy, representing high-energy states or useful energy, the grey areas indicate vagueness, suggesting uncertainty or areas of indeterminate energy states, and the white ones stand for anergy, or areas where energy cannot be harnessed for work [49–51].

The assessment of the areas shown in Figure 2, for the BCS and for the WCS are provided in Tables 2 and 3, respectively, for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

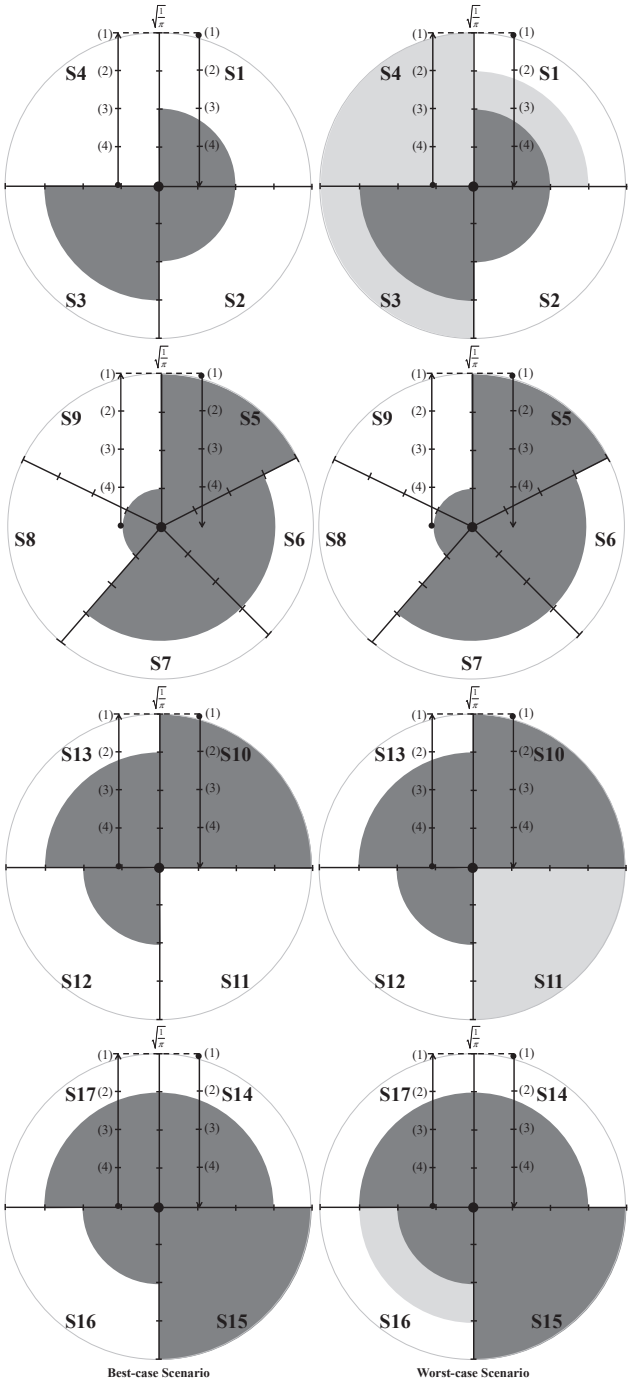


Figure 2. A pictorial reading of the participant one answers to the statements S1 to S17, in the best-case and worst-case scenarios at month 0. The dark, gray, and white colored areas correspond to exergy, vagueness, and anergy, respectively.

Table 2. Evaluation of exergy, vagueness, and anergy for artificial intelligence and machine learning awareness and usage (AI and ML—4) topic in month 0, in the best-case scenario, for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

Statement	AI and ML—4—Scale (4) (3) (2) (1)	AI and ML—4—Scale (1) (2) (3) (4)
S1	–	$exergy_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{0}$ $= -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{0} = \pi\left(0 - \left(\frac{2}{4}\sqrt{\frac{1}{\pi}}\right)^2\right) = 0.06$
	–	$vagueness_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\frac{2}{4}\sqrt{\frac{1}{\pi}}} = 0$
	–	$anergy_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.19$
S2	–	$exergy_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{0} = 0.06$
	–	$vagueness_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\frac{2}{4}\sqrt{\frac{1}{\pi}}} = 0$
	–	$anergy_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.19$
S3	$exergy_{S_3} = \frac{1}{4}\pi r^2\Big]_{0}^{\frac{3}{4}\sqrt{\frac{1}{\pi}}} = 0.14$	–
	$vagueness_{S_3} = \frac{1}{4}\pi r^2\Big]_{\frac{3}{4}\sqrt{\frac{1}{\pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\pi}}} = 0$	–
	$anergy_{S_3} = \frac{1}{4}\pi r^2\Big]_{\frac{3}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.11$	–
S4	$exergy_{S_4} = \frac{1}{4}\pi r^2\Big]_{0}^{0} = 0$	–
	$vagueness_{S_4} = \frac{1}{4}\pi r^2\Big]_{0}^{0} = 0$	–
	$anergy_{S_4} = \frac{1}{4}\pi r^2\Big]_{0}^{\sqrt{\frac{1}{\pi}}} = 0.25$	–

Similarly, by repeating the calculations presented above, it is possible to compute the values of the different forms of energy, i.e., exergy, vagueness, and anergy for all topics (i.e., AI and ML—4, DT and US—5, CE and PS—4, and CPS—4), for the various months during which the study was conducted, and for all participants. Furthermore, the Degree of Confidence (DoC) was computed according to Figure 3, using Equation (1), and the Quality of Information (QoI) was also computed using Equation (2), with all the findings presented in Table 4, for the BCS.

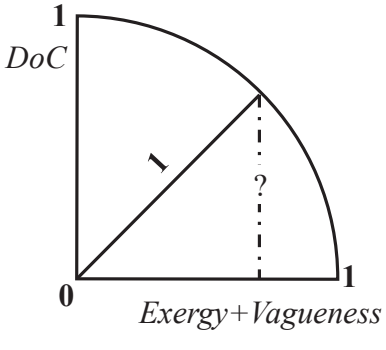


Figure 3. Evaluation of the Degree of Confidence (DoC) based on the values of exergy and vagueness.

Table 3. Evaluation of exergy, vagueness, and anergy for artificial intelligence and machine learning awareness and usage (AI and ML—4) topic in month 0, in the worst-case scenario, for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

Statement	AI and ML—4—Scale (4) (3) (2) (1)	AI andML—4—Scale (1) (2) (3) (4)
S1	–	$exergy_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{0} = 0.06$
	–	$vagueness_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\frac{3}{4}\sqrt{\frac{1}{\pi}}} = 0.08$
	–	$anergy_{S_1} = -\frac{1}{4}\pi r^2\Big]_{\frac{3}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.11$
S2	–	$exergy_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{0} = 0.06$
	–	$vagueness_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\frac{2}{4}\sqrt{\frac{1}{\pi}}} = 0$
	–	$anergy_{S_2} = -\frac{1}{4}\pi r^2\Big]_{\frac{2}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.19$
S3	$exergy_{S_3} = \frac{1}{4}\pi r^2\Big]_{0}^{\frac{3}{4}\sqrt{\frac{1}{\pi}}} = 0.14$	–
	$vagueness_{S_3} = \frac{1}{4}\pi r^2\Big]_{\frac{3}{4}\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0.11$	–
	$anergy_{S_3} = \frac{1}{4}\pi r^2\Big]_{\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0$	–
S4	$exergy_{S_4} = \frac{1}{4}\pi r^2\Big]_{0}^{0} = 0$	–
	$vagueness_{S_4} = \frac{1}{4}\pi r^2\Big]_{0}^{\sqrt{\frac{1}{\pi}}} = 0.25$	–
	$anergy_{S_4} = \frac{1}{4}\pi r^2\Big]_{\sqrt{\frac{1}{\pi}}}^{\sqrt{\frac{1}{\pi}}} = 0$	–

Table 4. Values of EXergy (EX), VAgueness (VA), ANergy (AN), Degree of Confidence (DoC), and the Quality of Information (QoI), regarding participant one, for all topics (i.e., AI and ML—4, DT and US—5, CE and PS—4, and CPS—4) for the study period, in the best-case scenario, for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

		Scale (4) (3) (2) (1)						Scale (1) (2) (3) (4)				
		EX	VA	AN	DoC	QoI		EX	VA	AN	DoC	QoI
Month 0	AI and ML—4_{4-1}	0.14	0	0.36	0.99	0.86	AI and ML—4_{1-4}	0.12	0	0.38	0.99	0.88
	DT and US—5_{4-1}	0.20	0	0	0.98	0.80	DT and US—5_{1-4}	0.25	0	0.55	0.97	0.75
	CE and PS—4_{4-1}	0.25	0	0.25	0.97	0.75	CE and PS—4_{1-4}	0.20	0	0.30	0.98	0.80
	CPS—4_{4-1}	0.31	0	0.19	0.95	0.69	CPS—4_{1-4}	0.28	0	0.22	0.96	0.72
	catch-all-clause	0.22	0	0.20	0.97	0.78	catch-all-clause	0.21	0	0.36	0.98	0.79
Month 1	AI and ML—4_{4-1}	–	–	–	–	–	AI and ML—4_{1-4}	0.33	0	0.67	0.94	0.67
	DT and US—5_{4-1}	–	–	–	–	–	DT and US—5_{1-4}	0.28	0	0.72	0.96	0.72
	CE and PS—4_{4-1}	0.75	0	0	0.66	0.25	CE and PS—4_{1-4}	0.06	0	0.19	1.0	0.94
	CPS—4_{4-1}	0.14	0	0.11	0.99	0.86	CPS—4_{1-4}	0.19	0	0.56	0.98	0.81
	catch-all-clause	0.44	0	0.06	0.83	0.56	catch-all-clause	0.22	0	0.54	0.97	0.78
Month 2	AI and ML—4_{4-1}	0.25	0	0	0.97	0.75	AI and ML—4_{1-4}	0.14	0	0.61	0.99	0.86
	DT and US—5_{4-1}	–	–	–	–	–	DT and US—5_{1-4}	0.24	0	0.76	0.97	0.76
	CE and PS—4_{4-1}	0.25	0	0	0.97	0.75	CE and PS—4_{1-4}	0.42	0	0.33	0.91	0.58
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.19	0	0.56	0.98	0.81
	catch-all-clause	0.19	0	0.06	0.98	0.81	catch-all-clause	0.25	0	0.57	0.95	0.75
Month 3	AI and ML—4_{4-1}	0.14	0	0.11	0.99	0.86	AI and ML—4_{1-4}	0.09	0	0.66	0.99	0.91
	DT and US—5_{4-1}	0.05	0	0.15	1.0	0.95	DT and US—5_{1-4}	0.26	0	0.54	0.97	0.74
	CE and PS—4_{4-1}	0.39	0	0.11	0.92	0.61	CE and PS—4_{1-4}	0.20	0	0.30	0.98	0.80
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.09	0	0.66	0.99	0.91
	catch-all-clause	0.16	0	0.14	0.98	0.84	catch-all-clause	0.16	0	0.54	0.98	0.84
Month 4	AI and ML—4_{4-1}	0.06	0	0.19	1.0	0.94	AI and ML—4_{1-4}	0.05	0	0.70	1.0	0.95
	DT and US—5_{4-1}	0.11	0	0.09	0.99	0.89	DT and US—5_{1-4}	0.12	0	0.68	0.99	0.88
	CE and PS—4_{4-1}	0.14	0	0.11	0.99	0.86	CE and PS—4_{1-4}	0.27	0	0.48	0.96	0.73
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.09	0	0.66	0.99	0.91
	catch-all-clause	0.09	0	0.14	1.0	0.91	catch-all-clause	0.13	0	0.63	0.99	0.87

Table 4. *Cont.*

		Scale (4) (3) (2) (1)						Scale (1) (2) (3) (4)				
		EX	**VA**	**AN**	**DoC**	**QoI**		**EX**	**VA**	**AN**	**DoC**	**QoI**
Month 5	AI and ML—4_{4-1}	0.06	0	0.19	1.0	0.94	AI and ML—4_{1-4}	0.05	0	0.70	1.0	0.95
	DT and US—5_{4-1}	0.11	0	0.09	0.99	0.89	DT and US—5_{1-4}	0.05	0	0.75	1.0	0.95
	CE and PS—4_{4-1}	0.06	0	0.19	1.0	0.94	CE and PS—4_{1-4}	0.19	0	0.56	0.98	0.81
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.14	0	0.61	0.99	0.86
	catch-all-clause	0.07	0	0.16	1.0	0.93	catch-all-clause	0.11	0	0.65	0.99	0.89

Similarly, by repeating the calculations presented above, it is possible to compute the values of the different forms of energy, i.e., exergy, vagueness, and anergy for all topics (i.e., AI and ML—4, DT and US—5, CE and PS—4, and CPS—4), for the various months during which the study was conducted, and for all participants. Furthermore, the Degree of Confidence (DoC) was computed according to Figure 3, using Equation (1), and the Quality of Information (QoI) was also computed using Equation (2), with all the findings presented in Table 4, for the BCS.

$$DoC = \sqrt{1 - (exergy + vagueness)^2} \tag{1}$$

$$QoI = 1 - (exergy + vagueness) \tag{2}$$

For both scales, i.e., ranging from strongly agree (4) to strongly disagree (1) and from strongly disagree (1) to strongly agree (4), the values of exergy, vagueness, and anergy presented in Table 4 are the sum of the respective areas. Therefore, in the case of the AI and ML—4 topic in month 0, in the best-case scenario, the value of exergy on the scale from strongly agree (4) to strongly disagree (1) is computed based on the values provided in Table 2.

$$exergy_{4-1} = exergy_{4-1_{S_3}} + exergy_{4-1_{S_4}} = 0.14 + 0 = 0.14$$

whereas for the scale strongly disagree (1) to strongly agree (4) is:

$$exergy_{1-4} = exergy_{1-4_{S_1}} + exergy_{1-4_{S_2}} = 0.06 + 0.06 = 0.12$$

Likewise, the values related to the different forms of energy, DoC, and QoI, for the BCS, were computed for the remaining participants, integrating a database, of which Table 4 represents only an excerpt, since it refers only to participant one. Program 2 describes the answers of participant one using the data provided in Table 4 for month 0.

Program 2. *A Logic Programming view of predicates AI and ML—4, DT and US—5, CE and PS—4, and CPS—4's extensions for the best-case scenario at month 0, for participant one.*

```
{
/* The sentences below state that the extension of predicates AI and ML—4_{4-1}, ..., cps—4_{1-4} in best-case scenario are based on explicitly
specified clauses and those that cannot be dropped */
    ¬ ai&ml − 4_{4-1} ( EX, VA, AN, DoC, QoI)
                                                ← not ai&ml − 4_{4-1} ( EX, VA, AN, DoC, QoI),
                                                  not exception_{ai&ml−4_{4-1}} ( EX, VA, AN, DoC, QoI)
    ai&ml − 4_{4-1} (0.14, 0, 0.36, 0.99, 0.86).
    ··· (the dots stand for the remaining predicates_{4-1} in Table 4)
}
{
    ¬ ai&ml − 4_{1-4} ( EX, VA, AN, DoC, QoI)
                                                ← not ai&ml − 4_{1-4} ( EX, VA, AN, DoC, QoI),
                                                  not exception_{ai&ml−4_{1-4}} ( EX, VA, AN, DoC, QoI)
    ai&ml − 4_{1-4} (0.12, 0, 0.38, 0.99, 0.88).
    ··· (the dots stand for the remaining predicates_{1-4} in Table 4)
}
```

Similarly, for the WCS, Table 5 presents, for participant one, the different forms of energy, DoC, and QoI for AI and ML—4, DT and US—5, CE and PS—4, and CPS—4, for the various months during which the study was conducted.

Table 5. Values of EXergy (EX), VAgueness (VA), ANergy (AN), Degree of Confidence (DoC), and the Quality of Information (QoI), regarding participant one, for all topics (i.e., AI and ML—4, DT and US—5, CE and PS—4, and CPS—4) for the study period, in the worst-case scenario, for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

		Scale (4) (3) (2) (1)						**Scale (1) (2) (3) (4)**				
		EX	**VA**	**AN**	**DoC**	**QoI**		**EX**	**VA**	**AN**	**DoC**	**QoI**
Month 0	AI and ML—4_{4-1}	0.14	0.36	0	0.87	0.50	AI and ML—4_{1-4}	0.12	0.08	0.30	0.98	0.80
	DT and US—5_{4-1}	0.20	0	0	0.98	0.80	DT and US—5_{1-4}	0.25	0	0.55	0.97	0.75
	CE and PS—4_{4-1}	0.25	0.25	0	0.87	0.50	CE and PS—4_{1-4}	0.20	0	0.30	0.98	0.80
	CPS—4_{4-1}	0.31	0.08	0.11	0.92	0.61	CPS—4_{1-4}	0.28	0	0.22	0.96	0.72
	catch-all-clause	0.22	0.17	0.03	0.91	0.61	catch-all-clause	0.21	0.02	0.34	0.97	0.77
Month 1	AI and ML—4_{4-1}	–	–	–	–	–	AI and ML—4_{1-4}	0.33	0.19	0.48	0.86	0.48
	DT and US—5_{4-1}	–	–	–	–	–	DT and US—5_{1-4}	0.28	0.06	0.66	0.94	0.66
	CE and PS—4_{4-1}	0.75	0	0	0.66	0.25	CE and PS—4_{1-4}	0.06	0	0.19	1.0	0.94
	CPS—4_{4-1}	0.14	0	0.11	0.99	0.86	CPS—4_{1-4}	0.19	0.16	0.41	0.94	0.65
	catch-all-clause	0.44	0	0.06	0.83	0.56	catch-all-clause	0.22	0.10	0.44	0.94	0.68
Month 2	AI and ML—4_{4-1}	0.25	0	0	0.97	0.75	AI and ML—4_{1-4}	0.14	0	0.61	0.99	0.86
	DT and US—5_{4-1}	–	–	–	–	–	DT and US—5_{1-4}	0.24	0	0.76	0.97	0.76
	CE and PS—4_{4-1}	0.25	0	0	0.97	0.75	CE and PS—4_{1-4}	0.42	0	0.33	0.91	0.58
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.19	0.08	0.48	0.96	0.73
	catch-all-clause	0.19	0	0.06	0.98	0.81	catch-all-clause	0.25	0.02	0.55	0.96	0.73
Month 3	AI and ML—4_{4-1}	0.14	0	0.11	0.99	0.86	AI and ML—4_{1-4}	0.09	0.05	0.61	0.99	0.86
	DT and US—5_{4-1}	0.05	0.06	0.09	0.99	0.87	DT and US—5_{1-4}	0.26	0.06	0.48	0.95	0.68
	CE and PS—4_{4-1}	0.39	0	0.11	0.92	0.61	CE and PS—4_{1-4}	0.20	0.08	0.22	0.96	0.72
	CPS—4_{4-1}	0.06	0.08	0.11	0.99	0.86	CPS—4_{1-4}	0.09	0	0.66	0.99	0.91
	catch-all-clause	0.16	0.04	0.11	0.97	0.80	catch-all-clause	0.16	0.05	0.49	0.97	0.79
Month 4	AI and ML—4_{4-1}	0.06	0.08	0.11	0.99	0.86	AI and ML—4_{1-4}	0.05	0.05	0.66	0.99	0.90
	DT and US—5_{4-1}	0.11	0	0.09	0.99	0.89	DT and US—5_{1-4}	0.12	0.04	0.64	0.99	0.84
	CE and PS—4_{4-1}	0.14	0	0.11	0.99	0.86	CE and PS—4_{1-4}	0.27	0.08	0.41	0.94	0.65
	CPS—4_{4-1}	0.06	0.08	0.11	0.99	0.86	CPS—4_{1-4}	0.09	0.05	0.61	0.99	0.86
	catch-all-clause	0.09	0.04	0.11	0.99	0.87	catch-all-clause	0.13	0.06	0.58	0.98	0.81
Month 5	AI and ML—4_{4-1}	0.06	0	0.19	1.0	0.94	AI and ML—4_{1-4}	0.05	0.05	0.65	0.99	0.90
	DT and US—5_{4-1}	0.11	0	0.09	0.99	0.89	DT and US—5_{1-4}	0.05	0.04	0.71	0.99	0.91
	CE and PS—4_{4-1}	0.06	0.08	0.11	0.99	0.86	CE and PS—4_{1-4}	0.19	0	0.56	0.98	0.81
	CPS—4_{4-1}	0.06	0	0.19	1.0	0.94	CPS—4_{1-4}	0.14	0.05	0.56	0.98	0.81
	catch-all-clause	0.07	0.02	0.15	0.995	0.91	catch-all-clause	0.11	0.04	0.62	0.985	0.85

The values related to the different forms of energy, DoC, and QoI, for the WCS, were also computed for the remaining participants, integrating a database, of which Table 5

represents only an excerpt, since it refers only to participant one. Program 3 describes the answers of participant one using the data provided in Table 5 for month 0.

Program 3. *A Logic Programming view of predicates AI and ML—4, DT and US—5, CE and PS—4, and CPS—4's extensions for the worst-case scenario at month 0, for participant one.*

```
{
/* The sentences below state that the extension of predicates AI and ML—4_{4-1}, ..., cps—4_{1-4} in worst-case
scenario are based on explicitly specified clauses and those that cannot be dropped */
    ¬ ai&ml − 4_{4−1} (EX, VA, AN, DoC, QoI)
                                        ← not ai&ml − 4_{4−1} (EX, VA, AN, DoC, QoI),
                                          not exception_{ai&ml−4_{4−1}} (EX, VA, AN, DoC, QoI)
    ai&ml − 4_{4−1} (0.14, 0.36, 0, 0.87, 0.50).
    ··· (the dots stand for the remaining predicates_{4-1} in Table 4)
}
{
    ¬ ai&ml − 4_{1−4} (EX, VA, AN, DoC, QoI)
                                        ← not ai&ml − 4_{1−4} (EX, VA, AN, DoC, QoI),
                                          not exception_{ai&ml−4_{1−4}} (EX, VA, AN, DoC, QoI)
    ai&ml − 4_{1−4} (0.12, 0.08, 0.30, 0.98, 0.80).
    ··· (the dots stand for the remaining predicates_{1-4} in Table 4)
}
```

Aiming to extract specific data or to conduct calculations based on records stored in the database (Tables 4 and 5), Program 4 is presented. It delineates predicates corresponding to each participant entry, establishes specific thresholds for categorization, and integrates rules for the calculation and categorization of averages based on these thresholds. Therefore, within Program 4, one may find:

- Facts (item_score, three arguments): Each fact denotes a score for a topic. The former argument is the topic code, the next one is the participant code, and the last one is the score.
- Retrieving Score (get_item_score, two arguments): This predicate returns the score for a particular participant using its code. The former argument is the participant code, whereas the last one is the score.
- Listing Participants Above a Specific Threshold (participants_above_threshold, two arguments): This predicate returns all participants with scores exceeding the specified threshold via the findall built-in predicate.
- Average Score (average_item_score, one argument): This predicate evaluates the average score for all participants via the built-in predicates.
- Maximum Score (max_item_score, one argument): This predicate returns the maximum score among all participants via the built-in predicate.
- Minimum Score (min_item_score, one argument): This predicate returns the minimum score among all participants via the built-in predicate.

Program 4. *An excerpt of the program based on the data provided in Table 4 for managing the participants' answers in the best-case scenario.*

```
% scores for the various topics for participant one at month 0 in the best-case scenario
AI and ML_4_4_1_exergy_score('AI and ML_4_4_1', 'Participant 1', 0.14).
AI and ML_4_1_4_exergy_score('AI and ML_4_1_4', 'Participant 1', 0.12).
AI and ML_4_4_1_vagueness_score('AI and ML_4_4_1', 'Participant 1', 0).
AI and ML_4_1_4_vagueness_score('AI and ML_4_1_4', 'Participant 1', 0).
AI and ML_4_4_1_anergy_score('AI and ML_4_4_1', 'Participant 1', 0.36).
AI and ML_4_1_4_anergy_score('AI and ML_4_1_4', 'Participant 1', 0.38).
AI and ML_4_4_1_doc_score('AI and ML_4_4_1', 'Participant 1', 0.99).
AI and ML_4_1_4_doc_score('AI and ML_4_1_4', 'Participant 1', 0.99).
```

```
AI and ML_4_4_1_qoi_score('AI and ML_4_4_1', 'Participant 1', 0.86).
AI and ML_4_1_4_qoi_score('AI and ML_4_1_4', 'Participant 1', 0.88).
. . . (the dots stand for the predicates DT and US_5_4_1; DT and US_5_1_4; CE and PS_4_4_1;
                                    CE and PS_4_1_4; CPS_4_4_1; and CPS_4_1_4 in Table 4)
% Retrieving the DoC score for a specific participant
get_ doc_score(ParticipantCode, Score): -
    doc_score(ParticipantCode, Score).
% Listing all participants with a DoC score above a specified threshold
participants_above_threshold(Threshold, ParticipantsCodes): -
    findall(ParticipantCode, (doc_score(ParticipantCode, Score), Score > Threshold),
                                                                    ParticipantsCodes).
% Calculating the average DoC score for all participants
average_ doc_score(Average): -
    findall(Score, doc_score(ParticipantCode, Score), Scores),
    sum_list(Scores, Total),
    length(Scores, Count),
    Count > 0, % Prevent division by zero
    Average is Total/Count.
% Finding the maximum DoC score among all participants
max_ doc_score(MaxScore): -
    findall(Score, doc_score(ParticipantCode, Score), Scores),
    max_list(Scores, MaxScore).
% Finding the minimum DoC score among all participants
min_ doc_score(MinScore): -
    findall(Score, doc_score(ParticipantCode, Score), Scores),
    min_list(Scores, MinScore).
% Retrieving the QoI score for a specific participant
get_ qoi_score(ParticipantCode, Score): -
    qoi_score(ParticipantCode, Score).
% Listing all participants with a QoI score above a specified threshold
participants_above_threshold(Threshold, ParticipantsCodes): -
    findall(ParticipantCode, (qoi_score(ParticipantCode, Score), Score > Threshold),
                                                                    ParticipantsCodes).
% Calculating the average QoI score for all participants
average_ qoi_score(Average): -
    findall(Score, qoi_score(ParticipantCode, Score), Scores),
    sum_list(Scores, Total),
    length(Scores, Count),
    Count > 0, % Prevent division by zero
    Average is Total/Count.
% Finding the maximum QoI score among all participants
max_ qoi_score(MaxScore): -
    findall(Score, qoi_score(ParticipantCode, Score), Scores),
    max_list(Scores, MaxScore).
% Finding the minimum QoI score among all participants
min_ qoi_score(MinScore): -
    findall(Score, qoi_score(ParticipantCode, Score), Scores),
    min_list(Scores, MinScore).
```

To illustrate the process of interacting with the database (Tables 4 and 5) using Program 4, several query examples are presented below. These examples emphasize the extraction of specific data or the execution of calculations using the scores:

```
% To obtain the QoI score for the 'Participant 1'
    ?- get_qoi_score('Participant 1', Score).
% To retrieve all participants with a QoI score above 0.75
    ?- participants_ qoi_score_above_threshold(0.75, ParticipantsCodes).
% To compute the average QoI score for all participants
    ?- average_qoi_score(Average).
```

```
% To retrieve the maximum QoI score among all participants
    ?- max_qoi_score(MaxScore).
% To retrieve the minimum QoI score among all participants
    ?- min_qoi_score(MinScore).
```

The sample queries illustrate the method of engaging with the database to extract specific data or to conduct calculations based on recorded scores. For instance, it is possible to monitor the fluctuations of a participant's entropic state (i.e., exergy + vagueness) and QoI across a six-month span, for the BCS (Figure 4) and for the WCS (Figure 5).

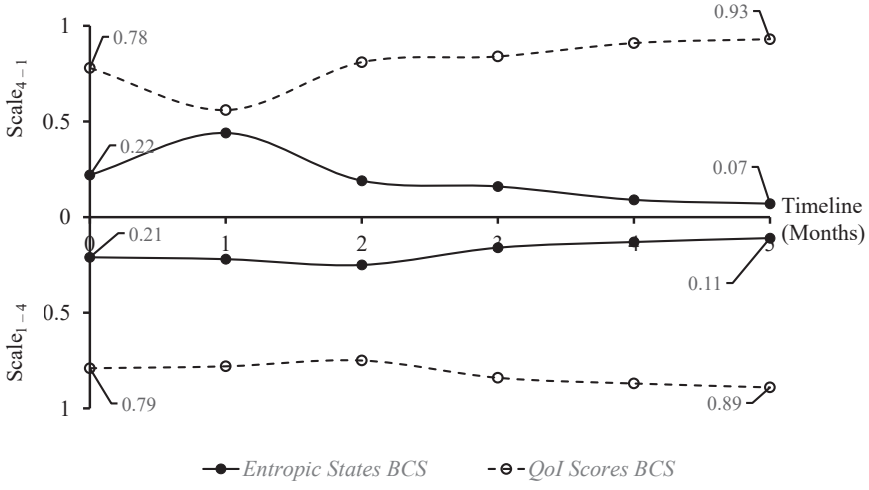


Figure 4. The evolution of participant one's Entropic State (ES) and Quality-of-Information (QoI) according to his/her answers within the Best-Case Scenario (BCS) over a six-month period for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

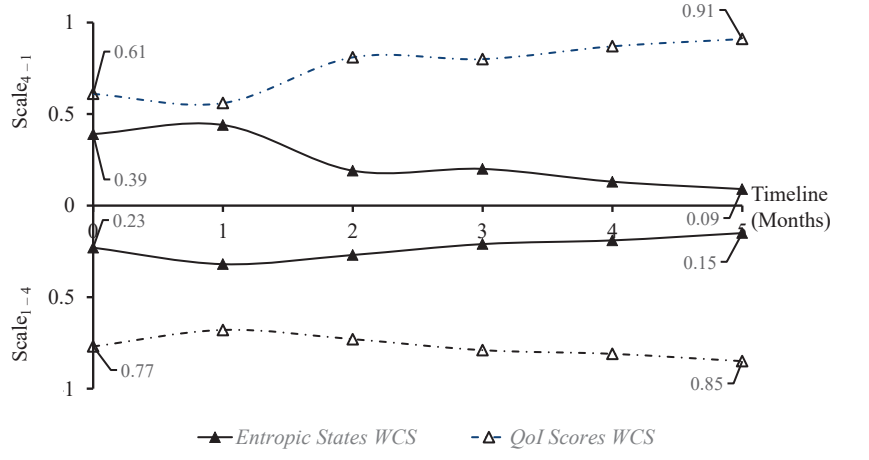


Figure 5. The evolution of participant one's Entropic State (ES) and Quality-of-Information (QoI) according to his/her answers within the Worst-Case Scenario (WCS) over a six-month period for both scales, i.e., from strongly agree (4) to strongly disagree (1), and from strongly disagree (1) to strongly agree (4).

When considering the significance of the results obtained from analyzing questionnaire responses over a six-month period within the context of SCs, the essay should explore the reliability and validity of these results to mitigate the influence of arbitrary factors. The study's novel approach to evaluating entropic efficiency in problem-solving scenarios suggests that the methodology is both rigorous and systematic, aiming to reflect the true relationship between the use of technology and citizen empowerment in SCs. Firstly, the significance of the results lies in the consistent trend observed over a sustained period (Figures 4 and 5). This is not a one-off measurement but a pattern emerging from data collected systematically over half a year, reducing the chance of random fluctuations or temporary biases affecting the findings. The geometric interpretation of citizens' entropic states provides a mathematical and visual representation of this evolution, reinforcing the argument that the observed changes are structured and significant. Secondly, the study uses a detailed and well-defined questionnaire, divided into specific topics (i.e., AI and ML—4, DT and US—5, CE and PS—4, CPS—4), to capture the multifaceted impact of technology. This comprehensive approach ensures that a wide range of factors influencing citizen empowerment are considered, which strengthens the argument against arbitrary influences.

The premise that lower entropy values denote order and higher values reflect disorder provides a quantifiable measure of understanding. Participants with structured and consistent responses display low entropy, suggesting a firm grasp of the concepts in question. Conversely, high values of entropy indicate varied and possibly chaotic responses, characteristic of less understanding or misconceptions. The extreme cases, i.e., individuals with no grasp of AI, ML, or DT presented high entropy, their responses lacking coherent structure. In contrast, the experts exhibited low entropy, their responses demonstrating coherent structure and a depth of understanding. The act of completing a questionnaire can raise awareness and pique interest in the topics covered. In this context, participants are nudged to reflect on concepts they might not have otherwise considered, potentially altering their level of understanding. It is essential to consider whether this heightened attention, spurred by the questionnaire itself, constitutes an artificial influence. Arguably, any form of measurement affects the subject being measured, a phenomenon well-documented in quantum mechanics as the observer effect. In the case of completing a questionnaire, this effect could manifest as an increase in the participants' awareness of the topics covered. The act of answering questions about these topics may compel individuals to reflect and thereby deepen their understanding, even if superficially. In the scenario where filling out a questionnaire leads to an enduring engagement with the concepts, fostering continued learning and curiosity, then the influence can be deemed constructive. However, if the impact is fleeting, dissipating soon after the questionnaire is completed, it could be considered artificial, i.e., a transient spike in awareness with no lasting educational value. The entropic approach employed in this study helps to gauge the quality of understanding that arises from this process. By examining the evolution of participants' entropic states over time, the researcher can ascertain whether the questionnaires have a lasting educational effect or merely a momentary one. In summary, the act of filling a questionnaire has the potential to both evaluate and influence the understanding of technical concepts. While there is a risk that the influence could be artificial, it ultimately depends on the persistence of the effect. In the present case, if the engagement with AI, ML, DT, and related concepts continues after the questionnaire filling, it can lead to a genuine enhancement of understanding. Therefore, it is not the immediate influence that should be under scrutiny but the long-term effects on the participants' comprehension and interest in these burgeoning fields of technology. In the future, it would be beneficial to regularly administer the surveys (e.g., every 4 months) after the initial 6-month period to better determine the persistence of the educational impact. Only through longitudinal studies is it possible to truly understand whether the questionnaires are simply catalysts for momentary awareness or effective tools for lasting awareness in complex domains such as AI, ML, and DT, just to name a few.

6. Result Analysis

This paper presents an inclusive study that delves into the integration and impact of AI, ML, DT, and CS within the context of SCs. A novel approach is proposed for evaluating entropic efficiency in problem-solving scenarios within AI, drawing parallels between the principles of thermodynamics and the organization of knowledge for computational interpretation. This approach aims to enhance the ability of AI systems to make intelligent decisions based on structured knowledge. Furthermore, the document explores the concept of citizen empowerment within SCs, arguing that the convergence of AI, ML, DT, and CS technologies provides residents with the tools to make informed choices, actively participate in community life, and adopt sustainable behaviors. It suggests that empowerment is pivotal for navigating the complexities of modern urban environments and underscores the potential of these technologies to reshape urban living. This research includes detailed questionnaires aimed at understanding public awareness, perception, and use of AI, ML, and DT technologies. It assesses the impact of these technologies on citizen empowerment and evaluates how they contribute to the creation of smart, sustainable urban ecosystems. Through the analysis of questionnaire answers over several months, the study presents a geometric interpretation of citizens' entropic states, offering insights into the evolving relationship between technology use and empowerment in smart cities. This analytical approach highlights the transformative potential of integrating AI, ML, DT, and CS into urban development strategies, emphasizing the importance of technology in enhancing citizen engagement, sustainability, and the overall quality of urban life.

When analyzing the results of this study, it becomes evident that the achieved outcomes are intertwined with various other research findings, connecting technological advances with enhancements in urban life. Indeed, the present findings align with several works, including Batty et al. [1] who conceptualized SCs as a convergence of digital networks and urban environments, and Allam and Newman [2], who underscored the transformative potential of technology in urban spaces. Additionally, it aligns with results obtained by Anthopoulos [3], who elucidates the interconnected nature of urban technology, thus underlining with the intertwined role of AI and ML as identified in this study. Furthermore, the insights drawn from Caragliu et al. [4] regarding the economic framework of SCs anticipate the economic implications of the current study's results. The implementation of the questionnaire over a period of six months is in line with the ideas presented by Kitchin [7] involving the notion of data-driven cities.

The longitudinal approach used in this study, marked by a geometric interpretation of citizens' entropic states, offers a novel perspective akin to the entropic frameworks presented by Neirotti et al. [10], who explore the informational structure of urban systems. Additionally, this study touches upon the themes of DT and its impact on governance and service delivery, contributing to the discussion initiated by Wirtz et al. [12,52]. These authors explore the idea of cities as platforms for innovation, a concept that the current study reinforces through its exploration of DT's transformative power. The current study also ventures into the realm of CS, a cornerstone of SCs, which echoes the environmental concerns addressed by Albino et al. [14]. The findings presented contribute to this ongoing discourse by showcasing how AI and ML can catalyze resource optimization and waste reduction, thus aligning with the sustainability goals put forth by Bokhari and Myeong [20] in their work on AI in SCs.

The current study contributes to a better understanding of the complexity of urban ecosystems through its innovative entropic approach, thereby complementing the research by Meijer and Bolíver [15], which examines the empirical and normative aspects of SCs. The application of entropy to KRR within the study could also augment the foundational work on KRR methodologies discussed by Lifschitz et al. [42]. In terms of citizen empowerment, the results presented in this study align with the work of Goldsmith et al. [16] where they advocate for the potential of data to enhance citizen participation.

The questionnaire used in this study mirrors this emphasis on engaging citizens, providing empirical data that reinforce the significance of AI, ML, DT, and CS in realizing

the vision of responsive and participatory urban spaces. By situating its methodology and results within these contexts, the current study not only reaffirms the findings of its predecessors but also paves the way for future studies. It encourages a deeper probe into the synergies between urban technology and societal benefits, advocating for a continuous dialogue between empirical findings and theoretical advancements. The richness of this comparison highlights the study's intricate relationship with the broader research landscape and its potential to contribute meaningful insights into the interdisciplinary study of SCs and technology's role in urban development and citizen empowerment. It is this interlacing of the study's findings with the work of other authors that propels the academic discussion forward, challenging and refining the understanding of SCs in the age of pervasive technology.

Although this study produced promising findings, it is essential to acknowledge certain limitations that impeded a deeper evaluation of the role of the connections among AI, ML, DT, and CS, in the empowerment of citizens within a smart city framework. The primary constraint stems from the sample size and its nature, i.e., an opportunity sample. By expanding the sample to encompass participants from all regions of Portugal, it becomes feasible to derive results that enable generalization across the entire Portuguese territory. Moreover, gathering additional data on the socio-demographic and socio-economic attributes of the participants will allow a more thorough examination of the factors that could impact the perceptions of the Portuguese population regarding the role of the connections among AI, ML, DT, and CS, in the empowerment of citizens within a smart city framework.

Finally, it is also important to address some possible criticisms/limitations by underscoring the research's intentional focus and the methodological underpinnings that guided the approach. The inherent nature of scholarly research in technology and urban development often carries a forward-looking perspective. This is not to overlook potential drawbacks or challenges but to explore and maximize the capabilities of emerging technologies for societal benefits. The study's positive stance reflects a proactive approach to problem-solving and innovation, a crucial element in the domains of AI, ML, DT, CS, and SCs. It aligns with the aspirational goals of these fields, which seek to harness technology for the greater good, optimize human life, and promote sustainable development. The "dark parts" of these concepts suggests a balanced view that encompasses potential risks, limitations, and negative implications. While this is indeed valuable for a comprehensive overview, the scope of any analysis is bound by its objectives. The optimistic character of the analysis serves a strategic purpose, i.e., to ideate, conceptualize, and propose solutions that can be iteratively refined and critically evaluated in future research. It is part of a strategic view where different studies contribute varying perspectives, eventually creating a balanced understanding. In summary, concerns about exploring the full spectrum of consequences in technological advancements are valid and the positive focus of the study is justified within its context and scope. The research in question serves as a constructive addition to the collective understanding of how emerging technologies can be leveraged for urban and societal betterment. Future research can and should explore the negative implications, as a natural progression of scholarly debate and as a necessary complement to this study.

7. Conclusions and Future Work

This work finishes off by synthesizing the impact of integrating artificial intelligence, machine learning, digital transformation, and computational sustainability in smart cities. It emphasizes that these technological pillars are crucial for transforming urban environments into more efficient, personalized, and sustainable habitats. This synergy between technology and urban management empowers citizens by equipping them with the knowledge and tools necessary for engaging actively in their communities and making sustainable choices. Looking forward, the text suggests a roadmap for future research that includes a deeper dive into the confluence of artificial intelligence, machine learning, digital transformation, and computational sustainability within urban ecosystems. It highlights the importance

of pioneering more refined methods for analyzing entropic efficiency in artificial intelligence, which could significantly improve the sophistication of knowledge representation and reasoning. This initiative aims to amplify the decision-making capabilities of artificial intelligence systems, making them more responsive and effective in urban settings. Furthermore, it advocates for an interdisciplinary approach to research, blending data science, urban studies, sustainability, and technology development to create smarter, more responsive cities. It proposes an investigation into new strategies for enhancing citizen empowerment through technology, aiming to better understand how these tools can foster greater civic participation, tackle urban issues, and promote sustainable development. This vision for future work underscores the potential of technology to revolutionize city living, making urban areas not just more technologically adept but also more inclusive and conducive to the well-being of all residents. By charting a course for future research and innovation, this study lays the groundwork for the next generation of smart cities that are sustainable, efficient, and empowering for citizens worldwide. Encouraging eco-innovative urban development is a complex endeavor fraught with limitations. However, by proactively identifying these limitations and implementing thoughtful mitigation strategies, cities can navigate these challenges. Mitigation is not merely about addressing current issues but anticipating future ones, ensuring that the smart city framework remains robust, inclusive, and adaptable. Ultimately, by joining innovation with foresight, the goal of sustainable and empowered urban living can be realized. Future research should focus on the longitudinal study of these mitigation strategies, assessing their effectiveness and refining them over time. It should also explore the intersectionality of eco-innovation with socio-economic factors, ensuring that the progress in urban development translates into tangible improvements in the quality of life for all citizens.

Author Contributions: Conceptualization, H.V., J.N. and J.R.; methodology, H.V., J.N., F.F.-R. and J.R.; software, V.A., J.R., F.F.-R. and J.N.; validation, V.A., J.N. and J.R.; formal analysis, V.A., H.V., J.R. and J.N.; investigation, V.A. and J.R.; writing—original draft preparation, V.A. and J.R.; writing—review and editing, H.V., J.N. and F.F.-R.; visualization, V.A. and J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PT national funds (FCT/MCTES, Fundação para a Ciência e Tecnologia and Ministério da Ciência, Tecnologia e Ensino Superior) through the projects UIDB/00319/2020; UIDB/50006/2020 and UIDP/50006/2020.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Batty, M.; Axhausen, K.W.; Giannotti, F.; Pozdnoukhov, A.; Bazzani, A.; Wachowicz, M.; Ouzounis, G.; Portugali, Y. Smart Cities of the Future. *Eur. Phys. J. Spec. Top.* **2012**, *214*, 481–518. [CrossRef]
2. Allam, Z.; Newman, P. Redefining the Smart City: Culture, Metabolism and Governance. *Smart Cities* **2018**, *1*, 4–25. [CrossRef]
3. Anthopoulos, L. Understanding the Smart City Domain: A Literature Review. In *Transforming City Governments for Successful Smart Cities*; Public Administration and Information Technology; Rodríguez-Bolívar, M., Ed.; Springer: Cham, Switzerland, 2015; Volume 8, pp. 9–21. [CrossRef]
4. Caragliu, A.; Del Bo, C.; Nijkamp, P. Smart Cities in Europe. *J. Urban Technol.* **2011**, *18*, 65–82. [CrossRef]
5. Angelidou, M. Smart cities: A conjuncture of four forces. *Cities* **2015**, *47*, 95–106. [CrossRef]
6. Bibri, S.; Krogstie, J. Smart sustainable cities of the future: An extensive interdisciplinary literature review. *Sustain. Cities Soc.* **2017**, *31*, 183–212. [CrossRef]
7. Kitchin, R. The real-time city? Big data and smart urbanism. *GeoJournal* **2014**, *79*, 1–14. [CrossRef]
8. Amović, M.; Govedarica, M.; Radulović, A.; Janković, I. Big Data in Smart City: Management Challenges. *Appl. Sci.* **2021**, *11*, 4557. [CrossRef]
9. Komninos, N.; Kakderi, C.; Panori, A.; Tsarchopoulos, P. Smart City Planning from an Evolutionary Perspective. *J. Urban Technol.* **2019**, *26*, 3–20. [CrossRef]

10. Neirotti, P.; De Marco, A.; Cagliano, A.C.; Mangano, G.; Scorrano, F. Current Trends in Smart City initiatives: Some Stylised Facts. *Cities* **2014**, *38*, 25–36. [CrossRef]
11. Chourabi, H.; Nam, T.; Walker, S.; Gil-Garcia, J.R.; Mellouli, S.; Nahon, K.; Pardo, T.A.; Scholl, H.J. Understanding Smart Cities: An Integrative Framework. In Proceedings of the 45th Hawaii International Conference on System Sciences; Sprague, R.H., Jr., Ed.; CPS—Conference Publishing Services: Los Alamitos, CA, USA, 2012; pp. 2289–2297. [CrossRef]
12. Wirtz, B.; Müller, W. A Meta-Analysis of Smart City Research and Its Future Research Implications. *Int. Public Manag. Rev.* **2021**, *21*, 18–39.
13. Batty, M. Big data, smart cities and city planning. *Dialogues Hum. Geogr.* **2013**, *3*, 274–279. [CrossRef] [PubMed]
14. Albino, V.; Berardi, U.; Dangelico, R.M. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *J. Urban Technol.* **2015**, *22*, 3–21. [CrossRef]
15. Meijer, A.; Bolívar, M. Governing the smart city: A review of the literature on smart urban governance. *Int. Rev. Adm. Sci.* **2016**, *82*, 392–408. [CrossRef]
16. Goldsmith, S.; Crawford, S. *The Responsive City: Engaging Communities through Data-Smart Governance*; Jossey-Bass: San Francisco, CA, USA, 2014.
17. Okwechime, E.; Duncan, P.; Edgar, D. Big data and smart cities: A public sector organizational learning perspective. *Inf. Syst. e-Bus. Manag.* **2018**, *16*, 601–625. [CrossRef]
18. Deep, G.; Verma, J. Embracing the future: AI and ML transforming urban environments in smart cities. *J. Artif. Intell.* **2023**, *5*, 57–73. [CrossRef]
19. United Nations Human Settlements Programme. *World Cities Report 2020: The Value of Sustainable Urbanization*; UN-Habitat: Nairobi, Kenya, 2020. [CrossRef]
20. Bokhari, S.; Myeong, S. Use of Artificial Intelligence in Smart Cities for Smart Decision-Making: A Social Innovation Perspective. *Sustainability* **2022**, *14*, 620. [CrossRef]
21. Rieder, E.; Schmuck, M.; Tugui, A. A scientific perspective on using artificial intelligence in sustainable urban development. *Big Data Cogn. Comput.* **2023**, *7*, 3. [CrossRef]
22. Ortega-Fernández, A.; Martín-Rojas, R.; García-Morales, V.J. Artificial intelligence in the urban environment: Smart cities as models for developing innovation and sustainability. *Sustainability* **2020**, *12*, 7860. [CrossRef]
23. Adel, A. Unlocking the future: Fostering human–machine collaboration and driving intelligent automation through industry 5.0 in smart cities. *Smart Cities* **2023**, *6*, 2742–2782. [CrossRef]
24. Hashem, I.; Usmani, R.; Almutairi, M.; Ibrahim, A.; Zakari, A.; Alotaibi, F.; Alhashmi, S.; Chiroma, H. Urban computing for sustainable smart cities: Recent advances, taxonomy, and open research challenges. *Sustainability* **2023**, *15*, 3916. [CrossRef]
25. Bellone, C.; Andreassi, F.; Naselli, F. How digitalization is affecting urban transformation. In *Smart Cities and Machine Learning in Urban Health*; Thomas, J.J., Geropanta, V., Karagianni, A., Panchenko, V., Vasant, P., Eds.; IGI Global: Hershey, PA, USA, 2022; pp. 69–83. [CrossRef]
26. Aggarwal, M. Human decision making through an entropic framework. *Expert Syst. Appl.* **2021**, *183*, 114926. [CrossRef]
27. Bratianu, C.; Bejinaru, R. Knowledge dynamics: A thermodynamics approach. *Kybernetes* **2020**, *49*, 6–21. [CrossRef]
28. Bratianu, C.; Vătămănescu, E.-M. The Entropic Knowledge Dynamics as a Driving Force of the Decision-Making Process. *Electron. J. Knowl. Manag.* **2018**, *16*, 1–12. [CrossRef]
29. Evans, B.P.; Prokopenko, M. A Maximum Entropy Model of Bounded Rational Decision-Making with Prior Beliefs and Market Feedback. *Entropy* **2021**, *23*, 669. [CrossRef] [PubMed]
30. Dincer, I.; Cengel, Y.A. Energy, Entropy and Exergy Concepts and Their Roles in Thermal Engineering. *Entropy* **2001**, *3*, 116–149. [CrossRef]
31. Deli, E.; Peters, J.; Kisvárday, Z. The Thermodynamics of Cognition: A Mathematical Treatment. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 784–793. [CrossRef] [PubMed]
32. Wenterodt, T.; Herwig, H. The Entropic Potential Concept: A New Way to Look at Energy Transfer Operations. *Entropy* **2014**, *16*, 2071–2084. [CrossRef]
33. Kakas, A.; Kowalski, R.; Toni, F. The role of abduction in logic programming. In *Handbook of Logic in Artificial Intelligence and Logic Programming*; Gabbay, D., Hogger, C., Robinson, I., Eds.; Oxford University Press: Oxford, UK, 1998; Volume 5, pp. 235–324. [CrossRef]
34. Pereira, L.; Anh, H. Evolution prospection. In *New Advances in Intelligent Decision*; Studies in Computational Intelligence; Nakamatsu, K., Phillips-Wren, G., Jain, L.C., Howlett, R.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 199, pp. 51–63. [CrossRef]
35. Neves, J.; Machado, J.; Analide, C.; Abelha, A.; Brito, L. The halt condition in genetic programming. In *Progress in Artificial Intelligence*; Lecture Notes in Computer Science; Neves, J., Santos, M.F., Machado, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4874, pp. 160–169. [CrossRef]
36. Neves, J. A logic interpreter to handle time and negation in logic databases. In Proceedings of the 1984 Annual Conference of the ACM on the 5th Generation Challenge; Muller, R., Pottmyer, J., Eds.; Association for Computing Machinery: New York, NY, USA, 1984; pp. 50–54. [CrossRef]
37. Kakas, A.C.; Mancarella, P. Abduction and Abductive Logic Programming. In Proceedings of the Eleventh International Conference on Logic Programming; Van Hentenryck, P., Ed.; MIT Press: Cambridge, CA, USA, 1994; pp. 18–19. [CrossRef]

38. Lucas, P. Quality checking of medical guidelines through logical abduction. In *Research and Development in Intelligent Systems XX*; Coenen, F., Preece, A., Mackintosh, A., Eds.; Springer-Verlag: London, UK, 2004; pp. 309–321. [CrossRef]
39. Maleszka, M.; Nguyen, N.T. Integration computing and collective intelligence. *Expert Syst. Appl.* **2015**, *42*, 332–340. [CrossRef]
40. Gupta, P.; Nguyen, T.N.; Gonzalez, C.; Woolley, A. Fostering collective intelligence in human-AI collaboration: Laying the groundwork for COHUMAIN. *Top. Cogn. Sci.* **2023**, *online version of record*. [CrossRef]
41. Brachman, R.; Levesque, H. *Knowledge Representation and Reasoning*; Morgan Kaufmann: Amsterdam, The Netherlands, 2004.
42. Lifschitz, V.; Morgenstern, L.; Plaisted, D. Knowledge Representation and Classical Logic. In *Handbook of Knowledge Representation*; Foundations of Artificial Intelligence; van Harmelen, F., Lifschitz, V., Porter, B., Eds.; Elsevier: Amsterdam, The Netherlands, 2008; Volume 3, pp. 3–88. [CrossRef]
43. Collins, C.; Dennehy, D.; Conboy, K.; Mikalef, P. Artificial intelligence in information systems research: A systematic literature review and research agenda. *Int. J. Inf. Manag.* **2021**, *60*, 102383. [CrossRef]
44. Xu, Y.; Liu, X.; Cao, X.; Huang, C.; Liu, E.; Qian, S.; Liu, X.; Wu, Y.; Dong, F.; Qiu, C.-W.; et al. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation* **2021**, *2*, 100179. [CrossRef]
45. Marinakis, V. Big Data for Energy Management and Energy-Efficient Buildings. *Energies* **2020**, *13*, 1555. [CrossRef]
46. OECD. Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers. 2021. Available online: https://www.oecd.org/finance/financial-markets/Artificial-intelligence-machine-learning-big-data-in-finance.pdf (accessed on 20 April 2024).
47. Harib, M.; Chaoui, H.; Miah, S. Evolution of adaptive learning for nonlinear dynamic systems: A systematic survey. *Intell. Robot.* **2022**, *2*, 37–71. [CrossRef]
48. Arbelaez-Ossa, L.; Lorenzini, G.; Milford, S.; Shaw, D.; Elger, B.; Rost, M. Integrating ethics in AI development: A qualitative study. *BMC Med. Ethics* **2024**, *25*, 10. [CrossRef] [PubMed]
49. Alves, V.; Miranda, J.; Dawa, H.; Fernandes, F.; Pombal, F.; Ribeiro, J.; Fdez-Riverola, F.; Analide, C.; Vicente, H.; Neves, J. An Entropic Approach to Technology Enable Learning and Social Computing. In *Machine Learning and Artificial Intelligence. Frontiers in Artificial Intelligence and Applications*; Kim, J.-L., Ed.; IOS Press: Amsterdam, The Netherlands, 2022; Volume 360, pp. 140–153. [CrossRef]
50. Neves, J.; Maia, N.; Marreiros, G.; Neves, M.; Fernandes, A.; Ribeiro, J.; Araújo, I.; Araújo, N.; Ávidos, L.; Ferraz, F.; et al. Entropy and Organizational Performance. In *Hybrid Artificial Intelligent Systems*; Lecture Notes in Computer Science; Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E., Eds.; Springer: Cham, Switzerland, 2019; Volume 11734, pp. 206–217. [CrossRef]
51. Neves, J.; Maia, N.; Marreiros, G.; Neves, M.; Fernandes, A.; Ribeiro, J.; Araújo, I.; Araújo, N.; Ávidos, L.; Ferraz, F.; et al. Employees balance and stability as key points in organizational performance. *Log. J. IGPL* **2021**, *30*, 664–678. [CrossRef]
52. Wirtz, D.; Tucker, A.; Briggs, C.; Schoemann, A.M. How and Why Social Media Affect Subjective Well-Being: Multi-Site Use and Social Comparison as Predictors of Change Across Time. *J. Happiness Stud.* **2021**, *22*, 1673–1691. [CrossRef]

algorithms

MDPI

Article

Equity in Transportation Asset Management: A Proposed Framework

Sara Arezoumand and Omar Smadi *

Department of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, IA 50011, USA; sarzmnd@iastate.edu
* Correspondence: smadi@iastate.edu

Abstract: Transportation asset management has historically overlooked equity considerations. However, recently, there has been a significant increase in concerns about this issue, leading to a range of research and practices aimed at achieving more equitable outcomes. Yet, addressing equity is challenging and time-consuming, given its complexity and multifaceted nature. Several factors can significantly impact the outcome of an analysis, including the definition of equity, the evaluation and quantification of its impacts, and the community classification. As a result, there can be a wide range of interpretations of what constitutes equity. Therefore, there is no single correct or incorrect approach for equity evaluation, and different perspectives, impacts, and analysis methods could be considered for this purpose. This study reviews previous research on how transportation agencies are integrating equity into transportation asset management, particularly pavement management systems. The primary objective is to investigate important equity factors for pavement management and propose a prototype framework that integrates economic, environmental, and social equity considerations into the decision-making process for pavement maintenance, rehabilitation, and reconstruction projects. The proposed framework consists of two main steps: (1) defining objectives based on the three equity dimensions, and (2) analyzing key factors for identifying underserved areas through a case study approach. The case study analyzed pavement condition and sociodemographic data for California's Bay Area. Statistical analysis and a machine learning method revealed that areas with higher poverty rates and worse air quality tend to have poorer pavement conditions, highlighting the need to consider these factors when defining underserved areas in Bay Area and promoting equity in pavement management decision-making. The proposed framework incorporates an optimization problem to simultaneously minimize disparities in pavement conditions between underserved and other areas, reduce greenhouse gas emissions from construction and traffic disruptions, and maximize overall network pavement condition subject to budget constraints. By incorporating all three equity aspects into a quantitative decision-support framework with specific objectives, this study proposes a novel approach for transportation agencies to promote sustainable and equitable asset management practices.

Keywords: equity; asset management; pavement decision-making; resource allocation

Citation: Arezoumand, S.; Smadi, O. Equity in Transportation Asset Management: A Proposed Framework. *Algorithms* **2024**, *17*, 305. https://doi.org/10.3390/a17070305

Academic Editor: Frank Werner

Received: 25 May 2024
Revised: 27 June 2024
Accepted: 5 July 2024
Published: 9 July 2024

1. Introduction

Transportation asset management significantly impacts various aspects of modern society, including mobility, health, safety, economic opportunities, and overall life quality. Decision-making is a very important step in asset management, as it involves a wide range of activities, including strategic planning, infrastructure design, treatment selection, and policy implementation. At the core of these impacts lie well-functioning transportation systems, which comprise infrastructure, vehicles, regulations, and user behavior, all of which interact to meet travel demand within a specific area [1]. Distress detection is one of the most critical topics in this domain. As transportation assets inevitably deteriorate over time due to various factors, including traffic loads and environmental conditions, efficient

and timely inspection is a crucial element of a successful infrastructure management system [2]. Effective asset management requires a meticulous decision-making process, which includes identifying current problems; establishing objectives, criteria, and constraints; creating solutions, such as new infrastructure construction; rehabilitating existing assets; and implementing management strategies.

The global cost of maintenance and repair for transportation infrastructure is remarkably high, amounting to hundreds of billions of USD annually. Numerous studies have consistently highlighted the detrimental impact of deteriorated transportation assets, such as pavements and bridges, on both public safety and economic productivity. A primary focus for researchers in this field is to minimize the substantial expenditures associated with inspection and maintenance efforts [3,4].

The complexity of decision-making intensifies when considering the multitude of stakeholder objectives, which often conflict. Highway agencies face the challenge of developing effective maintenance and rehabilitation (M&R) plans that accommodate limited funding and diverse stakeholder priorities. In transportation asset management, a wide range of stakeholders emerge, each representing distinct entities with varying objectives and vested interests. These stakeholders include local agencies overseeing highway infrastructure, highway users like commuters and businesses, environmental organizations advocating for ecological considerations, and local communities directly impacted by highway planning and operation. With such a diverse array of perspectives and priorities, decision-making processes require strategic coordination to address potential conflicts and maximize overall outcomes to provide an equitable system. Robust evaluation methods are crucial to achieving this balance by enabling a comprehensive assessment of trade-offs between various stakeholder priorities [5–7].

An effective transportation asset management program relies on various evaluation methods. These approaches differ in how they handle weight definition and analysis. In the field of transportation asset management, evaluation methods can be broadly categorized into parametric and non-parametric approaches, each with distinct implications for weight definition and analysis. Parametric methods, such as Stochastic Frontier Analysis (SFA), assume a specific functional form for the relationship between inputs and outputs, with weights often determined exogenously based on theoretical or empirical considerations. In contrast, non-parametric methods like data envelopment analysis (DEA) do not assume a specific functional form, deriving weights endogenously from the observed data. This allows for a flexible evaluation of efficiency without predefined weights. Multicriteria decision-making (MCDM) approaches often fall into the parametric category, with exogenously defined weights [8].

The United States has recently made significant investments in its highway infrastructure network to enhance sustainable development within the country. These extensive investments have resulted in the development of over 8 million lane miles of urban and rural roads throughout the country [9–13]. Sustainable development includes economic, environmental, and social aspects of decision-making. These dimensions may have conflicting relationships, requiring a balance based on decision-makers' preferences [14,15].

In the economic dimension, highway agencies aim to make cost-effective decisions for M&R plans within limited budgets [16]. According to Title 23 of the United States Code, the second aspect is concerned with the environmental dimension [17]. It considers greenhouse gas (GHG) emissions in M&R projects and aims to protect the natural environment while improving transportation systems. Environmentally friendly approaches may not always be the most cost-effective, and there should be a tradeoff between these dimensions [18–22]. Moreover, the Federal Highway Administration (FHWA) introduced a third aspect in the asset management practices of decision-making as social equity, which focuses on social responsibility and equity in the highway M&R process [23]. Developing sustainable M&R plans poses significant challenges for decision-makers, who must consider multiple conflicting objectives. Therefore, social equity is often overlooked as decision-makers mostly focus on the economic and/or environmental aspects [24–27]. By distributing

benefits fairly and equitably among all stakeholders, decision-makers can ensure that the transportation system meets the needs of all members of society, regardless of their background or location.

While there has been growing recognition of the importance of equity in transportation asset management, a limited number of studies have comprehensively explored its integration into decision-making processes. Existing research primarily focuses on individual aspects of equity, such as social equity or environmental justice, within the context of pavement maintenance optimization; however, these studies often rely on specific assumptions or lack consensus on how to best measure equity. For instance, in 2009, Thomopoulos et al. [28] proposed a method for evaluating the equity impacts of transportation infrastructure projects. The indicators used in this study are assumed to accurately reflect and measure the equity impacts of the project, even though there is no consensus on the most suitable indicators for assessing various aspects of equity. As a result, there is a possibility that the results may have been biased. In 2015, Boyles [29] conducted research to incorporate equity considerations into a network-level maintenance optimization problem. The study assumed the pavement deteriorates over time and can only be maintained through a single maintenance action, with the maintenance intervals for each facility identified as decision variables, subject to an annual budget constraint. This study made significant contributions to the field by incorporating equity considerations into a network-level pavement maintenance optimization problem. However, it is important to note that the assumptions made about the cost of maintenance, discount rate, and usage rates of facilities may not reflect real-world conditions.

In another study, in 2018, France-Mensah et al. [30] compared three methods including ranking-based, integer linear programming (ILP), and decision tree with needs-based allocation (DTN) [31] for the budget allocation of pavement M&R projects in a subset network in Texas. Following this research, they [32] conducted another study in 2019 to evaluate four different policies for incorporating social equity in highway M&R decision-making. For this purpose, they developed budget allocation models for each policy and used genetic algorithms (GA) to obtain policy-specific optimal solutions. Finally in 2022, Kothari et al. [23] proposed a sustainable pavement management plan, which considered all three aspects of sustainability, including economic, environmental, and social equity.

Gunathilaka and Amarasingha [33] also developed a framework based on the analytic network process (ANP) method for prioritizing pavement maintenance and rehabilitation projects considering social and economic factors, in Sri Lanka. The key factors in this research for equity evaluation were road user satisfaction, social equity, economic growth, environmental sustainability, road safety, technical feasibility, and project cost. Pairwise comparison interviews were conducted with nine transportation experts, and the obtained weights were converted into a matrix to obtain priorities.

Traditional approaches to incorporating equity often struggle to account for the inherent uncertainties in transportation planning. These uncertainties can stem from factors like population growth, traffic patterns, and economic fluctuations. As a result, achieving a truly balanced distribution of benefits across network users can be challenging. In 2017, Caggiani et al. [34] proposed a novel approach to address this limitation. They advocate for a paradigm shift towards incorporating flexible equity constraints represented by fuzzy sets. Fuzzy sets acknowledge and quantify these uncertainties, allowing for a more nuanced consideration of equity in decision-making. By introducing fuzzy programming, they achieved a more balanced distribution of benefits across network users, addressing both horizontal and vertical equity concerns.

Recent advancements in artificial intelligence and machine learning techniques present significant opportunities for optimizing decision-making processes in transportation asset management. Potential methods for optimization in this context include machine learning classifiers such as artificial neural networks (ANNs), random forest classifiers, and support vector machine (SVM) models. These models can address key optimization issues such as network capacity by enabling dynamic resource allocation, sample complexity by requiring

fewer labeled data for training, and computational complexity through efficient learning algorithms. Additionally, genetic algorithms, integer linear programming (ILP), and the analytic network process (ANP) have been successfully applied in previous studies for similar optimization problems, highlighting their potential applicability in this framework [35].

When it comes to incorporating equity into transportation and infrastructure systems, it goes beyond just research. The FHWA developed PlanWorks as a tool to facilitate collaborative decision-making during transportation planning and project development [36]. PlanWorks provides guidance on how and when to involve cross-disciplinary partners and stakeholder groups, encouraging transportation professionals to consider environmental equity throughout the entire planning and project development process. While the FHWA has made commendable efforts to incorporate environmental justice (EJ) considerations into transportation planning, there is a need to explore how EJ can be integrated with other equity aspects in transportation planning. The United States Department of Transportation (US DOT) has also taken a significant step towards prioritizing equity as a core strategic goal. The aim is to promote equity across the department's policies and programs, with the ultimate goal of reducing transportation-related inequities within the communities they serve [37].

In another practice, the US Government Accountability Office (GAO) conducted research to investigate uneven National Highway System (NHS) pavement conditions in communities with different characteristics. The findings indicate that 3.7 percent of the pavement in census tracts with higher underserved racial and ethnic populations is in poor condition, whereas only 1.3 percent of the pavement in census tracts with lower underserved racial and ethnic populations is in poor condition. Additionally, the investigation found a significant association between the prevalence of family poverty in a census tract and the condition of the National Highway System pavement in that area. Census tracts with higher rates of family poverty had a higher percentage of pavement in poor condition and a lower percentage in good condition [38]. The results highlighted the importance of considering underserved areas as a factor in the decision-making process and asset management planning.

According to the literature, most studies tried to consider different aspects of sustainability to develop an equitable asset management program, while there are a limited number of studies covering all three factors simultaneously. The objective of this study is to develop a prototype decision-support framework for allocating budgets for asset management projects by integrating all three decision parameters. The scope of implementation of the framework includes pavement maintenance, rehabilitation, and reconstruction projects.

Following this comprehensive literature review, Section 2 details the proposed decision-support framework. This framework involves a three-step process: defining objectives, analyzing data to identify underserved areas, and incorporating equity considerations into budget allocation for asset management projects, specifically focusing on pavement maintenance, rehabilitation, and reconstruction. The framework is followed by a case study analysis to investigate key factors influencing pavement conditions in underserved areas and to help define the underserved areas for the framework. Results are represented in Section 3. Section 4 discusses the potential limitations and biases of the framework, providing a critical evaluation of its robustness and applicability. Finally, Section 5 concludes with a summary of the key findings and implications, emphasizing the importance of integrating equity into transportation asset management decision-making.

2. Materials and Methods

The schematic representation of the framework is demonstrated in Figure 1. Furthermore, an analysis was conducted based on a case study to investigate the important factors that must be considered in the decision-making process. In other words, the focus is on identifying and highlighting the key factors that have a significant impact on equitable asset management planning. The case study serves as a real-life example, illustrating how considering certain factors to define an underserved area can lead to more inclusive and

equitable outcomes in transportation asset management. The implementation of the case study is discussed in the following paragraphs.

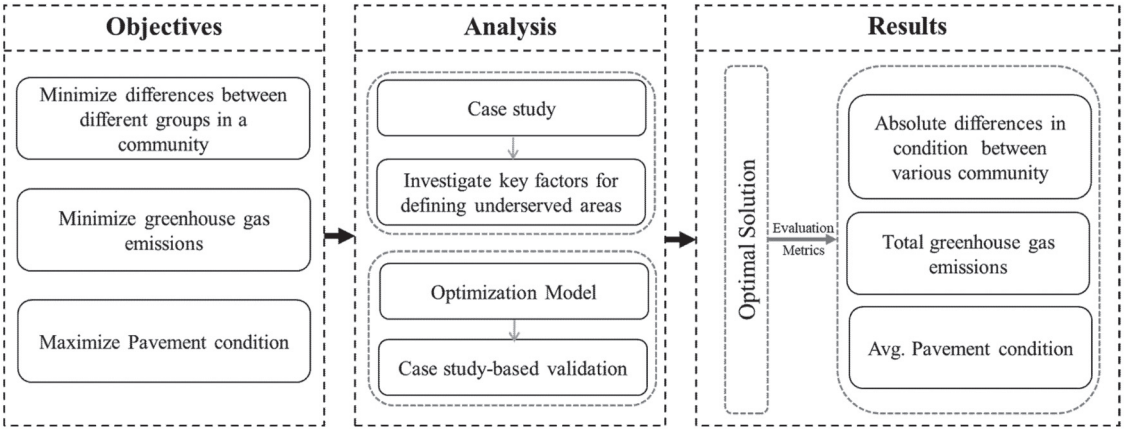


Figure 1. Schematic framework.

2.1. Framework

As shown in Figure 1, this framework consists of three steps. The first step involves defining the objective of the research. The second step deals with the analysis in which underserved areas are defined based on key factors, such as sociodemographics, using a case study approach. Then, the optimization model could be developed to select the best treatment option by optimizing the objectives. In the third and final step of the framework, the optimal solution obtained from the developed optimization model should be evaluated using the necessary evaluation metrics to demonstrate the efficiency and effectiveness of the chosen solution.

As previously discussed, the objective can be generally formulated as Equation (1), which is an optimization problem with three factors, including the economic, environmental, and social equity aspects, representing the three components of sustainable planning. The optimization technique proposed in this study to develop a sustainable pavement management plan is based on the literature [25].

$$M\&R\ option = f(SD, EN, EC) \quad (1)$$

where $M\&R\ option$ is the most optimum maintenance and rehabilitation activities; SD is sociodemographic factors; EN is environmental factors; and EC is economic factors.

Multiple factors such as type, measurement mechanism, and service provided can indicate the transportation asset performance with respect to a particular decision parameter. Regarding the focus of this paper, performance measurement is developed to assess pavements. In Equation (1), the first objective considers the social aspect and tries to enhance social equity by decreasing the condition gap between a disadvantaged group and others. The second objective focuses on the environmental aspect, with an emphasis on minimizing greenhouse gas (GHG) emissions [22]. Lastly, the third objective is about the economic aspect, which aims to reduce agency and road user costs by optimizing the road network condition [30]. A total number of 10 factors that affect the three decision parameters were used in this study based on the literature review, which is discussed in the following paragraphs.

2.2. Objectives

Pavement quality plays a significant role in ensuring smooth transportation flow and connectivity. Well-maintained pavements offer a smoother, safer, and more reliable

transportation infrastructure system, which positively impacts daily activities, mobility, and access to essential services, including jobs, education, healthcare, and business opportunities. Therefore, every member of society has the right to access a good-quality pavement system. However, underserved communities often face greater challenges with the quality of their transportation infrastructure. In simple terms, underserved communities have less access to well-maintained and reliable infrastructure compared to other regions.

Previous studies [38] suggest investigating various parameters' relationships with pavement condition (PC), such as population density, percentage of elderly people, race and ethnicity, traffic, education level, poverty rate, and unemployment rate. Using the factors, underserved areas were defined, as the results showed significant differences in pavement quality in areas with varying percentages of these factors. By minimizing differences in pavement quality among different groups in a community, transportation systems become more efficient and interconnected, benefiting not only underserved areas but the entire community.

The first aspect of a sustainable pavement management system is social equity. Therefore, addressing these disparities by minimizing objective 1 ensures that all community members have access to safe and well-maintained pavements, promoting social inclusivity and cohesion. The objective function can be formulated as Equation (2).

$$Minimize\ Objective_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} \left| PC_{UA,i} - PC_{OA,j} \right| \quad (2)$$

where

$Objective_1$: Minimize the absolute difference in pavement conditions between underserved and other areas.

$PC_{UA,i}$ = Pavement condition in underserved areas.

$PC_{OA,j}$ = Pavement condition in other areas.

n = Number of underserved areas.

m = Number of other areas.

constraints:

$PC_{UA,i}, PC_{OA,j}$: $specified\ ranges\ based\ on\ agency\ requirements$.

Budget constraints.

The environmental aspect of the pavement management system is defined as the second factor in a sustainable system. Reducing air pollution is a critical consideration in pavement planning, as air pollution, especially GHG emissions, significantly contributes to climate change and negatively affects air quality and public health. Therefore, another objective of this study is to improve air quality by selecting treatment options that produce less GHG emissions from two sources: traffic disruptions, which are influenced by traffic volume and treatment duration, and emissions from construction activities, which depend on the type of applied treatment (Equation (3)) [23,32].

$$Minimize\ Objective_2 = E_{C,k} + E_{T,k} \quad (3)$$

where

$Objective_2$: Minimize the total emissions from construction and traffic disruptions.

E_C = Emissions from construction activities in kg CO_2 equivalents.

E_T = Emissions from traffic disruptions in kg CO_2 equivalents.

constraints:

Budget constraints.

Emissions limits.

Treatment feasibility and schedules constraints.

The emissions from construction activities ($E_{C,k}$ are dependent on the specific type of M&R treatment performed on a given road section. This component accounts for emissions generated from material production, transportation of materials and equipment, as well as the construction processes themselves. Consequently, M&R activities that are more

extensive and intensive in nature will inevitably result in higher $E_{C,k}$ levels. This is due to the larger quantities of materials required and the longer durations of the construction projects, leading to prolonged usage of emissions-producing construction equipment. The $E_{C,k}$ can be quantified using the Equation (4) [23,32].

$$E_{C,k} = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{t=1}^{T} (gc_{itj}) \tag{4}$$

where

gc_{itj} = the average GHG emission due to a specific M&R treatment type i on section j at time t.

Different scenarios can be defined for construction activities, including doing nothing, applying preservation treatments, implementing rehabilitation treatments, performing reconstructions, or conducting maintenance. Preservations are proactive treatments that aim to protect the existing pavement and extend its service life, resulting in lower construction emissions compared to more extensive rehabilitation or reconstruction methods. Rehabilitation treatments involve partial pavement restoration, requiring moderate construction emissions but leading to reduced traffic disruptions compared to full reconstruction. Reconstruction treatments involving complete pavement replacement typically result in higher construction emissions due to the extensive nature of the work. Maintenance activities address minor pavement issues and prevent further deterioration, potentially causing lower construction emissions, but may lead to increased traffic disruptions if not appropriately planned.

Emissions due to traffic disruption ($E_{T,k}$) are another critical factor to consider. Traffic disruption also refers to the inconveniences and disturbances caused to traffic flow and road users during construction activities. It can lead to significant traffic congestion, resulting in increased fuel consumption and adverse effects on air quality. Lane closures, road diversions, and reduced road capacity can create bottlenecks, causing vehicles to move too slowly. Consequently, cars consume more fuel than usual, leading to higher emissions of GHG and other pollutants. Reduced traffic flow efficiency due to congestion can further exacerbate air pollution, as vehicles spend more time on the road emitting pollutants. To address these issues, pavement management systems should carefully consider planning and scheduling activities to minimize the impact on traffic flow.

The $E_{T,k}$ is a function of the traffic volume and the duration required to apply the M&R treatment. It can be quantified using the Equation (5) [23,32].

$$E_{T,k} = \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{t=1}^{T} (gd_{itj}) \cdot AADT_{jt} \tag{5}$$

where

$AADT_{jt}$ = the expected annual average daily traffic on section j in year t.

gd_{itj} = the marginal increase in GHG emissions of treatment i due to traffic disruptions on section j at time t for each unit of $AADT$.

Based on previous studies, the unit costs for the scenarios and their estimated GHG emissions from construction and traffic disruptions are included in Table 1, which serves as a schematic table for this [19,23,39].

The third aspect of sustainability is the economical point of view, in which pavement condition plays a vital role. Pavement condition assessment relies on various indicators or indices that reflect its overall state or level of service. These include pavement structural condition indicators such as the pavement structural number and distress score, as well as pavement functional condition indicators like the International Roughness Index (IRI) and riding quality. There are also indicators that combine both structural and functional conditions, such as the Pavement Condition Index (PCI) and Pavement Condition Rating (PCR). While different state Departments of Transportation (DOTs) use various pavement

condition indices based on their own policies and the data they collect in their pavement management systems, the PCI provides a standard way to assess pavement condition. It is a scored metric ranging from 0 to 100, where 0 indicates the most severe pavement deterioration and 100 represents the optimal pavement condition. In this study, the PCI is employed as the primary indicator for assessing pavement condition. The PCI is chosen due to its comprehensive coverage of all significant distress factors for pavements, including rutting, roughness, and cracking. This index is calculated using mathematical formulas, providing a complete picture of pavement condition and facilitating effective evaluation of pavement performance in different situations [40].

Table 1. Schematic Table for Environmental Analysis [23].

Treatments	Construction Emission (kg CO2eq = lane − mi)	Traffic Emission (kg CO2eq = lane − mi − AADT)	Agency Costs ($/lane − mile)
Do Nothing	0	0	0
Preservation	5700	0.5	37,000
Rehabilitation	28,000	2.5	300,000
Reconstruction	57,000	5.1	560,000
Maintenance	17,000	1.5	220,000

Over time, pavement sections deteriorate, leading to a reduction in their condition. However, implementing the appropriate M&R treatments at the right time could improve the pavement condition significantly, thereby reducing both agency and user costs. This is because well-maintained pavements exert less pressure on vehicles, resulting in reduced deterioration and increased vehicle service life, consequently lower maintenance costs for vehicle owners. Additionally, smoother pavements enhance fuel efficiency, reducing fuel consumption and associated expenses for drivers. On the other hand, deteriorated pavements can lead to increased vehicle operating costs, frequent repairs, and potential accidents due to uneven road surfaces.

Moreover, for agencies responsible for maintaining and managing road networks, the condition of pavements directly impacts their operational costs. Regular and timely maintenance of pavements can prevent minor issues from escalating into more severe problems, thereby reducing the need for expensive repairs and reconstruction. By optimizing the pavement condition through appropriate M&R strategies, agencies can effectively extend the service life of the pavement, maximizing their return on investment. On the other hand, neglecting pavement conditions can lead to premature failure, requiring costly emergency repairs and increasing the burden on the agency's budget.

Therefore, maximizing the pavement condition, as demonstrated in Equation (6), is an important objective from an economic efficiency standpoint. This not only translates to cost savings for both users and agencies over the long term but also ensures that budget allocations are utilized effectively by prioritizing the sections that require immediate attention based on their current condition and expected performance.

$$Maximize\ Objective_3 = \frac{1}{T} \times \sum_{t=1}^{T} PC_t \tag{6}$$

where

$Objective_3$: Maximize the average pavement condition over a given period.
PC_t = Pavement Condition at time t.
T = Total period of analysis.
constraints:
Pavement condition limits.
Treatment feasibility and schedules constraints.
Budget constraints.

2.3. Analysis

In the analysis section of this framework, the identification of underserved areas relies on sociodemographic key factors, including age, population density, poverty rate, and other mentioned variables. Employing a case study approach, the authors investigate whether there is a lack of equity in the decision-making process by assessing if certain communities or regions are suffering from worse pavement conditions in comparison to others.

Case Study

A comprehensive dataset was used as a case study in this research. The data was publicly available, containing sociodemographic data for California's Bay Area, sourced from the Office of Environmental Health Hazard Assessment (OEHHA) (The Office of Environmental Health Hazard Assessment 2023), alongside 20,764 miles of local pavement condition data obtained from the metropolitan transportation commission (MTC) website [41]. The dataset contains 1584 census tracts and is used as a resource to explore which significant factors should be considered in the decision-making process for pavement management and define underserved areas. Table 2 summarizes the dataset used in the research.

Table 2. California's Bay Area data summary.

Data	Description
Pavement Condition	20,764 miles of local roads Source: Metropolitan Transportation Commission (MTC) website
Sociodemographic	1584 census tracts Source: Office of Environmental Health Hazard Assessment (OEHHA)

According to this dataset, 43% of the street pavement (length) in the Bay Area is in excellent condition, with a pavement condition index (PCI) of 80 or higher. Moreover, 28% falls under the category of good condition, with a PCI ranging from 79 to 60. Furthermore, 9% of the pavement is considered to be at risk, with a PCI between 59 and 50, while 20% is classified as poor condition, with a PCI of 49 or lower. Figure 2 represents the distribution of pavement condition across the California Bay Area.

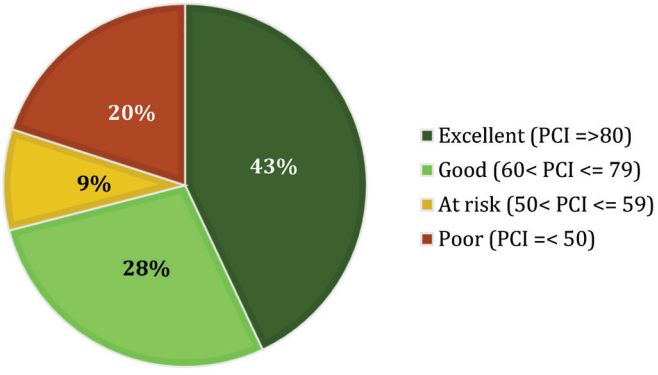


Figure 2. Pavement condition distribution in California's Bay Area.

According to the sociodemographic data, the total population in the area is around 7,758,000, with only 7.5% of individuals over 25 years old having less than a high school education, almost 60% of the population are not classified as white, and almost 14% are aged 65 and older. The dataset was then analyzed to investigate any relationships between pavement condition and various factors, including educational level, traffic, air

quality, race and ethnicity, age, income level, and population density. Initially, a non-parametric approach was employed to identify key factors influencing pavement quality in underserved areas. This exploratory phase prioritizes data-driven insights, allowing for a flexible analysis of efficiency without predefined weights. This approach is particularly well suited for uncovering complex relationships between various sociodemographic factors and pavement condition.

A statistical analysis was initially conducted, followed by the application of the Lasso (least absolute shrinkage and selection operator) regression method for more precise predictions and a better understanding of the impact of these factors on pavement condition. The detailed results are presented in Tables 3 and 4 in the result section.

The second step of analysis in this framework involves developing an optimization model that can solve Equation (1). This model for selecting maintenance and rehabilitation (M&R) options transitions to a parametric approach. The specific objective functions will be defined to quantify social equity, environmental impact, and economic efficiency. The weights for these objectives are determined exogenously, based on the established literature, policy considerations, and sustainability criteria.

The proposed framework incorporates objective constraints and regulations into the optimization process to ensure the proposed solutions are not only optimal but also feasible within the context of existing regulations and practical limitations. These constraints are derived from federal and state-level guidelines, standards, and regulations related to social equity, environmental justice, and sustainability considerations in transportation asset management. For instance, guidelines from the Federal Highway Administration (FHWA) and the Environmental Protection Agency (EPA) can be integrated as constraints to ensure compliance with environmental justice principles, air quality standards, and other relevant regulations. Additionally, state-level regulations and local ordinances specific to the region under consideration can be included as constraints, addressing factors such as minimum accessibility requirements for underserved communities, maximum allowable emissions levels, or specific criteria for defining and prioritizing disadvantaged areas. The constraints may also include budget limitations, an acceptable pavement performance range, and limitations on the frequency of M&R treatments applied to a pavement section. By incorporating these regulatory constraints, the framework ensures that any proposed solution adheres to established guidelines and standards, reducing subjectivity and distortion due to stakeholder perceptions or regional peculiarities. The final step in this framework involves reporting the results and defining evaluation metrics to assess the efficiency of the model based on a case study dataset.

2.4. Excepted Output

In the third and final step of the framework, the optimal solution obtained from the developed optimization model should be reported. However, it is worth noting that this does not mark the conclusion of the process; evaluation metrics are necessary to demonstrate the efficiency and effectiveness of the optimum solution. These evaluation metrics play a pivotal role in measuring the equity in the optimized version of pavement management strategies and their alignment with the defined objectives. One of the evaluation metrics involves calculating the absolute difference in pavement conditions between underserved areas and other regions. This metric helps determine whether the optimization model successfully addressed the objective of minimizing disparities in pavement quality among different communities. By quantifying the absolute difference, the extent to which the pavement management strategies improved conditions in underserved areas compared to other areas can be assessed.

In addition, evaluating the total amount of GHG emissions associated with pavement management activities and traffic disruption is essential for assessing the environmental impact of the optimization model. By quantifying and minimizing these emissions, the transportation agency can make significant contributions to its environmental sustainability goals and reduce the carbon footprint of the pavement management program. Another crit-

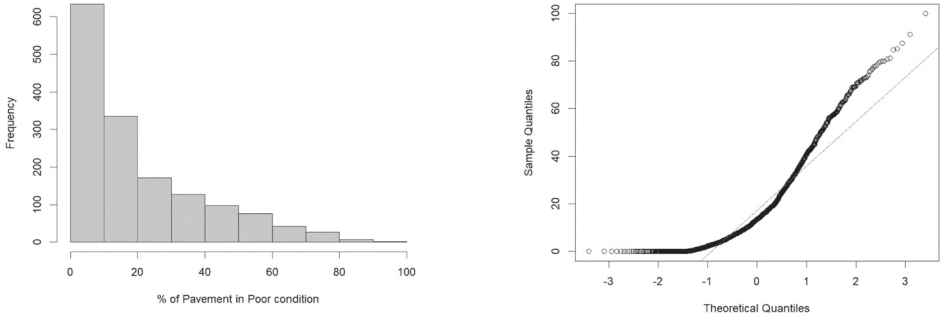
ical evaluation metric is the average pavement condition. This metric serves as a reflection of the overall effectiveness of the optimization model in achieving its primary objective, which is to maximize the pavement condition. A higher average pavement condition indicates that the model successfully improved the overall quality of the road network.

2.5. Proposed Optimization Algorithm

The pseudocode in Appendix A (Algorithm A1) outlines the proposed framework for optimizing pavement management decisions. It defines three objective functions: minimizing disparity in pavement quality between underserved and other areas minimizing greenhouse gas emissions and maximizing overall pavement condition over a specific time period. Additionally, two constraints are enforced: staying within budget and ensuring all pavements meet a minimum acceptable condition after treatment. The code could generate all possible treatment plans, filter out those violating the constraints, evaluate the remaining plans based on the objectives, and select the optimal plan according to a user-defined selection process.

3. Results and Discussion

In the first step of the analysis, the Shapiro–Wilk test was conducted to investigate whether the dataset followed a normal distribution. The Shapiro–Wilk test is a hypothesis test designed to assess whether a dataset follows a normal distribution. This test examines data from a sample, operating under the null hypothesis that the dataset is normally distributed. A high *p*-value suggests that the dataset conforms to a normal distribution, whereas a low *p*-value indicates a departure from normal distribution. According to the results (Figure 3), the data did not follow a normal distribution. Therefore, non-parametric tests were employed for further analysis.



(**a**) Histogram (**b**) Normal Q-Q plot

Figure 3. Histogram and normal QQ plot of pavement condition data.

As mentioned in the methodology, the dataset used in this study contains a wide range of potential factors, including age, educational level, poverty rate, race and ethnicity, population density, traffic, and air quality that could impact pavement condition. The Kruskal–Wallis test was utilized to analyze any significant relationships between the aforementioned factors and pavement conditions. This non-parametric test was selected due to the normality assumption of the dataset is violated. The Kruskal–Wallis test does not require data to be normally distributed, ensuring a more robust analysis. Results are presented in Table 3.

Table 3. Kruskal–Wallis test result.

	Poverty Rate	Air Quality	Traffic	Education Level	Race and Ethnicity	Population	Age
***p*-value**	0.001	0.0114	0.489	0.141	0.495	0.523	0.491

According to the results, the *p*-value for the poverty rate and air quality is less than 0.05, indicating a statistically significant relationship between these factors and pavement conditions. This suggests that the pavement condition may vary in areas with different rates of poverty and varying air quality. In other words, there is a significant difference in pavement conditions between areas with varying poverty rates and air quality levels. This highlights the importance for decision-makers to consider these factors in their management process to promote equity in their pavement management system.

However, factors such as population density, traffic, education, percentage of non-white population, and age, with *p*-values greater than 0.05, did not show a significant relationship with pavement condition in this specific dataset, which contradicts the findings in the Government Accountability Office (GAO) report [38]. This finding could be due to the fact that some agencies in the Bay Area have been considering race and ethnicity in their pavement management systems since 2017 [42]. However, further analysis needs to be conducted in individual areas to evaluate the effectiveness of that equity consideration. Currently, the Bay Area is being analyzed as a single data source. The results have also revealed a concerning observation regarding areas with higher poverty rates and areas with different ranges of air quality.

There are several machine learning methods for feature selection, with Lasso regression being one of the most common ones [9]. It can select useful features while discarding useless or redundant ones. In Lasso regression, discarding a feature will set its coefficient equal to 0. Therefore, in the next step, the Lasso regression method was applied to the dataset to achieve more precise results and determine the type of correlation, whether it is positive or negative. The results are shown in Table 4.

Table 4. Lasso regression method coefficient.

	Poverty Rate	Air Quality	Traffic	Education Level	Race and Ethnicity	Population	Age
Coefficient	0.10	0.27	0.00	0.00	0.00	0.0	0.01

According to the results, air quality showed a coefficient of 0.27, indicating a positive relationship with the percentage of pavement in poor condition. This suggests that areas with worse air quality correspond to a higher percentage of pavement in poor condition. Similarly, the poverty rate showed a coefficient of 0.1, supporting the observation that tracts with higher poverty rates tend to have a higher percentage of pavement in poor conditions. This implies that these two factors should be used as key indicators for defining underserved areas and should be incorporated as equity factors in the decision-making process. Decision-makers should consider these areas as underserved when allocating budgets or making decisions on M&R treatments if they want to establish an equitable management system.

On the other hand, the coefficients of other factors, such as race and ethnicity, educational level, population density, and traffic were aligned with the statistical results, showing coefficients of 0, indicating no significant relationship between these factors and pavement conditions in the dataset. The only contradictory finding was related to the factor of age. While the statistical analysis indicated that areas with a percentage of people older than 65 years had no correlation with pavement condition, the Lasso regression showed a slight increase in the percentage of pavement in poor condition, with a coefficient of 0.01.

This trend is further highlighted in Figures 4 and 5. Figure 4 demonstrates that tracts with higher poverty rates tend to have a greater percentage of pavement in poor conditions compared to areas with lower poverty rates. This indicates that these areas are more neglected or receive less attention for budget allocation for M&R projects, emphasizing the need for some equity adjustment in the decision-making process. Moreover, Figure 5 reveals that tracts with poor air quality mostly showed a higher percentage of pavement in poor condition compared to areas with better air quality. There are several possible explanations for these observations.

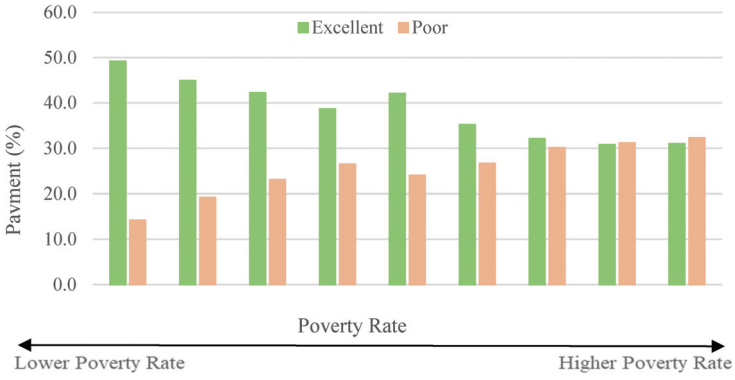


Figure 4. Pavement condition variation by poverty rate.

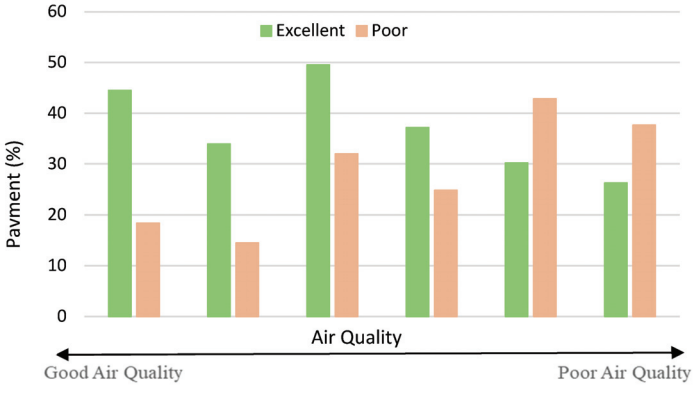


Figure 5. Pavement condition variation by air quality.

An important factor that could explain these findings is the land use type of the tracts, whether they are industrial or residential areas. Typically, air quality in industrial areas is lower compared to residential areas due to increased emissions and pollutants associated with industrial activities. Therefore, any potential relationship between the land use type and pavement condition should be investigated in future studies. Consequently, there is a possibility that the land use type should also be considered as another factor in defining underserved areas during the decision-making process. Additionally, the poor pavement condition itself could play a significant role in environmental issues. The presence of distressed pavement leads to frequent interruptions and traffic delays, which result in increased fuel consumption, higher emissions, and carbon footprint, which is one of the most important contributing factors in environmental equity assessments.

In conclusion, both the statistical analysis and Lasso regression results highlight air quality and poverty rate as the most influential factors for defining underserved areas, which might be considered in an equitable pavement management planning. The statistical analysis revealed that these factors have p-values less than 0.05, signifying their statistical significance. Moreover, the Lasso regression revealed non-zero coefficients, further supporting their importance in the decision-making process.

4. Potential Limitation and Biases

When proposing a framework that integrates social equity, environmental equity, and economic considerations into pavement management systems, it is crucial to recognize several significant limitations and potential biases. Firstly, the availability and quality of data related to social, environmental, and economic factors may vary across regions,

potentially constraining the framework's generalizability. This introduces a possible bias, where regions with more robust data collection infrastructure might be overrepresented in assessing the framework's effectiveness.

Another critical consideration involves the inherently subjective nature of assessing social and environmental equity. Divergent views among stakeholders on what constitutes equitable outcomes can introduce biases in decision-making. Additionally, economic conditions are dynamic, and the framework's reliance on economic factors may face challenges in adapting to changing circumstances. Economic biases may emerge if the framework lacks flexibility to accommodate variations in economic conditions.

Furthermore, the framework may not fully account for cultural and regional differences, posing limitations in its applicability in diverse contexts and potentially introducing cultural biases. Temporal dynamics also pose a challenge, as the framework may not adequately address long-term changes in social, environmental, or economic factors, limiting its relevance over time.

The effectiveness of the framework may also depend on stakeholder engagement, and challenges in obtaining meaningful participation could hinder successful implementation, potentially introducing biases in decision-making processes, especially if certain perspectives are underrepresented. Addressing these limitations through careful consideration, sensitivity analyses, and transparent decision-making processes is crucial for ensuring the robustness and applicability of the proposed framework in pavement management systems.

Finally, it is important to note the potential for endogeneity in these types of analyses. Endogeneity can arise when there is a bidirectional relationship between the variables considered, meaning that the causal relationship may not be straightforward. In situations where endogeneity is present, the framework's ability to disentangle and accurately model the causal relationships may be compromised. This could impact the effectiveness of the decision-support system, as interdependencies among variables may introduce biases in the allocation of budgets. For example, improving the conditions in an underserved area may lead to changes in the social, economic, and environmental factors, influencing the overall dynamics of the pavement management system.

To address the endogeneity concerns, advanced statistical techniques like instrumental variables (IV) or control function methods could be employed in. These techniques aim to extract the exogenous component of the endogenous variables, breaking the correlation between the endogenous variables and the error term. Additionally, sensitivity analyses using methods like the Hausman test or two-stage least squares (2SLS) can be conducted to check for the presence of endogeneity and assess the robustness of the results. By employing these advanced statistical techniques and sensitivity analyses, the framework's ability to accurately model the causal relationships can be enhanced, mitigating potential biases introduced by endogeneity in the allocation of budgets.

5. Conclusions

In the world of infrastructure development, asset management in a sustainable manner could affect people in different communities, environments, and economic growth. However, achieving sustainable asset management has many challenges when trying to balance economic, environmental, and social equity objectives.

Balancing the economic, environmental, and social aspects of pavement management involves making careful choices. On the economic front, efforts to enhance social and environmental equity may necessitate significant initial financial investments. For instance, initiatives like constructing new roads in underserved areas or electrifying vehicle fleets to mitigate environmental impact can strain already limited transportation budgets due to high upfront infrastructure and operating costs.

However, certain pavement management strategies demonstrate that economic goals need not always conflict with environmental and social equity objectives. These approaches often yield both environmental and social benefits at lower financial costs.

Policies solely focused on social equity in pavement management may result in overall efficiency losses, potentially leading to greater environmental impacts, higher user costs, and strained budgets for agencies. Nevertheless, well-designed investments targeting traditionally underserved groups have the potential to unlock productivity gains and catalyze economic growth.

This research aimed to develop a prototype decision-support framework for allocating budgets in transportation asset management projects, covering all three equity aspects simultaneously. The focus was on pavement maintenance, rehabilitation, and reconstruction projects. The framework proposed in this study consists of three steps. The first step was defining the research objective, followed by some analysis to investigate key factors in defining underserved areas based on a case study dataset, and then developing an optimization model and finalizing by evaluating the model with some metrics. The optimization model aims to minimize the assets' condition gap between underserved areas and the rest of the network, minimize greenhouse gas emissions, and maximize road network conditions.

The case study was a comprehensive dataset for California's Bay Area, containing sociodemographic and local pavement condition data. The analysis of this dataset revealed that areas with different poverty rates and air quality experience varying pavement conditions. Areas with higher poverty rates and worse air quality tend to have a higher percentage of pavement in poor condition. The Lasso regression method also provided more precise results, confirming the positive correlation between poor air quality and higher poverty rate with the percentage of pavement in poor condition. Furthermore, it revealed a slight increase in poor pavement conditions in areas with an older population. These results highlight that these factors should be considered in the decision-making process to establish an equitable pavement management system.

The findings indicate that land use type might also be an important factor to consider in decision-making process conditions and should be investigated in further studies. Moreover, it shows that poor pavement condition itself might contribute to environmental issues by causing traffic disruptions, increased fuel consumption, and higher emissions, influencing environmental equity assessments. In conclusion, this study's integrated decision-support framework offers valuable insights for sustainable pavement management.

It should be noted that the correlation between poor pavement conditions and impoverished socioeconomic conditions, as observed in the reviewed data, may not universally apply. Factors such as geographical location, economic development, and cultural influences could potentially modify the relationship between pavement conditions and socioeconomic conditions. Despite these considerations, the described framework appears robust and suitable for dissemination to other regions with available data. Considering economic, environmental, and social equity factors simultaneously allows for more informed budget allocation decisions, promoting equitable development and enhancing transportation infrastructure. Future studies can explore additional factors and land use characteristics to refine the framework and further enhance its application in real-world scenarios.

Author Contributions: Conceptualization, S.A. and O.S.; methodology, S.A.; software, S.A.; formal analysis, S.A.; investigation, S.A.; resources, S.A.; data curation, S.A.; writing—original draft preparation, S.A.; writing—review and editing, O.S.; visualization, S.A.; supervision, O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data were publicly available, containing sociodemographic data for California's Bay Area, sourced from the Office of Environmental Health Hazard Assessment (OEHHA) (The Office of Environmental Health Hazard Assessment 2023).

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

Appendix A

Algorithm A1. Proposed multi-objective pavement management algorithm (pseudocode).

```
#Define objective functions
function objective_1(pavement_conditions)
    underserved_condition ← average_condition (pavement_conditions, is_underserved=True)
    other_condition ← average_condition (pavement_conditions, is_underserved=False)
    return |underserved_condition—other_condition|
end function
function objective_2(treatment_plan)
    construction_emissions ← sum (treatment.construction for treatment in treatment_plan)
    traffic_emissions ← sum (treatment.traffic for treatment in treatment_plan)
     return construction_emissions + traffic_emissions
end function
function objective_3(treatment_plan, time_period)
    total_condition ← sum(section.condition_after_treatment(treatment) for section, treatment in zip(pavement_sections, treatment_plan))
    return total_condition/time_period
end function
#Define constraints
function budget_constraint(treatment_plan)
    total_cost ← sum(treatment.agency_cost for treatment in treatment_plan)
    return total_cost ≤ available_budget
end function
function performance_constraint(treatment_plan)
    for each section, treatment in zip(pavement_sections, treatment_plan) do
        if section.condition_after_treatment(treatment) < minimum_acceptable_condition then
            return False
        end if
    end for
    return True
end function
#Define the multi-objective optimization problem
function optimize_pavement_management(pavement_sections, available_budget, time_period)
    all_treatment_plans ← generate_all_treatment_plans(pavement_sections)
    feasible_plans ← []
    for each plan in all_treatment_plans do
        if budget_constraint(plan) and performance_constraint(plan) then
            feasible_plans.append(plan)
        end if
    end for
    objectives ← []
    for each plan in feasible_plans do
        obj1 ← objective_1(plan)
        obj2 ← objective_2(plan)
        obj3 ← objective_3(plan, time_period)
        objectives.append((obj1, obj2, obj3))
    end for
    optimal_plan ← select_optimal_plan(objectives)
    return optimal_plan
end function
```

References

1. Cascetta, E. *Transportation Systems Engineering: Theory and Methods*; Applied Optimization; Springer: Boston, MA, USA, 2001; Volume 49, ISBN 978-1-4757-6875-6.
2. Lõuk, R.; Riid, A.; Pihlak, R.; Tepljakov, A. Pavement Defect Segmentation in Orthoframes with a Pipeline of Three Convolutional Neural Networks. *Algorithms* **2020**, *13*, 198. [CrossRef]
3. Sami, A.A.; Sakib, S.; Deb, K.; Sarker, I.H. Improved YOLOv5-Based Real-Time Road Pavement Damage Detection in Road Infrastructure Management. *Algorithms* **2023**, *16*, 452. [CrossRef]
4. Arezoumand, S.; Sassani, A.; Smadi, O.; Buss, A. From Data to Decision: Integrated Approach to Pavement Preservation in Iowa through Treatment Effectiveness Analysis. *Int. J. Pavement Eng.* **2024**, *25*, 2361085. [CrossRef]
5. France-Mensah, J.; O'Brien, W.J. Developing a Sustainable Pavement Management Plan: Tradeoffs in Road Condition, User Costs, and Greenhouse Gas Emissions. *J. Manag. Eng.* **2019**, *35*, 04019005. [CrossRef]
6. Shafei, B.; Phares, B.; Saini, D.; Azad, S. *Assessment of Bridge Decks with Glass Fiber-Reinforced Polymer (GFRP) Reinforcement (No. MN 2023-13)*; Department of Transportation, Office of Research & Innovation: Saint Paul, MN, USA, 2023.
7. Shafei, B.; Phares, B.; Shi, W.; Azad, S. *Increase Service Life at Bridge Ends through Improved Abutment and Approach Slab Details and Water Management Practices (No. IHRB Projects TR-722 and TR-739)*; Iowa State University: Ames, IA, USA, 2023.
8. Russo, F.; Rindone, C. Evaluation Methods For Evacuation Planning. *WIT Trans. Built Environ.* **2010**, *111*, 335–343. [CrossRef]
9. Adesunkanmi, R.; Al-Hamdan, A.; Nlenanya, I. Prediction of Pavement Overall Condition Index Based on Wrapper Feature-Selection Techniques Using Municipal Pavement Data. *Transp. Res. Rec.* **2023**, *2678*, 03611981231195054. [CrossRef]
10. Arezoumand, S.; Mahmoudzadeh, A.; Golroo, A.; Mojaradi, B. Automatic Pavement Rutting Measurement by Fusing a High Speed-Shot Camera and a Linear Laser. *Constr. Build. Mater.* **2021**, *283*, 122668.
11. Al-Hamdan, A.B.; Nlenanya, I.; Smadi, O. Data-Driven Approach to Identify Maintained Pavement Segments and Estimate Maintenance Type for Local Roads. In Proceedings of the 13th International Conference on Low-Volume Roads, Cedar Rapids, IA, USA, 24–26 July 2023; p. 188.
12. Seyedshohadaie, S.R.; Damnjanovic, I.; Butenko, S. Risk-Based Maintenance and Rehabilitation Decisions for Transportation Infrastructure Networks. *Transp. Res. Part Policy Pract.* **2010**, *44*, 236–248. [CrossRef]
13. Tabesh, M.; Mahmoudzadeh, A.; Arezoumand, S. A Reliability-Base Method for Thermal Cracking Prediction in Asphalt Concrete. *Constr. Build. Mater.* **2023**, *409*, 133912. [CrossRef]
14. Chen, C.; Achtari, G.; Majkut, K.; Sheu, J.B. Balancing Equity and Cost in Rural Transportation Management with Multi-Objective Utility Analysis and Data Envelopment Analysis: A Case of Quinte West. *Transp. Res. Part Policy Pract.* **2017**, *95*, 148–165. [CrossRef]
15. Tatari, O.; Egilmez, G.; Kurmapu, D. Socio-Eco-Efficiency Analysis of Highways: A Data Envelopment Analysis. *J. Civ. Eng. Manag.* **2016**, *22*, 747–757. [CrossRef]
16. Arezoumand, S.; Sassani, A.; Smadi, O. Data-Driven Approach to Decision-Making for Pavement Preservation. *Eng. Proc.* **2023**, *36*, 61. [CrossRef]
17. Office of the Law Revision Counsel. 2012. United States Code, Title 23. Available online: https://www.fhwa.dot.gov/map21/docs/title23usc.pdf (accessed on 24 May 2024).
18. Azad, S.; Mirghaderi, S.R.; Epackachi, S. Numerical Investigation of Steel and Composite Beam-to-Encased Composite Column Connection via a through-Plate. *Structures* **2021**, *31*, 14–28. [CrossRef]
19. Lee, J.; Madanat, S. Optimal Policies for Greenhouse Gas Emission Minimization under Multiple Agency Budget Constraints in Pavement Management. *Transp. Res. Part Transp. Environ.* **2017**, *55*, 39–50. [CrossRef]
20. Ozcan-Deniz, G.; Zhu, Y. Multi-Objective Optimization of Greenhouse Gas Emissions in Highway Construction Projects. *Sustain. Cities Soc.* **2017**, *28*, 162–171. [CrossRef]
21. Sakhaeifar, M.; Tabesh, M.; Newcomb, D.; Lytton, R.; Zollinger, D.; Issa, I.M. *Compilation of Local Studies and Regional Calibration of Pavement ME Design for Rigid and Flexible Pavements in Oklahoma (No. FHWA-OK-2277)*; Department of Transportation: Oklahoma City, OK, USA, 2019.
22. Torres-Machi, C.; Pellicer, E.; Yepes, V.; Chamorro, A. Towards a Sustainable Optimization of Pavement Maintenance Programs under Budgetary Restrictions. *J. Clean. Prod.* **2017**, *148*, 90–102. [CrossRef]
23. Kothari, C.; Asce, S.M.; France-Mensah, J.; O'brien, W.J.; Asce, M. Developing a Sustainable Pavement Management Plan: Economics, Environment, and Social Equity. *J. Infrastruct. Syst.* **2022**, *28*, 04022009. [CrossRef]
24. Dadashova, B.; Dobrovolny, C.S.; Tabesh, M. *Detecting Pavement Distresses Using Crowdsourced Dashcam Camera Images*; University Transportation Center: San Marcos, TX, USA, 2021.
25. Deakin, M. The Case for Socially Inclusive Visioning in the Community-Based Approach to Sustainable Urban Regeneration. *Sustain. Cities Soc.* **2012**, *3*, 13–23. [CrossRef]
26. Hendricks, M.D.; Meyer, M.A.; Gharaibeh, N.G.; Van Zandt, S.; Masterson, J.; Cooper, J.T.; Horney, J.A.; Berke, P. The Development of a Participatory Assessment Technique for Infrastructure: Neighborhood-Level Monitoring towards Sustainable Infrastructure Systems. *Sustain. Cities Soc.* **2018**, *38*, 265–274. [CrossRef]
27. Trudeau, D. Integrating Social Equity in Sustainable Development Practice: Institutional Commitments and Patient Capital. *Sustain. Cities Soc.* **2018**, *41*, 601–610. [CrossRef]

28. Thomopoulos, N.; Grant-Muller, S.; Tight, M.R. Incorporating Equity Considerations in Transport Infrastructure Evaluation: Current Practice and a Proposed Methodology. *Eval. Program Plann.* **2009**, *32*, 351–359. [CrossRef]
29. Boyles, S.D. Equity and Network-Level Maintenance Scheduling. *EURO J. Transp. Logist.* **2015**, *4*, 175–193. [CrossRef]
30. France-Mensah, J.; O'Brien, W.J. Budget Allocation Models for Pavement Maintenance and Rehabilitation: Comparative Case Study. *J. Manag. Eng.* **2018**, *34*, 05018002. [CrossRef]
31. Chi, S.; Hwang, J.; Arellano, M.; Zhang, Z.; Murphy, M. Development of Network-Level Project Screening Methods Supporting the 4-Year Pavement Management Plan in Texas. *J. Manag. Eng.* **2013**, *29*, 482–494. [CrossRef]
32. France-Mensah, J.; Kothari, C.; O'Brien, W.J.; Jiao, J. Integrating Social Equity in Highway Maintenance and Rehabilitation Programming: A Quantitative Approach. *Sustain. Cities Soc.* **2019**, *48*, 101526. [CrossRef]
33. Gunathilaka, S.; Amarasingha, N. Using Social and Economic Factors for Ranking Pavement Maintenance and Rehabilitation Projects. *Asian Transp. Stud.* **2020**, *6*, 100026. [CrossRef]
34. Caggiani, L.; Camporeale, R.; Ottomanelli, M. Facing Equity in Transportation Network Design Problem: A Flexible Constraints Based Model. *Transp. Policy* **2017**, *55*, 9–17. [CrossRef]
35. Praticò, F.G.; Fedele, R.; Naumov, V.; Sauer, T. Detection and Monitoring of Bottom-Up Cracks in Road Pavement Using a Machine-Learning Approach. *Algorithms* **2020**, *13*, 81. [CrossRef]
36. Federal Highway Administration. *Federal Highway Administration Environmental Justice Reference Guide*; Department of Transportation: Washington, DC, USA, 2015.
37. U.S. Department of Transportation. *Centering Equity at the U.S. Department of Transportation*; Department of Transportation: Washington, DC, USA, 2022.
38. Repko, E. *National Highways: Analysis of Available Data Could Better Ensure Equitable Pavement Condition*; Government Accountability Office: Washington, DC, USA, 2022.
39. Lee, J.; Madanat, S.; Reger, D. Pavement Systems Reconstruction and Resurfacing Policies for Minimization of Life-Cycle Costs under Greenhouse Gas Emissions Constraints. *Transp. Res. Part B Methodol.* **2016**, *93*, 618–630. [CrossRef]
40. Yamany, M.S.; Abraham, D.M. Hybrid Approach to Incorporate Preventive Maintenance Effectiveness into Probabilistic Pavement Performance Models. *J. Transp. Eng. Part B Pavements* **2020**, *147*, 04020077. [CrossRef]
41. Pavement Conditions Index (PCI). Metropolitan Transportation Commission. Available online: https://mtc.ca.gov/operations/programs-projects/streets-roads-arterials/pavement-condition-index (accessed on 24 May 2024).
42. OakDOT. *Geographic Equity Toolbox and Equity Map, January 2022 Update City of Oakland Geographic Equity Toolbox (Version 2)*, Department of transpotation: Oakland, CA, USA, 2022.

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34
www.mdpi.com

Algorithms Editorial Office
E-mail: algorithms@mdpi.com
www.mdpi.com/journal/algorithms

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-2010-8