

Article

Predictive Model for Long-Term Lane Occupancy Rate Based on CT-Transformer and Variational Mode Decomposition

Gaoxiang Liu, Xin Yu * and Danyang Liu *

School of Urban Rail Transportation and Logistics, Beijing Union University, Beijing 100101, China; 20222086111207@buu.edu.cn

* Correspondence: 18810536705@163.com (X.Y.); 15011517665@163.com (D.L.)

Abstract: Lane occupancy is a crucial indicator of traffic flow and is significant for traffic management and planning. However, predicting lane occupancy is challenging due to numerous influencing factors, such as weather, holidays, and events, which render the data nonsmooth. To enhance lane occupancy prediction accuracy, this study introduces a fusion model that combines the CT-Transformer (CSPNet-Attention and Two-stage Transformer framework) with the Temporal Convolutional Neural Network-Long Short-Term Memory (TCN-LSTM) models alongside the Variational Mode. This includes a long-term lane occupancy prediction model utilizing the Variational Mode Decomposition (VMD) technique. Initially, the Variational Mode Decomposition decomposes the original traffic flow data into multiple smooth subsequences. Subsequently, each subsequence's autocorrelation and partial correlation coefficients ascertain the presence of seasonal characteristics. Based on these characteristics, the CT-Transformer and TCN-LSTM models process each subsequence for long-term lane occupancy rate prediction, respectively. Finally, predictions from both models are integrated using variable modes to derive the ultimate lane occupancy predictions. The core CT-Transformer model, an enhancement of the GBT (Two-stage Transformer) model, comprises two phases: autoregressive and prediction. The autoregressive phase leverages historical data for initial predictions inputted into the prediction phase. Here, the novel CSPNet-Attention mechanism replaces the conventional attention mechanism in the Encoder, reducing memory usage and computational resource loss, thereby enhancing the model's accuracy and robustness. Experiments on the PeMS public dataset demonstrate that the proposed model surpasses existing methods in predicting long-term lane occupancy, offering decent reliability and generalizability.

Keywords: lane occupancy prediction; time-series prediction; Transformer; TCN-LSTM; Variational Mode Decomposition; CSPNet-Attention



Citation: Liu, G.; Yu, X.; Liu, D. Predictive Model for Long-Term Lane Occupancy Rate Based on CT-Transformer and Variational Mode Decomposition. *Appl. Sci.* **2024**, *14*, 5346. <https://doi.org/10.3390/app14125346>

Academic Editor: Christos Bouras

Received: 24 May 2024

Revised: 18 June 2024

Accepted: 19 June 2024

Published: 20 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As urbanization progresses, it directly results in the continuous expansion of cities, accompanied by significant increases in traffic flow. Although the development of today's intelligent transportation systems and computer technology has matured, their contribution to alleviating traffic congestion and reducing traffic accidents remains insufficient. Traffic flow prediction plays a vital role in the intelligent traffic control and management system, which is crucial for alleviating urban traffic congestion and reducing traffic accidents. Generally, traffic flow data collected by sensors installed on roads at predetermined intervals often exhibit time-series distribution characteristics.

Lane occupancy prediction is a crucial indicator within traffic flow analysis. It refers to the percentage of lanes vehicles occupy on a roadway at any given time. Such predictions inform the degree of roadway congestion and traffic flow density. Lane occupancy prediction differs from traffic flow prediction by focusing specifically on traffic congestion and accidents resulting from lane occupancy [1]. Accurate predictions enable traffic management departments to comprehend road congestion scenarios better and implement

timely measures to mitigate congestion [2]. Real-time monitoring of lane occupancy allows traffic managers to adjust traffic signals, guide flows, and employ strategies to alleviate congestion and enhance road efficiency. It serves as a vital reference for transportation planners and designers. Accurate forecasts aid planners in assessing road network capacity, traffic demand, and identifying bottleneck locations for informed planning, design, and improvement decisions. The outcomes of lane occupancy prediction facilitate rational planning of new roads, expansion of existing ones, and adjustment of traffic flow distribution strategies to meet future demand.

Furthermore, it is crucial for traffic safety management. Accurate lane occupancy predictions enable traffic management and emergency response agencies to identify potential crash risk areas and congestion points proactively. Accurate prediction of lane occupancy facilitates the implementation of appropriate measures, including increasing police patrols, installing temporary traffic signs, and adjusting traffic routes to prevent accidents or mitigate adverse impacts on traffic flow.

Recent advancements in intelligent transportation systems and computer technology have significantly enhanced lane occupancy prediction methods. However, external factors such as weather conditions, traffic accidents, and traffic control influence lane occupancy rates owing to the openness and complexity of urban transportation systems [3]. Accurate lane occupancy prediction necessitates datasets with high-quality attributes, and the rapid advancement of sensor technology has facilitated more accessible data collection, thereby providing ample datasets for prediction. Data collectors, installed on or beneath road-sides, gather vehicle-related information, distributing it according to predefined intervals to exhibit time-series characteristics. However, holidays, weather, and other uncertain factors may affect the collected data, complicating lane occupancy rate prediction. Initially, traditional traffic flow prediction methods, including the Kalman filter and Markov models, were simple but ineffective under complex road conditions. As urbanization intensified, the increasing complexity and variability of urban roads led to the gradual abandonment of these methods. Subsequently, the Autoregressive Integrated Moving Average (ARIMA) model became a commonly used method for lane prediction. The ARIMA model attempts to uncover time-series patterns through autocorrelation and differencing, utilizing these patterns for future predictions [4]. Vector Autoregressive (VAR) models, akin to ARIMA, are favored in traffic flow forecasting for their high flexibility. However, these models exhibit limited capability in capturing the nonlinear features of traffic flow data, leading to suboptimal prediction performance. Standard models like k-nearest neighbors (KNNs) and support vector regression (SVR) also share these drawbacks. Recently, the surge in machine learning and deep learning has led to the development of various neural networks, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Models based on these neural networks have found widespread applications in traffic flow prediction. These neural network models, leveraging deep learning, outperform traditional traffic flow prediction models by effectively extracting spatial and temporal features from time-series data and modeling nonlinear, nonsmooth time series to accommodate the complex dynamics of traffic flow. Additionally, researchers have increasingly studied the Transformer-based model in recent years, demonstrating its superior long-range traffic flow prediction accuracy over prior models.

Long Short-Term Memory (LSTM) [5], a Recurrent Neural Network (RNN) variant, effectively addresses the long-term dependency issue, utilizing past information for future predictions. The state has three gates to control, which can selectively store, update, and output information. Central to LSTM is a unitary state governed by three gates that selectively store, update, and output information. It excels in processing and predicting significant events within traffic flows characterized by time series, especially those with lengthy intervals and delays. However, LSTM is not ideal for long-term prediction due to its computational intensity.

The Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) [6] model combines CNN and LSTM network structures to extract spatio-temporal features

and accomplish sequence modeling. This model preprocesses original traffic flow data, inputs it into the CNN to extract spatial features, then inputs this CNN-processed data into the LSTM layer for next-moment time-series prediction, and finally applies inverse normalization to obtain the predicted value. However, akin to LSTM, this model requires substantial computational resources, and its training process is susceptible to overfitting.

The advent of large-scale artificial intelligence models has spurred extensive research into the Transformer architecture. This deep learning model, predicated on the self-attention mechanism, demonstrates remarkable proficiency in processing sequential data, including text, speech, and images. The Transformer architecture differs from traditional RNN and CNN structures, relying solely on the self-attention mechanism to discern sequence dependencies. The Transformer offers significant advantages, including parallel computation capabilities that enhance training speed and the ability to learn long-distance dependencies, thereby yielding superior outcomes for extensive time-series data.

In recent years, extensive scholarly attention has been directed toward the Transformer architecture, resulting in the proposition of numerous enhanced models. These include the Autoformer [7], ETSformer [8], and FEDformer [9], among others. However, these Transformer model variants are computationally intensive and characterized by complex parameters, leading to significant memory usage and computational resource consumption. The GBT (Two-stage Transformer) [10] addresses the critical issue of overfitting associated with improper decoder input initialization by innovatively dividing the traditional Transformer architecture into two segments: the Auto-Regression and Self-Regression stages. This approach enables the model to enhance its predictive capacity while maintaining low temporal and spatial complexity.

The quality and characteristics of input data significantly influence the accuracy and efficiency of predictive models, which are crucial for the models' performance. This influence stems from the models' learning and inferential capabilities relying on the statistical regularities within the input data. In making forecasts, researchers often overlook the inherent patterns of changes in traffic flow data that demonstrate the characteristics of a time-series distribution; however, these laws significantly impact the accuracy of traffic flow predictions. Consequently, numerous academics have integrated various data decomposition methods into traffic flow prediction to address this issue. The data decomposition method, a signal processing technique, decomposes complex signals into simpler ones, thereby facilitating the analysis and comprehension of the signal's characteristics and structure. Data decomposition methods have extensive applications in time-series forecasting. They enable forecasting models to delve deeply into the intrinsic patterns governing traffic flow data changes, thereby augmenting their capability to extract data characteristics.

Researchers have employed data decomposition methods to preprocess traffic flow data, inputting the decomposed signals into various prediction models and synthesizing each model's outputs. This process significantly enhances the models' capability to extract data features, improving the prediction accuracy and robustness. Commonly employed data decomposition methods, including the Fourier Transform (FT), Wavelet Transform (WT) [11], and Empirical Mode Decomposition (EMD) [12], decompose signals within the frequency, time–frequency, and time domains, respectively, unveiling the signal's structure and characteristics from diverse perspectives. Applying these decomposition methods to traffic flow prediction yielded superior outcomes compared with a singular predictive model. However, the Fourier Transform struggles with non-stationary data and fails to analyze signals' local characteristics adequately. At the same time, the Wavelet Transform is constrained by Heisenberg's uncertainty principle, leading to pseudo-spectra and spectral aliasing in signals with mutations. Empirical Mode Decomposition, conversely, is subject to mode aliasing and endpoint effects.

Section 2 delves into a comprehensive review of this paper's research findings on traffic flow prediction. Section 3 outlines the fusion model's overall architecture, including Variational Mode Decomposition [13], CSPNet-Attention [14], the Two-stage Transformer, and the Temporal Convolutional Neural Network-Long Short-Term Memory Network

(TCN-LSTM) [15]. Section 4 is experimental data processing and experimental validation. Section 5 summarizes the research conducted in this paper and offers insights into the future development of traffic flow prediction.

This study employs Variational Mode Decomposition (VMD) as an adaptive and fully non-recursive method for modal change and signal processing to mitigate the complexity and nonlinearity of time series. Secondly, this study adopts a new attention mechanism, CSPNet-Attention, to replace the traditional attention mechanism, which reduces the computational complexity and memory consumption by sparsifying the attention matrix and achieves similar or even more than the prediction accuracy. Then, we take into full consideration the seasonal characteristics of the experimental data and adopt a model that integrates the Two-stage Transformer and TCN-LSTM model to conduct the prediction experiment. The overall fusion model is named CT+. The experimental results are reconstructed together using VMD to get the final experimental results.

2. Related Works

Traffic flow forecasting is a crucial foundation for traffic management and transportation planning. Traffic flow prediction is vital for both traffic managers and participants. It optimizes traffic signal control, mitigates congestion, enhances road capacity, and provides precise road condition information, guiding travel mode and path selection and improving the travel experience. Consequently, traffic flow prediction significantly enhances urban transportation efficiency and service quality. Deep learning technology is currently extensively applied in this domain, with numerous scholars proposing various traffic flow prediction models leveraging deep learning, see Supplementary Materials Table S1. Wumaier and colleagues [16] employed the random forest algorithm to extract multiple bootstrap samples from the original dataset, constructing a decision tree for each to develop a short-term traffic flow prediction model. This model effectively minimizes prediction error and variance, enhancing the accuracy and efficiency of short-term flow predictions, thereby surmounting traditional methodological constraints. Zhang et al. [17] applied the spatio-temporal feature selection algorithm (STFSA) to identify the optimal time lag and spatial road section count for the input data through correlation analysis, subsequently extracting relevant spatio-temporal traffic flow features and converting them into two-dimensional matrices embedded with spatio-temporal information. Subsequently, CNNs are utilized to extract and learn features from these two-dimensional matrices, facilitating the construction of a prediction model. This approach leads to the proposal of a short-term traffic flow prediction model that integrates spatio-temporal analysis within a CNN-based deep learning framework. Du et al. [18] leveraged a deep learning framework combining irregular convolutional residual networks with LSTM units to propose an urban traffic flow prediction model based on a Deep Irregular Convolutional Residual LSTM Network (DST-ICRL). Chen et al. [19] introduced a trainable location matrix within the graph convolutional network framework to facilitate adaptive learning of spatial patterns in graph structures. They proposed a Location-Based Graph Convolutional Network (Location-GCN) and amalgamated Long Short-Term Memory (LSTM) with the Location-GCN to devise an end-to-end traffic prediction model.

As deep learning technology advances, numerous innovative frameworks have emerged, with the Transformer model proposed by Vaswani et al. [20] of Google standing out as particularly notable. This model eschews traditional CNN and RNN techniques, adopting Attention as its core method for representation learning. Furthermore, it integrates techniques like Positional Encoding, Feed-Forward Networks, Residual Networks, and Regularization to enhance its generalization capabilities and computational efficiency. The model performs excellently across various Natural Language Processing (NLP) tasks.

Given the Transformer architecture's significant impact on deep learning, numerous researchers have focused on its enhancement and expansion. Their objectives include improving computational efficiency and generalization capacity, mitigating overfitting or underfitting risks, and boosting adaptability to diverse tasks and datasets. Innovations such

as Autoformer, FEDformer, BERT [21], and Pyraformer [22] exemplify the continual evolution and optimization of the Transformer model, underscoring the research community's dedication to advancing model performance. Throughout this evolution, the attention mechanism—a technique for flexible information selection and synthesis—has emerged as a focal point of research. Its development has spurred the creation of numerous variants, such as multi-head Attention [20], bottleneck attention [23], cross-modality Attention [24], and graph attention [25]. These variants showcase the attention mechanisms' robust adaptability and innovation potential across various application contexts. The advent of attention mechanisms signifies their transition from supplementary functions to fundamental modeling tools, establishing them as essential technologies in fields including image processing, speech recognition, knowledge graphs, and recommender systems. Specifically, within the crucial domain of traffic flow prediction, integrating deep learning techniques with attention mechanisms has led researchers to propose various innovative models to enhance prediction accuracy and reliability.

For instance, Zhang et al. [26] introduced the innovative GGCN-SA, a deep learning-based traffic flow prediction model. It leverages a Graph Convolutional Network (GCN) for spatial correlation and Gated Recurrent Units (GRUs) for temporal dependence, further employing a soft attention mechanism to aggregate spatiotemporal information selectively, thereby improving its characterization of traffic flow's spatio-temporal features. Lin et al. [27] proposed STAtt-Net, a novel Spatial and Temporal Attention-based Convolutional Network framework, to accurately predict citywide traffic flow. Initially, the system represents traffic data as a two-dimensional, two-channel matrix, and each cell correlates with the traffic flow of a specific area. Acknowledging the transportation system's temporal correlation and dependence, Researchers delineate the periodic patterns of the traffic data into three principal components: weekly trends, daily periodicity, and hourly proximity. Subsequently, STAtt-Net, utilizing the attention mechanism with STBlock as the foundational unit, learns the traffic flow's temporal and spatial dependencies. This approach achieves superior prediction performance, particularly in addressing issues like spatial non-stationarity that challenge traditional methods.

Given the nonlinear and stochastic characteristics inherent to time-series data, including temperature changes, rainfall, interest rate fluctuations, exchange rate fluctuations, and traffic flow, accurate analysis and forecasting of these data are paramount. Such characteristics indicate that simple linear models are insufficient for capturing the complex interrelationships among data, necessitating advanced nonlinear models like neural networks and support vector machines to uncover underlying connections. Randomness implies that data values or their occurrence probabilities are indeterminate, adhering to certain probability distributions or stochastic processes, necessitating the application of mathematical tools like probability theory, mathematical statistics, and stochastic methods for their analysis and processing. Additionally, data instability, reflected in the variability of statistical properties like mean, variance, and covariance over time or space, necessitates employing techniques such as smoothing, differencing, and covariance analysis for effective management.

Considering the earlier analysis, accurately analyzing and predicting time-series data—including temperature changes, rainfall, interest rate fluctuations, exchange rate fluctuations, and traffic flow—presents a formidable research challenge with significant practical application value. Consequently, numerous scholars have investigated the pre-model training decomposition of data to improve its smoothness, thereby augmenting the model's capability to discern the data's intrinsic structure. These decomposition techniques transform complex, nonlinear, and nonsmooth raw data into stable subsequences that are more amenable to analysis and prediction, thereby revealing more profound data properties and laying a solid foundation for subsequent model training and prediction. In this endeavor, scholars persist in refining and innovating decomposition techniques to suit various data types, further enhancing the accuracy and efficiency of time-series analysis and prediction.

Li et al. [28] initially employed wavelet decomposition to categorize original traffic flow data into high-frequency and low-frequency subsets, enhancing data stability. Subsequently, they applied a fusion model of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to predict the subsequent day's traffic flow. Zhao et al. [29] initially applied a fast Fourier Transform and its inverse within the time-series correlation module to decompose traffic flow data. They then used an attention-based encoder–decoder graph convolutional model to predict multi-step future traffic flow. Tian et al. [30] used the Empirical Mode Decomposition method to segment traffic flow into components of varied frequencies, leveraging the self-similarity of each to choose appropriate prediction models for short-term forecasting.

Compared with wavelet decomposition, fast Fourier Transform, and Empirical Mode Decomposition, Variational Mode Decomposition autonomously identifies the requisite number of modal decompositions, preventing modal aliasing and thus potentially enhancing post-processing prediction accuracy. Research by numerous scholars has demonstrated significant improvements in model prediction accuracy following the application of Variational Mode Decomposition to time-series data. For instance, Zhao et al. [31] initially smoothed traffic flow data using Variational Mode Decomposition, inputted the resulting subsequences into an LSTM network prediction model, optimized the model parameters using the dung beetle algorithm (IDBO), and aggregated the subsequences' predicted values to derive final predictions. Yu et al. [32] applied Variational Mode Decomposition to segment historical traffic flow data into K components, determined by varying K sample entropy values. They then smoothed each component using wavelet thresholding to obtain denoised subsequences, predicted each using an LSTM network, and aggregated these predictions for final results. Huang et al. [33] addressed the non-stationarity and complexity of traffic flow data by decomposing it into K subsequences using Variational Mode Decomposition and constructing a CNN-GRU network within the Keras framework to extract temporal features. Given the uncertainty of traffic flow data influenced by external factors like travel demand and traffic signals, Kim et al. [13] proposed a hybrid model incorporating Variational Mode Decomposition for traffic flow prediction.

Building on the preceding analysis, employing suitable data decomposition techniques coupled with integrating various prediction methods can substantially elevate the prediction model's performance and enhance its future event forecasting accuracy. Data decomposition simplifies and reduces its noise by segmenting the original dataset into several sub-signals, facilitating more efficient extraction of essential features and patterns. Furthermore, selecting algorithms or models that align with the data's characteristics is crucial to establishing a mapping relationship between the dataset and the predictive target. Deep learning algorithms for model optimization and enhancement further boost the prediction methods' accuracy and applicability. This holistic approach augments the prediction model's efficacy and offers a more accurate and reliable method for forecasting future events in complex data scenarios.

3. Model Structure

This section primarily introduces the overall architecture of the model utilized in this study, the components of the model, and the processing of experimental data.

3.1. Overall Model Architecture

This study initially segments time-series data into numerous sub-series using the Variational Mode Decomposition (VMD) technique, designed to uncover the data's intrinsic structure and dynamic properties. Subsequently, the system calculates autocorrelation and partial correlation coefficients for each subseries, representing a critical step in identifying temporal dependencies within the data. The system employs distinct models for training, depending on the presence or absence of seasonal trends in each subsequence: the CT-Transformer model for subsequences with seasonal trends to capture cyclical variations and the TCN-LSTM model for those without, focusing on long-term dependency extraction and

learning. This differential approach allows for the most suitable analysis and prediction of each subsequence.

The VMD technique is ultimately applied to reconstruct the subsequences trained by the models, synthesizing the final predicted sequences. This integration process culminates in a composite prediction framework that leverages the multidimensional nature of time-series data to enhance prediction accuracy. Figure 1 illustrates the detailed model architecture design, delineating the complete process from data decomposition through model training to sequence reconstruction.

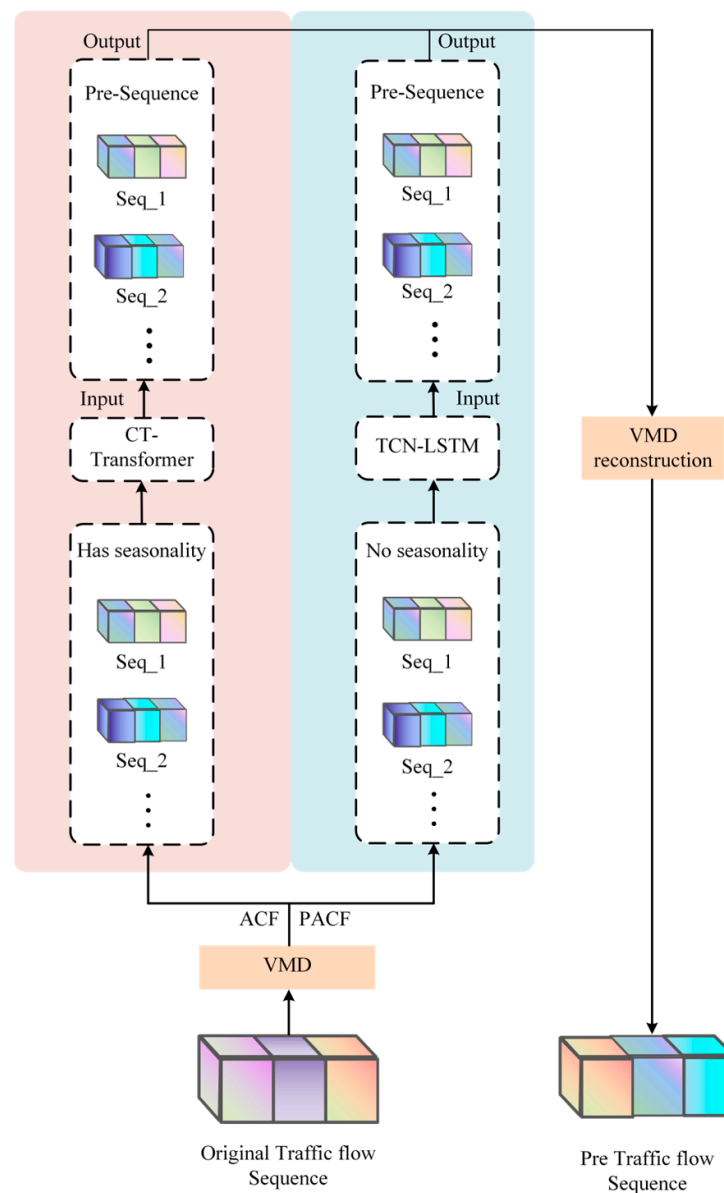


Figure 1. Overall model architecture.

3.2. Variational Mode Decomposition

The key advantage of VMD [34] is its ability to adaptively determine the number of modal decompositions based on the signal's actual conditions. An adaptive search mechanism precisely matches the modes' center frequencies and bandwidths, efficiently separating the intrinsic modal components (IMFs) and segmenting the signal into frequency domains, thereby identifying key signal components effectively [35].

Assuming the original time-series data are decomposed into k components, each a modal component with a finite bandwidth and centered frequency, while minimizing the estimated bandwidths' sum, with the constraint that the sum of all modes equals the original signal, the VMD-constrained variational model proceeds as follows in Equation (1):

$$\min_{\{u_k, \omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \quad (1)$$

$$s.t. \sum_k u_k = f$$

Here, $u_k = \{u_1, u_2, \dots, u_k\}$ represents the modal function, $\omega_k = \{\omega_1, \omega_2, \dots, \omega_k\}$ signifies the central frequency of each mode, $j/\pi t$ is a complex term where j denotes the imaginary unit and $1/\pi t$ is a decay function, and $e^{-j\omega_k t}$ is an exponential function where ω_k is a frequency variable. Researchers convert the constrained optimization problem into an unconstrained variational problem by applying a quadratic penalty term and the Lagrangian multiplier method. Consequently, they formulate the augmented Lagrangian function as follows in Equation (2):

$$L(\{u_k\}, \{\omega_k\}, \lambda) := \alpha \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \langle \lambda(t), f(t) - \sum_k u_k(t) \rangle \quad (2)$$

Here, α represents the penalty parameter, and λ signifies the Lagrange multiplier.

The comprehensive algorithmic flowchart of the Variational Mode Decomposition (VMD) algorithm is illustrated in Figure 2 below:

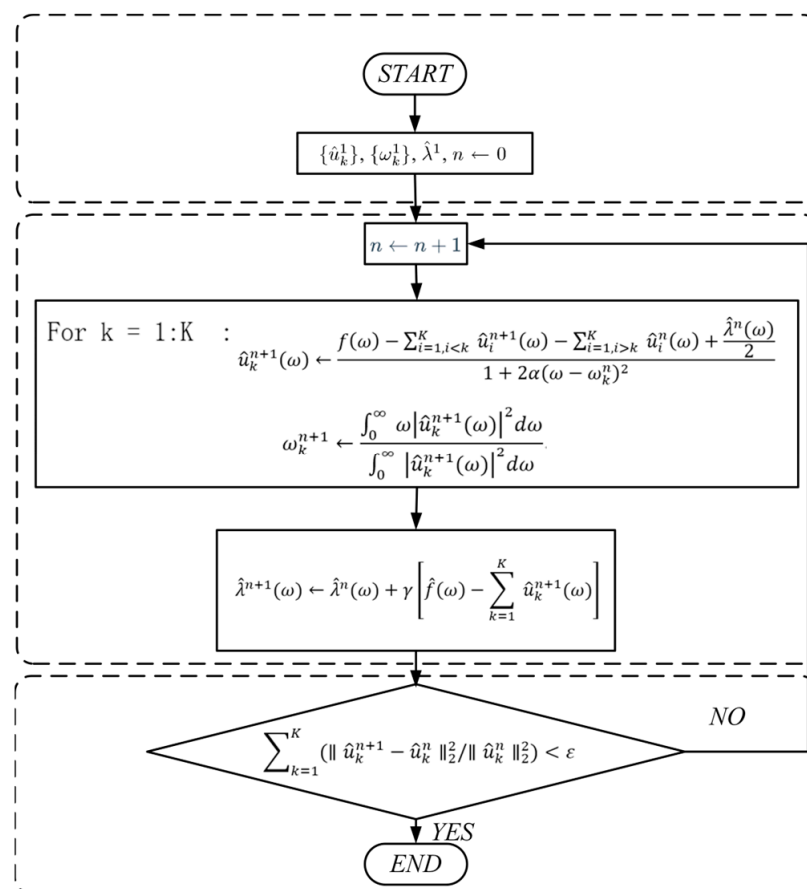


Figure 2. Flowchart of VMD algorithm.

In Figure 2, we first transform the constrained optimization problem into an unconstrained variational problem using a quadratic penalty term and the Lagrange multiplier method and subsequently update the generalized function until the specified iterative constraints are satisfied.

3.3. CSPNet-Attention

The CSPNet-Attention mechanism, building upon the foundation of the Cross Stage Partial Network (CSPNet) [14], was developed to mitigate the computational demands imposed by extensive inference calculations observed in prior works while also addressing the redundancy of gradient information during the network optimization process. The inception of the Cross Stage Partial Network aimed to harmonize the first and final feature maps of network stages, thereby accommodating gradient variability. CSPNet streamlines the optimization of the F-function (the mapping from the input x_0 to target y) by predominantly integrating feature maps from the initial and terminal stages of the network, thus curtailing computational load. The optimization formula is delineated as follows in Equation (3):

$$y = M([x_0', T(F(x_0''))]) \quad (3)$$

In the CSPNet-Attention framework, the input x_0 is bifurcated into two segments along the channel dimension, represented as $x_0 = [x_0', x_0'']$, where T signifies the transition function employed to truncate the gradient flow across $H_1, H_2, \dots, H_K$, where H_K refers to the operating function of the k th layer in a Convolutional Neural Network (CNN). This function usually consists of a set of convolutional layers and a nonlinear activation function. Essentially, H_k represents the computational process within the k th layer of the network.

M represents the transition function designated for amalgamating the bifurcated segments. Given an input within $R^{L \times d}$, where L symbolizes the input length and d the input dimension, it is partitioned by the dimension into $X = [X_1^{L \times d_1}, X_2^{L \times d_2}]$, essentially dividing the input into two components. Here, X_1 , after transiting through a one-dimensional convolutional layer A, is concatenated at the block's conclusion, while X_2 serves as the input to the self-attention block B. The outputs of A and B are concatenated across dimensions to formulate the entire block's output. The output matrix of the first level in the CSPNet-Attention architecture is articulated as follows in Equation (4):

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} X_1 W_c \\ \text{Concat}(A(X_{2h})W_h)_{h=1}^H \end{bmatrix} \quad (4)$$

In the CSPNet-Attention framework, $A(X_{2h})$ signifies the scaled dot product attention within the h th self-attention block, where W_h represents the $d_h \times d_h$ linear projection matrix. Here, H denotes the number of heads, d_h specifies the dimension of each head, and W_c is the weight matrix for the one-dimensional convolutional layer, with dimensions $d/2 \times d/2$. The formulation updates the weights within CSPNet-Attention as follows in Equation (5):

$$\begin{aligned} W_h' &= f(g_2, W_h) \\ W_c' &= f(g_1, W_c) \end{aligned} \quad (5)$$

In this framework, f represents the weight update function, while g signifies the gradient propagated along the i th path. Notably, the network integrates the gradient of the segmented part independently, illustrating a distinct approach to gradient processing within the network.

The overall network architecture of CSPNet-Attention is shown in Figure 3 below:

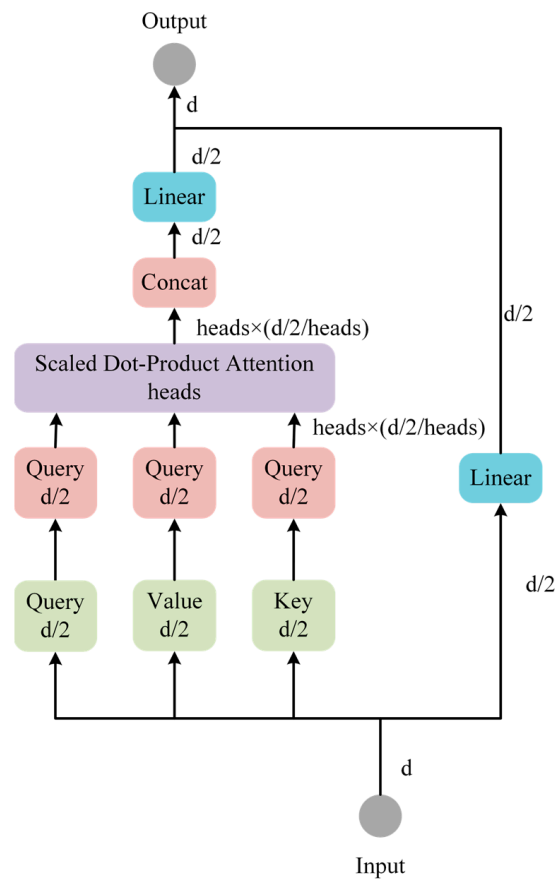


Figure 3. Overall architecture of the cross-stage part of the network.

In this network architecture, a bifurcation splits the input along its dimensionality. A self-attentive block, having half the input's dimensionality, processes the left segment. Meanwhile, a 1×1 convolutional layer directly routes the right segment to the end of the entire block.

3.4. Two-Stage Transformer

This study introduces an innovative framework for time-series forecasting algorithms, the Two-stage Transformer [10]. It optimizes the traditional Transformer forecasting method's structure by segmenting it into an Auto-Regression Stage and a Self-Regression Stage. This design addresses the discrepancies in statistical properties between the input and predicted sequences. Central to this framework, the Two-stage Transformer optimizes the traditional Transformer architecture through transformative and decoupling operations, significantly reducing time and space complexity.

To further enhance model performance, this study substitutes the original self-attention mechanism with the CSPNet-Attention mechanism in the Encoder, augmenting the model's capacity to capture sequence features. Moreover, the Two-stage Transformer incorporates an innovative Error Score Modification (EMS) module, employing a learnable Gaussian distribution $N(0, \sigma_i^2)$, where $i \in [1, l_{out}]$, to optimize the model's prediction accuracy and robustness. Within EMS, the distribution's center is set to zero, enabling the model to bias auto-attention score assignments towards earlier, more reliable predictive elements. Consequently, this approach amplifies the influence of earlier predictor elements on subsequent ones in the autoregressive process, thereby improving the model's overall performance in time-series prediction.

Algorithm 1 illustrates the algorithmic flow of the error score correction module, with its algorithmic formulation provided as follows in Equation (6):

$$ESM(Q, K) = QK^T + \begin{bmatrix} N(0, \sigma_1^2) \\ N(0, \sigma_2^2) \\ \dots \\ N(0, \sigma_{lout}^2) \end{bmatrix} \quad (6)$$

Algorithm 1 Masked self-attention with ESM

Input: Tensor $x \in \mathbb{R}^{L \times d}$

Layer params: Linear(x): Linear projection layer of x

1: $Q, K, V = \text{Linear}_1(x) (Q, K, V \in \mathbb{R}^{L \times d})$

2: $\sigma = \text{Linear}_2(x) (\sigma \in \mathbb{R}^{L \times 1})$

3: Calculate self-attention scores $A = QK^T (A \in \mathbb{R}^{L \times L})$

4: $\sigma = \text{Sigmoid}(5\sigma) + 1 \times 10^{-5}$

5: $\sigma = 3^\sigma - 1$

6: $\sigma = \text{Repeat}(\sigma, \text{dim} = 1) (\sigma \in \mathbb{R}^{L \times L})$

7: Generate Gauss distribution matrix $G \sim N(0, \sigma^2) (G \in \mathbb{R}^{L \times L})$

8: Add self-attention scores with ESM: $A = A + G$

9: $A = \text{mask}(A)$

Output: Masked self-attention scores A

In the $N(0, \sigma_i^2)$ configuration, σ_i can undergo continuous adaptation throughout the model training process, rendering the error score correction module highly adaptable.

We add CSPNet-Attention to replace the self-attention mechanism in the first stage, and the improved model architecture is shown in Figure 4 below.

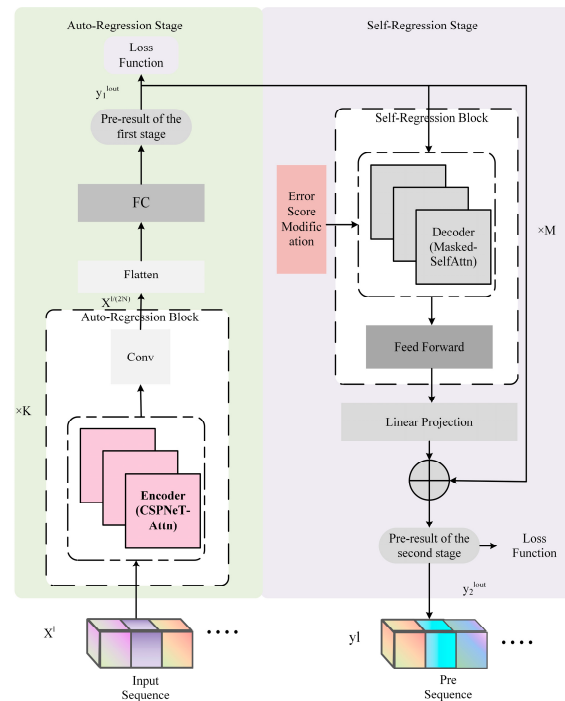


Figure 4. Overall architecture of the Two-stage Transformer Model with integrated CSPNet-Attention.

In this framework, we input the input variables into the Encoder in the first stage, followed by a convolution operation, and then the output is flattened into a one-dimensional tensor, and finally, a linear transformation is realized through a fully connected layer; in the second stage, the output of the first stage is first used as input into a decoder, and then

non-linear transformation and feature extraction are carried out through the feed-forward neural network (which consists of two FC layers and an activation function). Finally, the result is mapped back to the input dimension by linear projection.

3.5. TCN-LSTM

The Temporal Convolutional Neural Network-Long Short-Term Memory (TCN-LSTM) model epitomizes a cutting-edge hybrid neural network architecture, adeptly merging the benefits of Temporal Convolutional Networks (TCNs) and Long Short-Term Memory (LSTM). TCNs, by harnessing their unique convolutional structure for parallel processing of data series, significantly bolster a model's proficiency in capturing long-term data dependencies, thus addressing time-series prediction challenges. It consists of a one-dimensional full convolutional network and several stacked expansive randomized convolutional networks. This characteristic renders TCN exceptionally suitable for analyzing and predicting sequence data with intricate time-dependent structures. Meanwhile, LSTM, a distinct class of Recurrent Neural Networks, addresses the short-term memory challenges traditional RNNs face with lengthy data sequences. LSTM efficiently orchestrates long-term information storage and retrieval via a gating mechanism, enabling sustained historical memory and agile capture of short-term sequence dependencies. This capability is pivotal in comprehending and forecasting time series of variable lengths and intricate patterns.

Merging TCN's long-term dependency-capturing prowess with LSTM's adept management of short-term dependencies, the TCN-LSTM model emerges as a potent and versatile solution for time-series prediction. Fusing the strengths of both networks, the model optimizes time-series data processing depth and breadth and amplifies its adaptability and precision for intricate sequence prediction tasks.

The TCN algorithm [36] unfolds in two primary phases: convolution operations and residual linking. Its convolution operations utilize dilated convolution to expand the receptive field by interspersing zeros between convolution kernels. The mathematical expression for the dilated convolution employed in the TCN model is given by Equation (7):

$$Y[t] = \sum_{i=0}^{K-1} w[i] \cdot X[t - d \cdot i] \quad (7)$$

where $Y[t]$ represents the output at time t , $X[t]$ is the input signal, $W[i]$ denotes the weight at the i th position of the convolution kernel, k is the size of the kernel, and d represents the dilation factor, which determines the spacing between the weights in the convolution operation.

The residual block in the TCN model comprises two convolutional layers coupled with a residual connection. In this, 1D-FCN is a one-dimensional fully convolutional network that is responsible for processing the input data and slides the convolution kernel over the time step or sequence length of the input data to extract temporal or sequential features. The computation process within the residual block unfolds as follows:

1. The input x is processed through a dilated convolutional layer, resulting in the output y .
2. The output y is then added to the input x to form the residual block's output.

The mathematical expression for the residual block is the following Equation (8):

$$\text{Output} = x + \text{Conv1D}(x) \quad (8)$$

Long Short-Term Memory (LSTM) represents a specialized Recurrent Neural Network (RNN) variant designed to learn long-term temporal dependencies. The essence of LSTM is its cell state, akin to a conveyor belt running through the network, which is impacted by a minimal number of linear transformations, thereby ensuring a consistent flow of information. Additionally, LSTM manages the addition or removal of information to/from the cell

state through an ingeniously designed gate mechanism. The mathematical expression for the core function in the LSTM is given in the following Equation (9):

$$\begin{aligned}\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t\end{aligned}\quad (9)$$

In the context of LSTM, $\odot$ denotes the element-wise multiplication operation. $\tilde{c}_t$ denotes the candidate cell state, which represents the result of processing the new information at the current moment t . $\tanh$ is the hyperbolic tangent function, one of the activation functions, which compresses the input into the range $[-1, 1]$. W_c is the weight matrix connecting the input x_t to the candidate cell state c_t , U_c is the weight matrix connecting the hidden state h_{t-1} from the previous moment, $\tilde{c}_t$, h_{t-1} is the hidden state of the previous moment, and b_c is the bias vector. f_t is the forgetting gate, which determines which parts of the cell state should be forgotten or retained. Its output value is within $[0, 1]$. c_{t-1} is the cell state at the previous moment, and i_t is the input gate that determines how much of the current moment's input is written to the cell state. Its output value is within $[0, 1]$. The output gate and the cell state jointly determine the hidden state unit h_t of the LSTM c_t . The formulation for generating h_t is as follows in Equation (10):

$$h_t = o_t \odot \tanh(c_t) \quad (10)$$

In LSTM, a connection exists not only between the hidden units h_{t-1} and h_t but also in a linear self-loop between cell states c_{t-1} and c_t . This linear self-loop mechanism, functioning as an information sliding process, retains past information when the gating unit is activated and discards it upon deactivation. It enables LSTM to address long-term dependency issues in RNNs. Considering the synergistic combination of TCN and LSTM, leveraging their respective strengths, processed time-series data are initially input into the TCN and subsequently passed through LSTM via convolution and residual block processing, with the final output derived from a fully connected layer. The TCN-LSTM model amalgamation aims to serve as a potent instrument for time-series prediction, with the TCN layer tasked with feature extraction and the LSTM layer dedicated to sequence modeling. This hybrid methodology enhances the model's capacity to learn from data embodying complex temporal dynamics [37]. The overall network architecture of the TCN-LSTM model is shown in Figure 5 below:

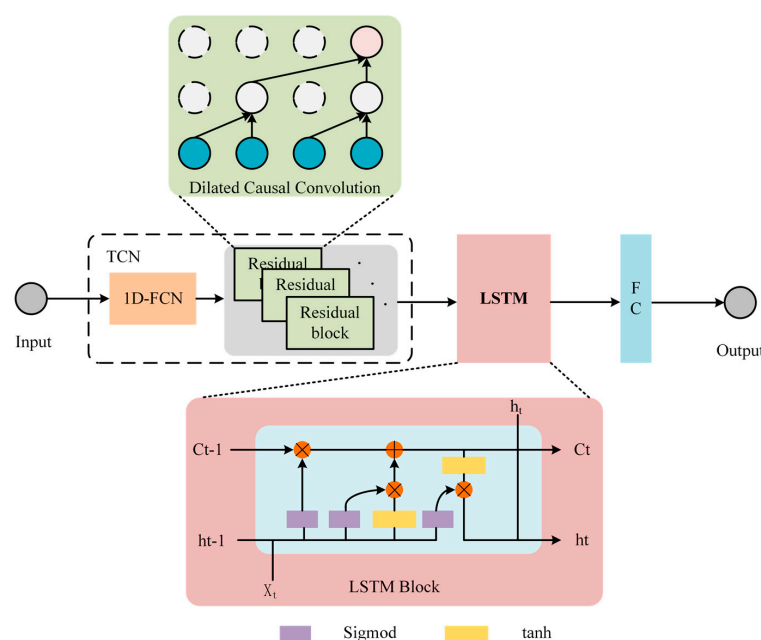


Figure 5. Overall architecture of the TCN-LSTM model.

4. Experiments and Analysis

4.1. Experimental Data Processing

The traffic flow dataset (PeMS) used in this study originates from a comprehensive data collection initiative by the California Department of Transportation (Caltrans) across the state's freeway system, supplemented by contributions from other state transportation agencies and partners. The dataset spans 48 months from 2016 to 2018 and concentrates on roadway occupancy along San Francisco's freeway. Aggregating and organizing this data enabled researchers to offer a unified, comprehensive analytical framework for assessing the freeway system's operational performance. The framework builds on a thorough analysis of the freeway network's current state, facilitating the development of effective operational strategies and identifying and examining critical bottleneck zones for potential improvements, thereby offering robust data support for traffic management decisions.

Collecting and analyzing this dataset is crucial in optimizing traffic flow management and enhancing road usage efficiency. This study's data were meticulously segmented into training, validation, and testing sets, with allocations of 70%, 10%, and 20%, respectively. This allocation aims to ensure the model training's effectiveness and generalizability while affording ample data to validate and test the model's performance, striving for high accuracy and reliability. Details of the dataset are provided in Table 1.

Table 1. Description of PeMS data details.

Dataset	Time	Acquisition Frequency	Total Number	Splitting Strategy		
				Training	Validation	Test
PeMS	1 July 2016–2 July 2018	60 min	17,544	0–12,281	12,281–14,035	14,035–17,544

This study employs an innovative seasonal characterization approach to explore seasonal trends within the PeMS Traffic dataset. As depicted in Figure 6, the data were first segmented by season, and then average values were calculated for each season. Comparative analysis reveals that the dataset exhibits more pronounced trend fluctuations during summer and winter, as opposed to spring and fall.

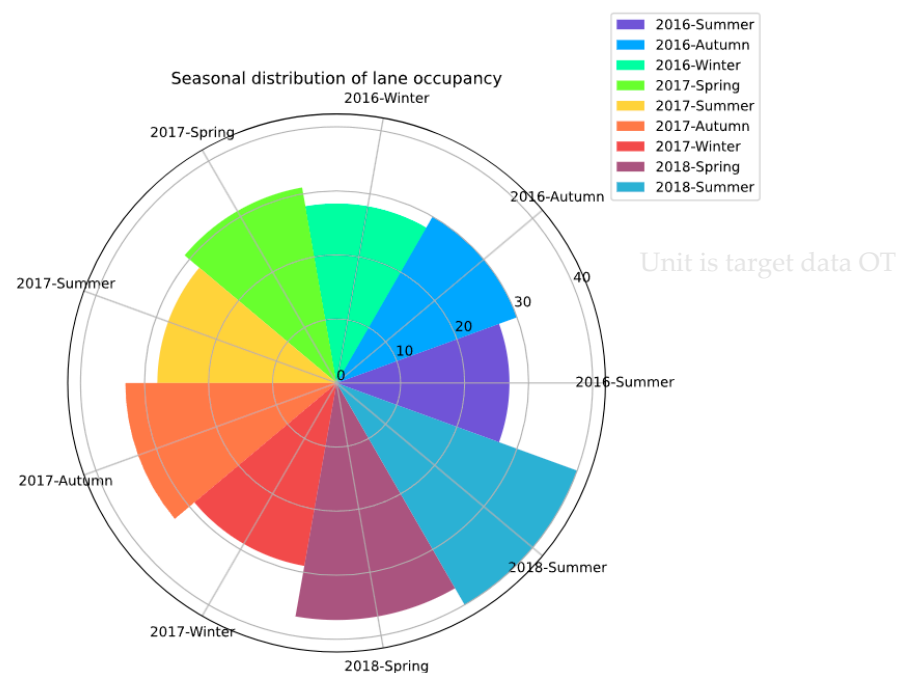


Figure 6. Segmentation illustrating the seasonal characteristics of the PeMS Traffic dataset.

Integrating the computation of autocorrelation coefficients (ACF) and partial autocorrelation coefficients (PACF) within the Variational Mode Decomposition (VMD) framework enhances the analysis further. This addition aimed to facilitate an in-depth assessment of the temporal properties of each subsequence derived from VMD. Calculating the ACF and PACF values for each subseries enabled the accurate determination of seasonal trends within each; higher values of ACF and PACF indicate stronger correlations with the season. This analysis not only deepened the understanding of seasonal variations within the dataset but also, by melding VMD with time-series characterization, propelled the identification of seasonal trends. Applying this approach yields a more precise and systematic framework for the seasonal characterization of the PeMS Traffic dataset, thus enriching the comprehension of traffic flow trends.

Assuming that the time series is $\{x_t\}$, the autocovariance function (ACVF) of lag order k is as follows in Equation (11):

$$\begin{aligned} c_k &= \frac{1}{n} \sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x}) \\ c_0 &= \frac{1}{n} \sum_{t=1}^n (x_t - \bar{x})^2 \end{aligned} \quad (11)$$

The autocorrelation coefficients are defined as follows in Equation (12):

$$acf_k = \frac{c_k}{c_0} = Cov(x_t, x_{t+k}) \quad (12)$$

The partial autocorrelation function (PACF) measures the linear relationship between the series $\{x_t\}$ and its lag $\{x_{t-k}\}$ of order k , excluding the linear influence of all intervening lags $\{x_{t-1}, x_{t-2}, \dots, x_{t-(k-1)}\}$, and is calculated as follows in Equation (13):

$$pacf_k = \begin{cases} Cov(x_1, x_0) = r_1 & \text{if } k = 1; \\ Cov(x_k - x_k^{k-1}, x_0 - x_0^{k-1}) & \text{if } k \geq 2; \end{cases} \quad (13)$$

4.2. Evaluation of Performance Indicators

In this study, we utilized the Mean Absolute Error (MAE) and Mean Square Error (MSE) as evaluation indices to assess the experimental outcomes. The MSE is particularly adept at reflecting the model's performance, especially when the error distribution is approximately normal, offering a robust performance evaluation. Furthermore, employing MAE as a loss function can enhance the model's generalization capabilities in scenarios characterized by substantial outliers or noise within the data. The computational formulas for MAE and MSE are defined as follows in Equations (14) and (15):

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (14)$$

The Mean Absolute Error (MAE) is the average of the absolute differences between the target and predicted values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (15)$$

The Mean Square Error (MSE) is the most commonly utilized regression loss function, calculated by summing the squares of the discrepancies between predicted and actual values. Taken together, MAE (Mean Absolute Error) and MSE (Mean Squared Error) serve as evaluation metrics; smaller values of these metrics indicate superior model performance. Although MAPE provides an intuitive understanding of percentage error in some cases,

it is usually limited in practical applications owing to its treatment of zero values and over-sensitivity to small values, so we chose to use MAE and MSE, which are more stable and versatile.

4.3. Comparative Experiment

In this study's experimental environment, we trained our model on a platform that included Python 3.11.5, PyTorch 2.11.0, and CUDA 12.1. We configured the hardware with an i5-13490f CPU, a GeForce RTX 4060ti GPU (8 GB), a 2048 GB hard disk, and 32 GB of RAM, operating on Windows 11 Professional Edition. Table 2 presents the configuration of the parameters in the CT-Transformer model.

Table 2. Experimental parameters.

Hyper-Parameters/Settings	Values
Embedding dimensions of the Auto-Regression stage	32
Embedding dimensions of CSPNet-Attention(factor)	32
CSPNet-Attention(scale)	5
CSPNet-Attention(dropout rate)	None
Self-Regression stage	0.1
Kernel size of Conv layers	3
Input window size	96/192/336/720
Standardization	Z-score
Loss function	MSE
Optimizer	Adam
Dropout rate	0.1
Learning rate	1×10^{-4}
Learning rate decreasing rate	Half/epoch
Batch size	32

In CSPNet-Attention, the “factor” parameter governs the sparsity of sparse self-attention, determining the maximum relative distance considered for each position. “Scale” is an optional factor that adjusts the attention scores and is commonly employed in attention mechanisms to modulate the attention distribution range. “Dropout rate” refers to the probability of omitting units during the attention calculation, a regularization strategy designed to prevent model overfitting. In the TCN-LSTM hybrid model, we configured the prediction window to 96/192/336/720, set the learning rate at 0.001 and the dropout rate at 0.05, and conducted 20 training epochs. We chose a batch size of 32, allocated 64 units to the hidden layer, and applied a convolutional layer with a kernel size of 3.

Our proposed CT-Transformer model addresses the limitations of traditional forecasting methods in handling non-stationary time series and their significant computational resource consumption. We benchmarked against two advanced state-of-the-art models, NHiTS and SCINet. NHiTS, a neurally-hierarchical interpolated time-series forecasting method, employs multi-rate data sampling, linear projection, and temporal interpolation of the input signal to tackle long-term forecasting challenges. SCINet, employing sample convolution and interactions, introduces a novel method that significantly enhances the efficiency of temporal modeling and forecasting in time-series analysis. This method constructs a binary tree-structured temporal convolutional network, considerably improving temporal data handling. Additionally, for comparison, the study selected four cutting-edge Transformer variants: Autoformer, ETSformer, FEDformer, and GBT. Among these, Autoformer employs autocorrelation decomposition for long-term forecasting, splitting the input series into deep seasonal trends and stochastic fluctuations, and models cyclical dependencies using an autocorrelation attention mechanism. ETSformer integrates exponential smoothing and seasonal-level growth decomposition. FEDformer employs frequency-enhanced decomposition, utilizing Fourier and wavelet transforma-

tions. The GBT (Two-stage Transformer) breaks down the forecasting process into autoregression phases, tackling the disparity in statistical characteristics between input and predicted sequences. The research further highlights its superiority by comparing the CT-Transformer with several state-of-the-art forecasting models. The researchers align the hyperparameters of the involved models, including the proposed model, with those of several baseline models.

The research presents the experimental results of the proposed model and the baseline models in Table 3, setting the prediction window to four distinct sizes: 96, 192, 336, and 720. The team conducted three experiments for each size to ensure stable experimental outcomes. The CT-Transformer outperforms the baseline models' overall prediction performance, as demonstrated in the table. At a prediction window of 96, the CT-Transformer achieved relative reductions in mean absolute error (MAE) of 45.9%, 26.9%, 51.3%, 26.4%, 49.6%, and 8.70% compared with Autoformer, ETSformer, FEDformer, NHiTS, SCINet, and GBT, respectively. It also reduced the mean squared error (MSE) by 65.2%, 37.2%, 70.2%, 38.3%, 67.6%, and 14.4%, respectively. At prediction windows of 192, 336, and 720, the CT-Transformer similarly demonstrated significant reductions in MAE and MSE, showcasing superior accuracy and robustness over advanced time-series forecasting models. These comparisons are depicted in bar charts in Figures 7 and 8, comparing the MAE and MSE values of the proposed fusion model with those of the baseline models across the four prediction lengths.

Comparison of MAE values of four prediction lengths by different methods

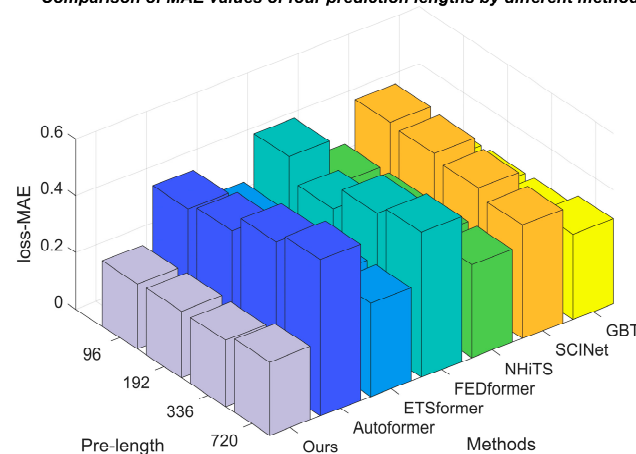


Figure 7. Comparison of MAE Values Across Different Prediction Window Sizes.

Comparison of MSE values of four prediction lengths by different methods

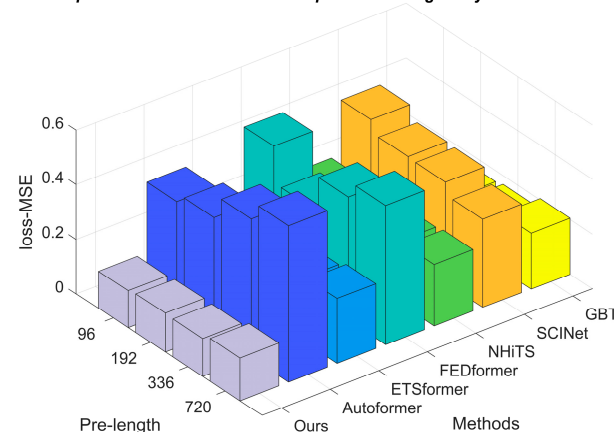


Figure 8. Comparison of MSE values across different prediction window sizes.

To more clearly illustrate the superiority of the proposed model, Figure 9 presents comparison plots between our model and the baseline models, showcasing the valid values across various prediction window sizes.

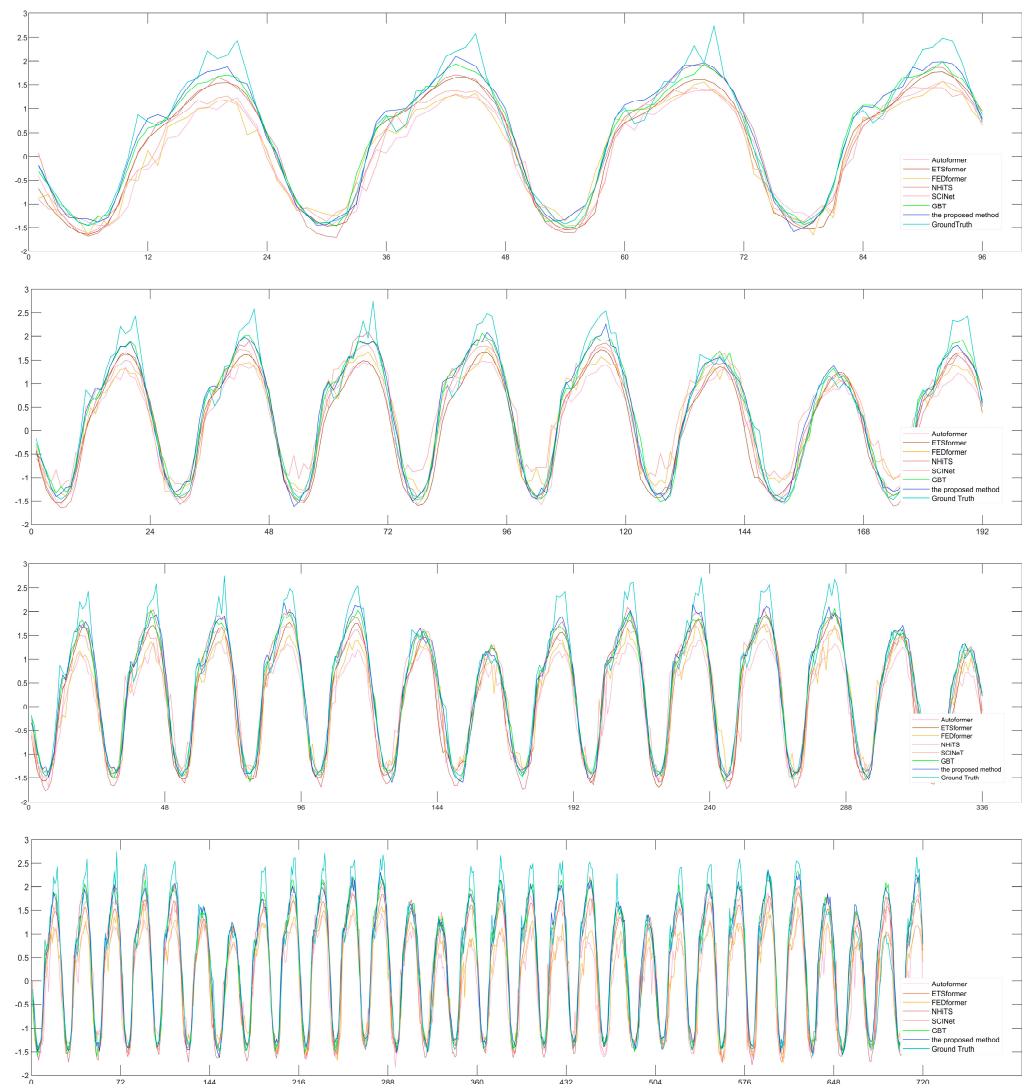


Figure 9. Comparative prediction results of different models at prediction window sizes of 96, 192, 336, and 720.

Table 3. Comparative experimental results of different models at various prediction window sizes.

Methods	Source	Metrics	PeMS Traffic Dataset			
			96	192	336	720
CT-Transformer Module		MAE	0.231	0.236	0.234	0.258
		MSE	0.137	0.142	0.138	0.156
Autoformer	NeurIPS2021	MAE	0.427	0.450	0.513	0.551
		MSE	0.394	0.425	0.509	0.573
ETSformer	arXiv2022	MAE	0.316	0.325	0.324	0.329
		MSE	0.218	0.233	0.220	0.238
FEDformer	PMLR2022	MAE	0.474	0.392	0.477	0.511
		MSE	0.460	0.326	0.452	0.510

Table 3. Cont.

Methods	Source	Metrics	PeMS Traffic Dataset			
			96	192	336	720
NHiTS	AAAI2022	MAE	0.314	0.315	0.324	0.328
		MSE	0.222	0.229	0.220	0.227
SCINet	NeurIPS2022	MAE	0.458	0.449	0.429	0.400
		MSE	0.423	0.374	0.370	0.324
GBT	NeuralNetworks2023	MAE	0.253	0.257	0.260	0.299
		MSE	0.160	0.160	0.168	0.205

4.4. Ablation Study

In this subsection, we conduct ablation studies to assess the functionality of the CT-Transformer, Variational Mode Decomposition, CSPNet-Attention, and TCN-LSTM within the fusion model presented in this paper. We evaluate five ablation variants of the fusion model: (1) Ours: the complete CT-Transformer model incorporating Variational Mode Decomposition fused with TCN-LSTM; (2) no-V: the complete CT-Transformer model excluding Variational Mode Decomposition, fused with TCN-LSTM; (3) no-C: the CT-Transformer model devoid of CSPNet-Attention, coupled with TCN-LSTM in the fusion model; (4) no-T: the complete CT-Transformer model without TCN-LSTM integration; (5) no-VT: the complete CT-Transformer model without Variational Mode Decomposition and TCN-LSTM; and (6) no-VCT: the model excluding Variational Mode Decomposition, CSPNet-Attention, and TCN-LSTM. Figure 10 below illustrates the results of the ablation studies.

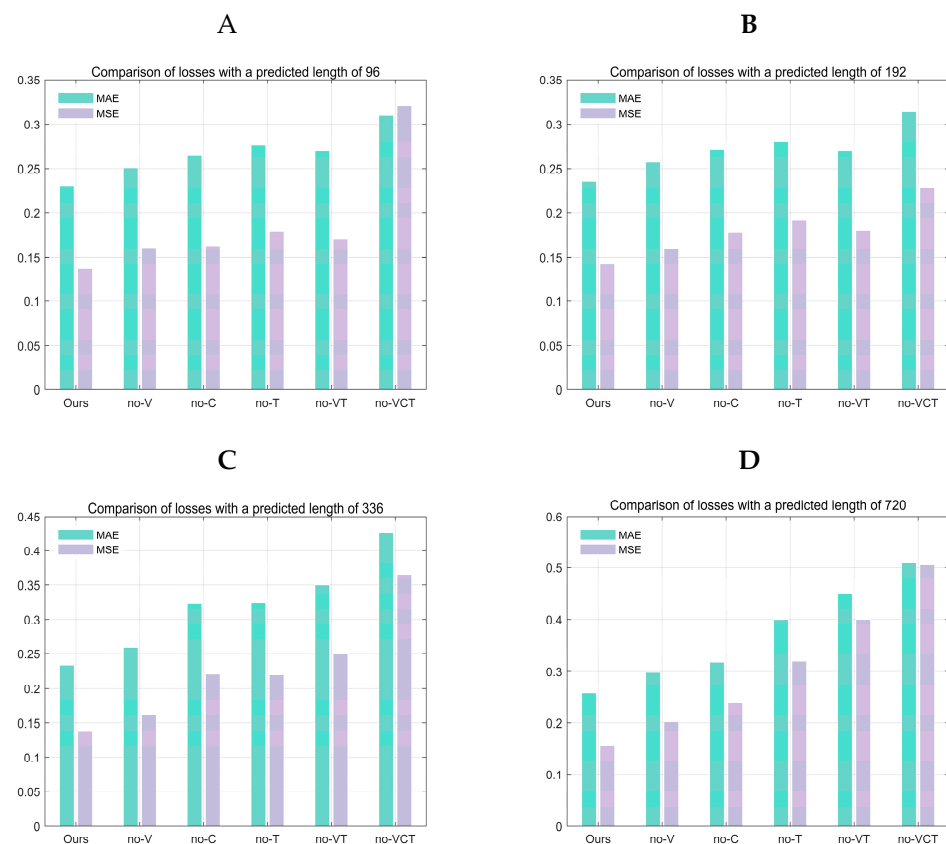


Figure 10. Results of ablation studies across four prediction window sizes.

A in Figure 10 represents the results of the ablation experiment when the predicted window size is 96, B represents the results of the ablation experiment when the predicted window size is 192, C represents the results of the ablation experiment when the predicted window size is 336, and D represents the results of the ablation experiment when the predicted window size is 720.

Table 4 below illustrates that the “Ours” method outperforms the alternative methods in predictions using the PeMS Traffic dataset. Upon removal of the Variational Mode Decomposition, CSPNet-Attention, and TCN-LSTM, individually or collectively, we observed increases in MAE values of 8.7%, 14.7%, 19.5%, and 34.2% and MSE values of 17.5%, 18.9%, 30.7%, and 134%, respectively.

Table 4. Comparative table of ablation study results.

Methods	Metrics	PeMS Traffic Dataset			
		96	192	336	720
Ours	MAE	0.231	0.236	0.234	0.258
	MSE	0.137	0.142	0.138	0.156
no-V	MAE	0.251	0.257	0.259	0.298
	MSE	0.161	0.160	0.162	0.202
no-C	MAE	0.265	0.271	0.324	0.317
	MSE	0.163	0.178	0.221	0.238
no-T	MAE	0.276	0.280	0.325	0.400
	MSE	0.179	0.191	0.220	0.319
No-VT	MAE	0.270	0.271	0.348	0.449
	MSE	0.171	0.182	0.249	0.399
no-VCT	MAE	0.310	0.315	0.426	0.509
	MSE	0.321	0.229	0.365	0.505

The findings reveal that omitting three critical elements from the fusion model substantially reduces its effectiveness owing to various negative impacts. This evidence underscores the vital importance of each component in boosting the model’s performance within the context of this research.

5. Conclusion and Future Directions

This study introduces a model architecture integrating Variational Mode Decomposition, the CT-Transformer, and TCN-LSTM for analyzing non-stationary traffic flow data with time-series characteristics. Variational Mode Decomposition is initially applied to segment the data into multiple subsequences. Subsequently, each subsequence’s autocorrelation and partial autocorrelation coefficients are analyzed to assess their seasonal characteristics. We propose replacing the attention mechanism in the Encoder module of the two-stage Transformer’s first stage with CPSNet-Attention to reduce computational costs and memory usage, thereby enhancing model prediction accuracy. We designate the refined model as the CT-Transformer.

Furthermore, given the TCN-LSTM model’s robust long-term dependency modeling capabilities, subsequences lacking seasonal characteristics after modal decomposition are inputted into it. In contrast, those with seasonal characteristics are directed to the CT-Transformer. Ultimately, the output predictions from both models undergo variational mode reconstruction, culminating in the final predictive outcomes. Subsequent validations of the dataset demonstrate the fusion model’s substantial predictive capabilities and robustness.

Additionally, we recognize the importance of explainable AI (XAI) in achieving the interpretability of the proposed technique. Following recent trends in other vertical domains, incorporating XAI can provide transparent and interpretable insights into the model's decision-making process. This approach not only enhances the trustworthiness of the model but also facilitates its adoption in practical applications. Future work will explore the integration of XAI methods to further improve the interpretability and explainability of our traffic flow prediction model.

Future traffic flow prediction endeavors will increasingly incorporate a broader array of external factors, extending beyond weather and temperature to include holidays, the surge in new energy vehicles, and the implementation of new road traffic regulations. Recent studies reveal that conditions at the current location dictate alterations in traffic flow, and flows at adjacent locations also significantly influence these changes. This interplay among adjacent locations suggests a spatial correlation or dependence within traffic flows. Consequently, future research in traffic flow prediction may evolve by examining inter-location traffic flows within a defined radius. Researchers anticipate that this avenue of research will enhance the accuracy and promptness of traffic flow predictions, thereby providing more dependable support for transportation planning and management decisions. A comprehensive analysis of spatial correlation's impact on traffic flow will elucidate the intricacies of urban transportation systems and furnish effective strategies for alleviating traffic congestion and optimizing transportation networks.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app14125346/s1>. Table S1: Related job descriptions. Table S2: Relevant technologies covered by the study. Table S3: Abbreviations and corresponding explanations.

Author Contributions: Conceptualization, G.L.; methodology, G.L.; software, G.L.; validation, G.L.; formal analysis, G.L.; investigation, G.L. and X.Y.; resources, G.L.; data curation, G.L.; writing—original draft preparation, G.L.; writing—review and editing, G.L.; visualization, G.L. and D.L.; supervision, X.Y.; project administration, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Innovation for Education Fund, Science and Technology Development Center, Ministry of Education (2018A03017).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Since we are using publicly available data, we provide here a link to the data used in the experiment, https://drive.google.com/drive/folders/1ZOYpTUa82_jCcXIdTmyr0LXQfvaM9vIy?usp=sharing, accessed on 18 June 2024.

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

References

1. Guan, K.; Ai, B.; Fricke, A.; He, D.; Zhong, Z.; Matolak, D.W.; Kurner, T. Excess propagation loss modeling of semiclosed obstacles for intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2171–2181. [\[CrossRef\]](#)
2. Harrou, F.; Dairi, A.; Zeroual, A.; Sun, Y. Forecasting of Bicycle and Pedestrian Traffic Using Flexible and Efficient Hybrid Deep Learning Approach. *Appl. Sci.* **2022**, *12*, 4482. [\[CrossRef\]](#)
3. Tian, R.; Bi, J.; Zhang, Q.; Liu, Y. Research on Lane Occupancy Rate Forecasting Based on the Capsule Network. *IEEE Access* **2020**, *8*, 38776–38785. [\[CrossRef\]](#)
4. Wu, F.; Zheng, C.; Zhang, C.; Ma, J.; Sun, K. Multi-View Multi-Attention Graph Neural Network for Traffic Flow Forecasting. *Appl. Sci.* **2023**, *13*, 711. [\[CrossRef\]](#)
5. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
6. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
7. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.

8. Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; Hoi, S. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv* **2022**, arXiv:2202.01381.
9. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022.
10. Shen, L.; Wei, Y.; Wang, Y. GBT: Two-stage transformer framework for nonstationary time series forecasting. *Neural Netw.* **2023**, *165*, 953–970. [[CrossRef](#)] [[PubMed](#)]
11. Grossmann, A.; Morlet, J. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.* **1984**, *15*, 723–736. [[CrossRef](#)]
12. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
13. Kim, E.-J.; Park, H.-C.; Kho, S.-Y.; Kim, D.-K. A Hybrid Approach Based on Variational Mode Decomposition for Analyzing and Predicting Urban Travel Speed. *J. Adv. Transp.* **2019**, *2019*, 3958127. [[CrossRef](#)]
14. Wang, C.Y.; Liao HY, M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, Seattle, WA, USA, 13–19 June 2020; pp. 390–391.
15. Gopali, S.; Abri, F.; Siami-Namini, S.; Namin, A.S. A Comparative Study of Detecting Anomalies in Time Series Data Using LSTM and TCN Models. *arXiv* **2021**. [[CrossRef](#)]
16. Wumaier, H.; Gao, J.; Zhou, J. Short-term forecasting method for dynamic traffic flow based on stochastic forest algorithm. *J. Intell. Fuzzy Syst. Appl. Eng. Technol.* **2020**, *39*, 1501–1513. [[CrossRef](#)]
17. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y. Short-term traffic flow prediction based on spatiotemporal analysis and CNN deep learning. *Transp. A Transp. Sci.* **2019**, *15*, 1688–1711.
18. Du, B.; Peng, H.; Wang, S.; Alam Bhuiyan, M.Z.; Wang, L.; Gong, G.; Liu, L.; Li, J. Deep Irregular Convolutional Residual LSTM for Urban Traffic Passenger Flows Prediction. *IEEE Trans. Intell. Transp-Ortation Syst.* **2019**, *21*, 972–985. [[CrossRef](#)]
19. Chen, Z.; Lu, Z.; Chen, Q.; Zhong, H.; Zhang, Y.; Xue, J.; Wu, C. A spatial-temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism. *Inf. Sci.* **2022**, *611*, 522–539. [[CrossRef](#)]
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
21. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
22. Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A.X.; Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In Proceedings of the International Conference on Learning Representations, Online, 25–29 April 2021.
23. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
24. Zhu, X.; Cao, B.; Xu, S.; Liu, B.; Cao, J. Joint Visual-Textual Sentiment Analysis Based on Cross-Modality Attention Mechanism. In Proceedings of the MultiMedia Modeling MMM 2019, Thessaloniki, Greece, 8–11 January 2019; Kompatsiaris, I., Huet, B., Mezaris, V., Gurrin, C., Cheng, W.H., Vrochidis, S., Eds.; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany, 2019; Volume 11295.
25. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *Stat* **2017**, *1050*, 10–48550.
26. Zhang, H.; Chen, L.; Cao, J.; Zhang, X.; Kan, S. A combined traffic flow forecasting model based on graph convolutional network and attention mechanism. *Int. J. Mod. Phys. C* **2021**, *32*, 2150158. [[CrossRef](#)]
27. Lin, J.; Lin, C.; Ye, Q. Attention based convolutional networks for traffic flow prediction. *Multimed. Tools Appl.* **2024**, *83*, 7379–7394. [[CrossRef](#)]
28. Li, Y.; Chai, S.; Ma, Z.; Wang, G. A Hybrid Deep Learning Framework for Long-Term Traffic Flow Prediction. *IEEE Access* **2021**, *9*, 11264–11271. [[CrossRef](#)]
29. Zhao, S.; Li, X. An Attention Encoder-Decoder Dual Graph Convolutional Network with Time Series Correlation for Multi-Step Traffic Flow Prediction. *J. Adv. Transp.* **2022**, *2022*, 1–17. [[CrossRef](#)]
30. Tian, Z. Approach for Short-Term Traffic Flow Prediction Based on Empirical Mode Decomposition and Combination Model Fusion. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5566–5576. [[CrossRef](#)]
31. Zhao, K.; Guo, D.; Sun, M.; Zhao, C.; Shuai, H. Short-Term Traffic Flow Prediction Based on VMD and IDBO-LSTM. *IEEE Access* **2023**, *11*, 97072–97088. [[CrossRef](#)]
32. Yu, Y.; Shang, Q.; Xie, T. A Hybrid Model for Short-Term Traffic flow Prediction Based on Variational Mode Decomposition, Wavelet Threshold Denoising, and Long Short-Term Memory Neural Network. *Complexity* **2021**, *2021*, 7756299. [[CrossRef](#)]
33. Huang, X.; Ma, C.; Zhao, Y.; Wang, K.; Meng, W. A hybrid model of neural network with VMD-CNN-GRU for traffic flow prediction. *Int. J. Mod. Phys. C* **2023**, *34*, 2350159. [[CrossRef](#)]
34. Dragomiretskiy, K.; Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Process.* **2013**, *62*, 531–544. [[CrossRef](#)]
35. Guo, K.; Yu, X.; Liu, G.; Tang, S. A Long-Term Traffic Flow Prediction Model Based on Variational Mode Decomposition and Auto-Correlation Mechanism. *Appl. Sci.* **2023**, *13*, 7139. [[CrossRef](#)]

36. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
37. Qu, D.; Chen, K.; Wang, S.; Wang, Q. A Two-Stage Decomposition-Reinforcement Learning Optimal Combined Short-Time Traffic Flow Prediction Model Considering Multiple Factors. *Appl. Sci.* **2022**, *12*, 7978. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.