ITC 2/54

Information Technology
and Control
Vol. 54 / No. 2 / 2025
pp. 380-395
DOI 10.5755/j01.itc.54.2.39228

A Prediction Method for Highway Traffic Flow Based on
the IHPO-VMD-LSTM-Informer Model

Received 2024/10/24 | Accepted after revision 2025/02/04

HOW TO CITE: Wang, R., Cao, Y., Ji, X., Qiao, D. (2025). A Prediction Method for Highway Traffic Flow Based on the IHPO-VMD-LSTM-Informer Model. *Information Technology and Control*, 54(2), 380-395. https://doi.org/10.5755/j01.itc.54.2.39228

A Prediction Method for Highway Traffic Flow Based on the IHPO-VMD-LSTM-Informer Model

Ruinan Wang

University of Sydney, College of Science, NSW 2008, Australia

Yan Cao

Shandong Medical College, Medical Imaging Department, Linyi 250002, China

Xingyu Ji

University of Sydney, College of Business, NSW 2008, Australia

Di Qiao*

Shandong Women's University, School of Tourism, Jinan 250300, China

Corresponding authors: qiaod2323@163.com

Accurate and timely predictions of highway traffic flow are crucial for implementing intelligent highway management. This paper introduces a novel prediction approach for highway traffic flow by employing the IHPO-VMD-LSTM-Informer model, aiming at enhancing prediction accuracy. Initially, key indicators measuring highway traffic are identified, and Nonlinear Principal Component Analysis (NPCA) is applied to minimize the dimensionality and interdependence among these indicators. This reduction process replaces the original complex indicators with fewer numbers of principal components, thereby simplifying the feature matrix's structure. Subsequently, Variational Modal Decomposition (VMD) processes historical highway traffic flow data, enhanced by the strategically improved Hunter-Prey Optimization (HPO) algorithm. This optimization facilitates adaptive parameter adjustments for the VMD, enabling effective decomposition of highway traffic flow time series data. The Sample Entropy (SE) of Intrinsic Modal Functions (IMFs) from this decomposition is used with the substantial indicators to form a comprehensive feature matrix. Then, the predictive module combines a Long Short-Term Memory (LSTM) network with the Informer architecture to accurately predict highway traffic flow from the feature matrix. The effectiveness of the proposed model is verified using a public motorway traffic dataset KDD CUP 2017. The results indicate that the proposed model outperforms available ones in terms of prediction accuracy, where MAPE and RMSE have 8.09 and 2,84, thus significantly advancing intelligent highway management.

KEYWORDS: Highway traffic flow, Nonlinear Principal Component Analysis, Variational Modal Decomposition, Hunter-Prey Optimization, LSTM

1. Introduction

Intelligence in highway traffic management leverages cutting-edge information technology and sophisticated data analysis to monitor and perceive the operational state of highways in real time. This approach effectively addresses traffic congestion during highway operations, making predictions for traffic flow in critical research areas in intelligent traffic flow management. The nature of highway traffic flow involves the distribution of vehicles across varying times and spaces, contributing to its inherent unpredictability, non-linearity, and non-stationarity. These complexities even make available prediction methods suffer from substantial errors, failing to satisfy the demands of contemporary societal development. Consequently, more accurate, real-time prediction methods to serve current requirements better have been required [6].

Initially, gathering data on motorway traffic was challenging due to its scarce availability and poor quality, which often is restricted to forecasting studies of small datasets. Consequently, many conventional prediction methods relied heavily on complex models grounded in purely mathematical theories, neglecting traffic flows' intrinsic properties and evolutionary patterns [1]. In short-term traffic flow forecasting, variations in research subjects, prediction intervals, and various data formats complicate direct comparisons of methodologies' strengths and weaknesses. Current research typically focuses on single-step forecasts covering brief intervals from 1 to 15 minutes, aiming to enhance the precision and utility of models by integrating multiple forecasting techniques. The literature faces several significant challenges regarding the prediction of highway traffic flow, which are briefly expressed below [22]:

1 Traffic Vulnerability refers to a reduction in service quality across a transport system when disruptions occur in a traffic network. Factors such as the topology of road networks, traffic demands, and environmental conditions jointly influence traffic vulnerability. For example, rapid urban growth has substantially increased highway traffic demands, exacerbating traffic volatility, and diminishing the system's resilience to external shocks, complicating forecasting efforts.

2 Traffic Modeling, using linear and nonlinear system theories, provides mathematical models whose aim is to predict future traffic volumes using available data, without adequately accounting for traffic's evolutionary traits.

3 Prediction Accuracy must be highly dependable and useful to refine traffic management effectively. Current methodologies, which typically analyze only real-time traffic data, struggle to meet online prediction requirements when high traffic volumes and unstable system operations occur. Thus, a new approach within a big data framework has emerged to address these evolving needs.

Thus, data-driven methods are promising to deal with those issues.

In addressing the challenges of highway traffic flow prediction, this paper introduces a method called the IHPO-VMD-LSTM-Informer. The novel contributions of the research are detailed below: Initially, a nonlinear principal component analysis technique [15] is crafted to reduce the dimensions and correlations among the indicators measuring highway traffic flow. By utilizing only key principle components instead of numerous original indicators, the input feature matrix is simplified for the prediction module. Furthermore, traffic flow data is decomposed using Variational Modal Decomposition (VMD), paired with the multi-strategy improved Hunter-Prey Optimization (IHPO) algorithm. This combination allows for adaptive decomposition, effectively mitigating the issues of over or under-decomposition typically caused by manual parameter adjustments. This approach enables deeper data feature mining via Sample Entropy (SE) and other indicators. Additionally, the prediction model integrates the Long Short-Term Memory (LSTM) network with the Informer model to enhance feature extraction and prediction accuracy. This integration processes time series data through the LSTM model to capture spatial and temporal features, which are then fed into the Informer model's encoder for generating traffic flow predictions. In summary, the IHPO-VMD-LSTM-Informer model was a significant advancement in traffic flow prediction. To address the key limitations of available methods such as over-reliance on comprehensive feature sets, manual parameter tuning, and isolated feature processing, the proposed approach not only enhances prediction accuracy but also contributes to understanding and managing highway traffic dynamics more effectively.

The contributions of the article are as follows:

1. The relatively low prediction accuracy of fluctuating highway flow traffic is highly improved by combining 4 different methods such as HPO, VDM, LTSM, and Informer.
2. The indicators measuring highway traffic flow are preprocessed using Nonlinear Principal Component Analysis to eliminate redundant components so that a smaller number of principal components are reached, which eases the computational process. Also, the nonlinear structure of the data is better treated.
3. More refined feature extraction is also realized by using VDA and IHPO supported by Sample Entropy, which makes the process effective.
4. The LSTM-Informer model is more prone to time series data which extracts spatial and temporal features and generates better predictions.

The paper is structured as follows: Section 1 introduces the current situation and problems related to the approaches used to predict highway traffic flow. Section 2 presents the related work, describing the prediction models. Section 3 deals with the feature extraction of the time series data of highway traffic flow and constructs the feature matrix. Section 4 proposes the prediction module for highway traffic flow using time series and gives the overall process of the prediction. Section 5 validates and tests the effectiveness of the proposed model. Section 6 discusses the implications for specific applications. Section 7 concludes the research and provides an outlook.

2.Related Work

Previously, the methods called exponential smoothing, Kalman filtering, and time series analysis predominantly governed traffic flow predictions. However, data-driven approaches for predicting traffic flows arise. Depending on their inherent model features, contemporary prediction methods can be categorized into 4 main types: linear and nonlinear system theory, knowledge discovery for intelligent prediction, and advanced methods [2].

1. Linear system theory includes various models such as a historical average, time series, Kalman filter, and linear regression. Time series forecasting analyzes trends by examining data's sequence over time, which is straightforward to implement and typically delivers accurate predictions when data is ample, and traffic patterns are consistent. However, one downside is that time series forecasting requires extensive parameter estimation, which can hinder a model's portability and flexibility. Additionally, model identification and testing within a prediction framework can be quite complex. The Kalman filter technique, an enhanced autoregressive data processing approach, integrates state and observation equations into a state space model, deriving its predictive algorithm from modern control theory's Kalman filter principles. Abadi et al. [1] initially utilized a dynamic traffic simulator to generate flows on all network links by leveraging available traffic details, projected demand, and historical data from sensors on links. Subsequently, they applied an optimization technique to refine the origin-destination matrix powering the simulator and employed both real-time and estimated traffic data for predicting future flows on each link for up to 30 minutes. Their predictive technique employed an autoregressive model that adjusted to unforeseen incidents. For a practical application, they operated a macroscopic traffic flow simulator to fore-

cast flows in a traffic network in San Francisco, California, USA, and used Monte Carlo methods to assess the method's precision. Zhang et al. [25] introduced a new and efficient method for predicting short-term traffic flows, which enhanced the Kalman filter to detect and eliminate noise, thereby preserving valuable signals through a specifically designed cost function. They verified its effectiveness in short-term traffic predictions via comprehensive testing on 4 benchmark datasets. Their results not only surpassed the conventional Kalman filter but also outperformed other widespread parametric and non-parametric methods. On the other hand, Chan et al. [3] explored a method for advancing neural networks using exponential smoothing, aimed at improving neural networks previously deployed in traffic flow predictions. This technique preprocessed traffic data through exponential smoothing before it was fed into a neural network for training, resulting in preprocessed data that exhibited fewer non-smooth aspects, discontinuities, and clusters, making it better suited to train neural networks.

2 Nonlinear system theory consists of wavelet analysis, mutation theory-based forecasting, and chaos theory-based prediction. The wavelet analysis-based approach decomposes historical time series data of traffic. Its strength is in the localized examination of temporal and spatial frequency data. Despite its utility, wavelet analysis acts only as one component in the broader prediction framework, requiring integration with additional models to forecast effectively. Kushchenko et al. [11] utilized Morlet wavelets for analyzing traffic flow characteristics. They focused on data spanning a week, from Monday to Sunday, assessing average speeds and vehicle counts on various road sections. Then, wavelet spectra and scan plots were developed and scrutinized to determine correlations between extreme points and variations in vehicle speeds and counts. Ni et al. [14] recognized traffic flow's inherent nonlinearity and substantial disruptions, noting its variable characteristics across different time-frequency dimensions, using the wavelet analysis to break down a comprehensive set of raw traffic signals into distinct time-sequence signals, each showcasing unique features. Subsequently, these wavelet-transformed signals were processed using the ARIMA model.

3 Knowledge discovery encompasses tools like support vector machines (SVM) and artificial neural network (ANN) models. SVMs are recognized for their robust generalization capabilities. They are also adaptable to a wide range of machine learning tasks beyond pattern recognition, for example, function approximation. However, SVMs may be less effective with very large datasets. Artificial neural networks or connectionist systems, are computational models that mimic key aspects of the human brain's structure and function. These models generally require substantial data with long training duration and are less adaptable. They are typically optimized for specific scenarios, which might limit their broader application. Feng et al. [5] developed an innovative prediction method for short-term traffic flow utilizing a spatiotemporal correlation adaptive multi-kernel support vector machine (AMSVM), dubbed AMSVM-STC. Initially, they explored nonlinear and stochastic characteristics of traffic flows, employing a combination of Gaussian and polynomial kernels to construct the AMSVM. Subsequently, they applied an adaptive particle swarm optimization technique to refine AMSVM's parameters, allowing the hybrid kernel's weights to adjust in response to real-time traffic trends. Furthermore, they integrated spatiotemporal correlation data into AMSVM to enhance the predictability of short-term traffic flow, demonstrating that the algorithm effectively adapts to rapid changes during peak traffic periods. Meanwhile, Wang et al. [19] implemented a backpropagation (BP) neural network for forecasting traffic flow. They configured the network's input layer with variables such as date, time, car count, and average speed, while the output layer predicted total traffic flow for specified periods. Their simulations confirmed the accuracy of the proposed model, although the network's initial configuration—specifically its weights and thresholds—significantly influenced its performance. Stanulov et al. [18] provided a comparison of machine learning methods. Moreover, data fusion based on sensor data was studied in [9].

4 Advanced Techniques, Graph Neural Networks (GNNs), and Reinforcement Learning (RL)

emerged as powerful tools for traffic prediction, leveraging spatial relationships between different road segments. They modeled traffic as a graph, allowing for the capture of intricate spatial dependencies that conventional LSTM models may overlook [21]. Despite their strengths, GNNs often require a comprehensive and high-quality representation of the network structure, which can be challenging to obtain in real-world scenarios. Furthermore, GNNs typically focus on spatial patterns and may not adequately capture temporal dynamics without integration with recurrent structures like LSTMs [23]. Reinforcement Learning (RL) was applied to traffic prediction and management, offering a dynamic approach to adaptively optimize predictions based on real-time feedback from traffic conditions [10]. RL-based methods can learn to adjust predictions based on the current state of traffic flow and congestion. However, RL methods often necessitate a substantial amount of training data to effectively learn optimal policies, which can be challenging to obtain in practice [12]. In addition, the results of the research progress based on Graph Neural Networks were discussed [8]. Attention-enhanced graph convolutional LSTM network (AGC-LSTM) was proposed to provide short-time forecasts [26]. Additionally, they may suffer from high variance in predictions, making them less reliable compared to more deterministic approaches.

In summary, although the mentioned methods singly achieved some success, the actual prediction effect of a single model is still poor due to the high randomness of traffic flow, which makes predictions difficult. Therefore, to address the issues, this paper tries to improve the prediction accuracy of highway traffic flow by combining optimization with a deep learning model, and a decomposition approach.

3. Feature Extraction for Data of Highway Traffic Flow

3.1 Nonlinear Principal Component Analysis (NPCA)

The article employs NPCA to examine the key impact indicators of highway traffic flow. By substituting the original numerous significant indicators with a computed few principal components, the analysis aims to diminish both the dimensionality and correlation issues of the indicators. This approach not only lessens the number of attributes required for the neural network but also enhances its responsiveness to these indicators, thereby improving the prediction accuracy. The detailed steps are outlined below:

Step 1: Generate the original data matrix. Let there are m main indicators affecting the traffic volume of the highway, and the sample data consists of n years, then the original data matrix X is given in Equation (1):

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} \tag{1}$$

where x_{ij} represents the observed value of the jth indicator in the ith year.

Step 2: Data log-centered transformation. The matrix X is log-centered to obtain the transformed matrix $\boldsymbol{Y} = \left(y_{ij}\right)_{n\times m}$, where y_{ij} represents the observed value of the jth column of the ith row after the log-centered transformation is run and denoted as

$$y_{ij} = \lg\left(x_{ij}\right) - \frac{\sum_{m}^{j=1}\lg\left(x_{ij}\right)}{m} \tag{2}$$

Step 3: Determine the covariance matrix. Calculate the covariance matrix $\boldsymbol{S} = \left(S_{ij}\right)_{m\times m}$ for matrix Y, where S_{ij} denotes the value of the ith row and the jth column in the covariance matrix S.

Step 4: Determine the non-linear principal components. Firstly, solve the eigenvalue λ_j of the covariance matrix S, so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ are obtained and then find its corresponding normalized eigenvectors (a_1, a_2, ···, a_m) to form the eigenmoment matrix $\boldsymbol{A} = \left(a_{ij}\right)_{m\times m}$, where a_{ij} represents the value of the ith row and the jth column in the eigenmoment matrix $\boldsymbol{A} = \left(a_{ij}\right)_{m\times m}$.

The ith nonlinear principal component Y_i is denoted by

$$Y_i = \sum_{m}^{j=1}\left[\lg\left(x_{ij}\right) a_{ij}\right] \tag{3}$$

Step 5: Determine the number of non-linear principal components. Calculate the explained variance contribution τ_j and cumulative contribution C of the jth eigenvalue, denoted as

$$\tau_j = \frac{\lambda_j}{\sum_{j=1}^{m} \lambda_j} \tag{4}$$

$$C = \frac{\sum_{j=1}^{c} \lambda_j}{\sum_{j=1}^{m} \lambda_j} \tag{5}$$

According to the cumulative contribution rate $C \geq 0.9$, Equation (3) is used to determine the $c(c \leq m)$ principal components of the indicators representing highway traffic volume. Thus, the first c non-linear principal components, which are fewer than the original number of indicators, are used to comprehensively reflect the information of the original m indicators.

3.2 Variational Modal Decomposition (VMD)

Variational Mode Decomposition (VMD) is an innovative, non-recursive technique for decomposing time series data. This method simplifies the initial prediction challenge by breaking down the time series into several subsequences, allowing for independent modeling of each. The VMD operates like Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) approaches, as it adaptively divides the input sequence into K components [13].

However, unlike the EMD and EEMD, which may retain multiple frequency components within each sequence, the VMD is advantageous because it ensures that the resultant decomposition of each subsequence is characterized by a restricted bandwidth and a distinct center frequency. This uniqueness arises from the constraints placed by the VMD during the calculation of the K subsequences, which continuously updates and fine-tunes the bandwidth and center frequency of each component. The computation for the bandwidth of the sequence component u_k is carried out through the following steps: (i) apply the Hilbert transform to derive the corresponding analytic signal u_k, yielding a one-sided spectrum; (ii) utilize the exponential operator to modulate each component's spectrum into the baseband; and (iii) determine the bandwidth of each mode by calculating the squared norms of the gradients of the demodulated signals. Consequently, the constrained variational problem can be expressed as follows:

$$\begin{cases} \min\limits_{u_k, w_k} \left\{ \sum_{k=1}^{K} \left\| \partial \left[\left(\delta(t) + \frac{\mathrm{j}}{\pi t} \right) * u_{k,t} \right] \exp\left(-\mathrm{j} w_k t\right) \right\|_2^2 \right\} \\ \text{s.t.} \sum_{k=1}^{N} u_{k,t} = x_{\text{init},t} \end{cases} \tag{6}$$

where $u_{k,t}$ denotes the kth sequence component at time t. w_k denotes the center frequency of the kth sequence component. δ (t) denotes the Dirac δ-distribution. (*) denotes the convolution operator. $x_{\text{init},t}$ denotes the original sequence at time t. (s.t.) refers to the constraint part of the optimization problem in Equation (6), which indicates that the sum of the several sequence components is equal to the original sequence.

The VMD algorithm is capable of breaking down highly complex original runoff data into several components of reduced complexity [20]. This transformation enables the prediction model to more effectively capture essential features and minimize the impact of noise. Nonetheless, a challenge associated with the VMD algorithm is to select appropriate values for the decomposition parameter K and the penalty factor α. Using unsuitable parameters for K and α could adversely affect the accuracy of the final predictions. In the research, we propose a multi-strategy enhancement of the Hunter-Prey Optimization, which is an Improved Hunter-Prey Optimization (IHPO) algorithm aimed at identifying the optimal values for both K and α.

3.3 IHPO

3.3.1 Standard HPO

The Hunter-Prey Optimization (HPO) algorithm, introduced in 2022, is an intelligent optimization technique inspired by the hunting behaviors of predators pursuing their targets, consisting of 2 populations: hunters and prey. Hunters focus on attacking prey that are located at greater distances from the main prey population. In response, hunters adjust their positions toward the remote prey, while the prey move toward safer locations. When compared to other similar algorithms, the HPO demonstrates substantial exploration and exploitation capabilities, improving its convergence accuracy and global optimization effectiveness. The principles underlying this algorithm are outlined as follows:

1 Initialize the population

$$x_i = rand(1,d)\cdot(UB - LB) + LB \,, \tag{7}$$

where x_i characterizes the position of the hunter or prey, LB, UB and d characterizes the minimum, maximum, and dimension of the variable.

2 Search and utilization mechanisms

$$\begin{cases} \boldsymbol{P} = \vec{\boldsymbol{R}}_1 < E_q \\ \boldsymbol{I}_{\mathrm{DX}} = (\boldsymbol{P} == 0) \\ Z = R_2 \cdot \boldsymbol{I}_{\mathrm{DX}} + \vec{\boldsymbol{R}}_3 \cdot (\sim \boldsymbol{I}_{\mathrm{DX}}) \end{cases} , \tag{8}$$

where $\boldsymbol{P}$ represents a random vector representing the variable numbers. $\vec{\boldsymbol{R}}_1$ and $\vec{\boldsymbol{R}}_3$ represent random vectors in the range [0,1]. R_2 represents random numbers in the range [0,1]. Z represents the adaptive parameter. $\boldsymbol{I}_{\mathrm{DX}}$ represents the index number of the vector satisfying the condition. E_q represents the balance parameter of exploration and exploitation as expressed in Equation (9)

$$E_q = 1 - \frac{0.98 \cdot T_i}{T_{\max}} \,, \tag{9}$$

where T_i and $T_{\max}$ represent the current number of iterations and the maximum number of iterations.

Calculate the average value μ for all positions as:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \vec{x}_i \,. \tag{10}$$

Compute the Euclidean distance $D_{(i)}$

$$D_{(i)} = \sqrt{\sum_{j=1}^{d}\left(x_{i,j} - \mu_j\right)^2} \,, \tag{11}$$

where $x_{i,j}$ and μ_j represent the individual position of the population and the mean value of the position.

If the average position and maximum distance are calculated once per iteration, the algorithm will experience a delayed convergence, so a decreasing mechanism is proposed:

$$L_{\mathrm{best}} = \mathrm{ROUND}(E_q \cdot N) \,, \tag{12}$$

where N represents population size.

Calculate the prey position again:

$$\vec{P}_{\mathrm{pse}} = \vec{x}_i \mid D\left(L_{\mathrm{best}}\right) . \tag{13}$$

When a prey is attacked, it will try to escape to a safe location to enhance its chances of survival, and the hunter may choose another prey, thus the hunter and the prey are selected dynamically.

$$\begin{aligned} x_i(t{+}1) &= x_i(t) + 0.5\{[2E_q Z P_{\mathrm{pos}(j)} - x_i(t)] + \\ &+ [2(1-E_q) Z\mu(j) - x_i(t)]\}, R_5 < \beta \end{aligned} \tag{14}$$

$$x_i(t{+}1) = B_g + E\,Z\cos(2\pi R_4)\cdot[B_g - x_i(t)],\ R_5 > \beta \,, \tag{15}$$

where B_g represents global optimum. $x_i(t)$ and $x_i(t+1)$ represent the current position and next iteration position of hunter/prey. $R_5 \in [0,1]$ represent random numbers. β represents a regulation parameter. Equations (14)-(15) represent the updating expressions of the hunter position and the prey position.

In the HPO framework, various updating methods are chosen randomly based on a probabilistic parameter β. The rules governing the hunters are primarily focused on global search strategies, whereas the updating approach for the prey leans towards local exploitation. During the global search phase, hunters progressively shift toward the average position of the entire population. Conversely, in the local exploitation phase, prey moves randomly within the vicinity of the global optimum. This updating behavior tends to favor local exploitation, which often results in an inadequate mechanism to escape from local optimal solutions. After multiple iterations are run, if the algorithm does not alter the location of the local optimum, the population can quickly converge to this local optimum, leading to a stagnation phase in the algorithm's performance. To address the limitations of the conventional HPO, the research introduces a multi-strategy improved HPO algorithm.

3.3.2 Improvements in the Algorithm

(1) The set of good points to initialize populations

In the HPO, initial populations are generated by randomization. Since the random numbers generated by the system are not completely random, uneven distribution of populations may occur. To prevent this type of initialized population from being too aggregated, which may cause the algorithm to fall into local optimum prematurely, the IHPO adopts the set of good points to initialize the population. Let G_s be a unit cube in s-dimensional space, if $r \in G_s$, the shape is: $P_n(k) = \{r_1(k), r_2(k), \cdots, r_s(k)\}, 1 \le k \le n$, n represents the number of points, and its deviation $\sigma(n)$ satisfies $\sigma(n) = E_q(r,\varepsilon)n^{\varepsilon-1}$, where $E_q(r,\varepsilon)$ is a constant related to r and ε only, then $P_n(\varpi)$ is called the good point set and r is the good point. Initializing the population by the set of good points allows the initial population to be evenly distributed in the search space, thus covering more search range. The population is initialized as in Equation (16).

$$x_i(j) = \left(UB_j - LB_j\right) r_j^i \cdot \varpi + LB_j \ . \tag{16}$$

(3)Search optimization

In the HPO, the population is dynamically and gradually transitioned from a global search to a local exploitation phase by using a nonlinear stochastic sinusoidal parameter γ. γ calculated as:

$$\gamma = 2 \times \left[\sin(\omega\pi t / T) + 1\right] \times \left(2r_2 - 1\right). \tag{17}$$

In this study, $r_2 \in [0,1)$ represents a random variable, while ω is utilized to modify the slope of the nonlinear curve, specifically set to $\omega = -0.2$. Additionally, within the HPO framework, the parameter ρ governs the updating step for the population. Equation (18) facilitates the adjustment of ρ, which linearly decreases the value of ρ from 1 to 0.02. If ρ decreases too rapidly during the initial phase, it may result in insufficient global search capabilities for the algorithm. Conversely, if the decrease is too slow in the later phases, it can compromise the accuracy of local development. The enhanced parameter ρ is updated as indicated in Equation (18):

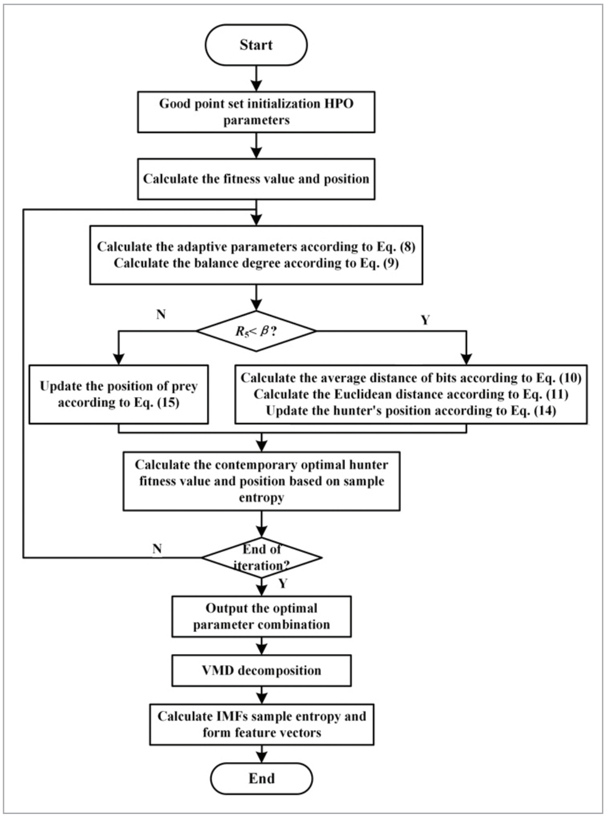
$$\rho = 1 + \cos(t\pi / T) \, . \tag{18}$$

3.4 IHPO-VMD-Based Feature Extraction

In the process of the VMD decomposing time series data related to highway traffic flow, the predefined number of modal decompositions and the penalty factor play a crucial role in determining the effectiveness of the decomposition. An inappropriate number of decompositions can lead to over-decomposition or under-decomposition, which may result in the intrinsic mode function (IMF) components falling outside the suitable bandwidth if the penalty factor is set incorrectly. Consequently, this paper employs the IHPO algorithm to fine-tune the parameter combination $[K, \alpha]$ for the VMD. Figure 1 presents its flow chart.

Figure 1

The feature extraction process steps are based on the IHPO-VMD algorithm.



Step 1: Initialization of Parameters: Specify the population size, the number of iterations, and the initial positions of the prey, and establish the upper and lower bounds. Additionally, input the time series data related to highway traffic flow.

Step 2: Fitness Function: Utilize sample entropy as the fitness function to assess the effectiveness of the VMD decomposition. This involves calculating the fitness score and determining the initial optimal position for the hunter.

Step 3: Update the adaptive and equilibrium parameters to update the hunter position xi(t+1) when $R_5 < \beta$ and xi(t+1) shows the prey position when $R_5 > \beta$.

Step 4: Calculate the contemporary hunter fitness, filter the global optimal fitness score and the optimal individual position, and loop iteration until the end to output the optimal parameter combination $[K, \alpha]$.

Step 5: Perform the VMD decomposition of the time series data according to the optimized parameter combinations to obtain the final IMF components.

Step 6: The sample entropy of each IMF component is resolved, and the eigenvectors are formed together with the quantized values of the factors influencing highway traffic.

4. Feature Classification Based on LSTM-Informer Model

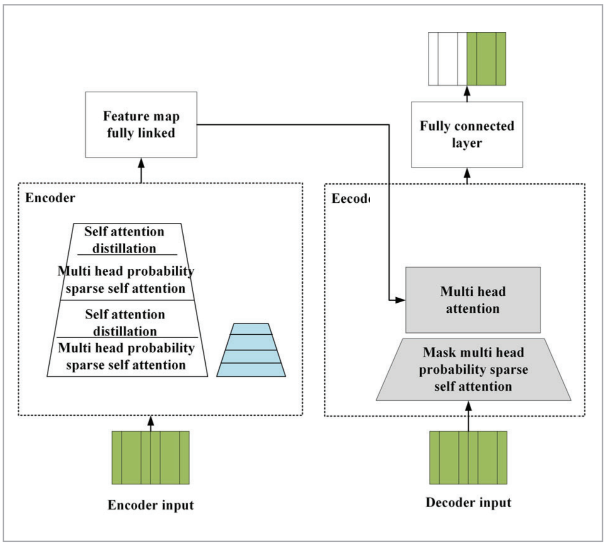
4.1 Principles of Informer

The Informer model leverages a neural network framework for predicting highway traffic flow, effectively capturing temporal correlations in traffic data. It employs a probabilistic sparse self-attention mechanism to identify the most significant queries, which helps lower the computational complexity associated with the self-attention matrix. Additionally, self-attention distillation is utilized to reduce the dimensionality and the number of parameters in the network, addressing issues related to excessive memory consumption caused by multilayer network stacking [7].

Furthermore, the Informer model generates all predicted values using a generative decoder in a single computation, enhancing prediction speed and mitigating the risk of cumulative error growth that can occur when predictions are iteratively accumulated. Figure 2 depicts the structure of the Informer model featuring an encoder that processes numerous long sequence inputs, which undergo 2 main operations [17]: probabilistic sparse self-attention and self-attention distillation. These operations extract the most critical attention signals, significantly shrinking the network size while employing layer stacking to bolster robustness. This results in an attention feature map that is then sent to the decoder's multi-head attention module.

Figure 2

The components of the structure of the proposed Informer model



The decoder accepts long sequences, filling the predicted values with zeros. The inputs to the decoder are masked using multi-head probabilistic sparse self-attention, after which multi-head attention calculates a weighted combination with the attention feature map produced by the encoder. This process allows for direct prediction of multi-step output results. Finally, the decoder's output is processed through a fully connected layer, transforming the high-dimensional output of the Informer model into a format that aligns with the specific prediction requirements.

4.2 LSTM-Informer Model

To enhance prediction capabilities, conventional self-attention mechanisms necessitate quadratic dot product calculations and consume substantial memory, which significantly hampers the speed of output predictions for lengthy sequences. In similarity computations, conventional self-attention often struggles with overly focusing on irrelevant details while neglecting essential information. To address these issues and accelerate the prediction process for long sequences, probabilistic sparse self-attention is employed. Initially, the input sequence matrix is transformed into 3 distinct matrices: the query, the key, and the value es expressed in Equation (19).

$$\begin{cases} \boldsymbol{O} = \boldsymbol{I}\boldsymbol{W}_O \\ \boldsymbol{P} = \boldsymbol{I}\boldsymbol{W}_P \\ \boldsymbol{Q} = \boldsymbol{I}\boldsymbol{W}_Q \end{cases}, \tag{19}$$

where $\boldsymbol{O}$, $\boldsymbol{P}$, $\boldsymbol{Q}$ represent the query, key, and value matrices, respectively. $\boldsymbol{I}$ denotes the input matrix.

$\boldsymbol{W}_{\boldsymbol{O}}$, $\boldsymbol{W}_{\boldsymbol{P}}$, $\boldsymbol{W}_{\boldsymbol{Q}}$ show the weight matrices, respectively.

Secondly, the key matrix $\boldsymbol{P}$ is sampled to obtain the sampled matrix $\overline{\boldsymbol{P}}$, and then the M-value regarding $\overline{\boldsymbol{P}}$ is found for the ith row o_i of the query matrix $\boldsymbol{O}$.

$$M\left(o_i, \overline{\boldsymbol{P}}\right) = \max_j \left\{ \frac{o_i \overline{\boldsymbol{p}}_j^{\mathrm{T}}}{\sqrt{d}} \right\} - \frac{1}{L_p} \sum_{j=1}^{L_p} \frac{o_i \overline{\boldsymbol{p}}_j^{\mathrm{T}}}{\sqrt{d}}, \tag{20}$$

where $M\left(o_i, \overline{\boldsymbol{P}}\right)$ denotes an indicator for evaluating the importance of o_i. The larger the value of $M\left(o_i, \overline{\boldsymbol{P}}\right)$, the more important o_i is. $\overline{\boldsymbol{p}}_j^{\mathrm{T}}$ represents the jth line of $\overline{\boldsymbol{P}}$. $\sqrt{d}$ denotes the scaling factor and d represents the number of columns of $\boldsymbol{P}$, L_p denotes the number of rows of $\boldsymbol{P}$. Then, u o_i is found with the largest value of M, these u o_i are formed into a new query matrix $\overline{\boldsymbol{O}}$, and the score value $o_i \overline{\boldsymbol{p}}_j^{\mathrm{T}} / \sqrt{d}$ about $\boldsymbol{P}$ is calculated for $\overline{\boldsymbol{O}}$.

Finally, the probabilistic sparse self-attention is obtained as follows:

$$A(\boldsymbol{O}, \boldsymbol{P}, \boldsymbol{Q}) = \mathrm{Softmax}\left(\frac{\overline{\boldsymbol{O}} \boldsymbol{P}^{\mathrm{T}}}{\sqrt{d}} \right) \boldsymbol{Q}. \tag{21}$$

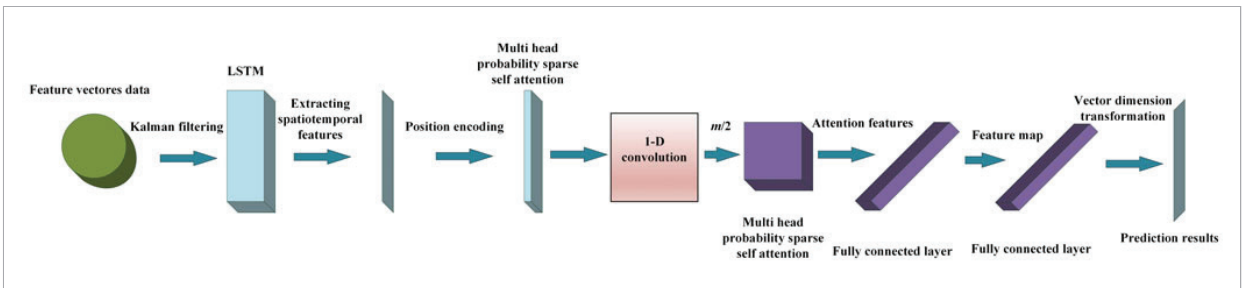
Utilizing multiple probabilistic sparse self-attention mechanisms can enhance the model's capability to handle long sequence prediction tasks by allowing parameter sharing across these mechanisms while maintaining a low computational cost.

$$A_{\mathrm{M}}(\boldsymbol{O}, \boldsymbol{P}, \boldsymbol{Q}) = \mathrm{Concat}\left(H_1, H_2, \cdots, H_c\right). \tag{22}$$

$A_{\mathrm{M}}(\boldsymbol{O},\boldsymbol{P},\boldsymbol{Q})$ represents the multi-head probabilistic sparse self-attention. H_c denotes the c-th "head", $H_c = A(\boldsymbol{O}_c, \boldsymbol{P}_c, \boldsymbol{Q}_c)$. $\boldsymbol{O}_c, \boldsymbol{P}_c, \boldsymbol{Q}_c$ are the $\boldsymbol{O}, \boldsymbol{P}, \boldsymbol{Q}$ matrices of the c-th "head", respectively.

The LSTM-Informer model is developed, as illustrated in Figure 3. Initially, the feature vector data representing the time series of highway traffic flow is fed into the LSTM model. Following in-depth feature learning and position encoding, a multi-head probabilistic sparse self-attention mechanism focuses on the evolving features of the vector. Subsequently, maximum pooling and one-dimensional convolution are applied to remove redundant combinations from the final output feature maps. Finally, another application of multi-head probabilistic sparse self-attention emphasizes the important features and the decoder of the Informer model is adjusted to utilize a fully connected layer, yielding the ultimate predictions.

Figure 3
The structure of the LSTM-Informer model



4.3 Highway Traffic Prediction Process Based on LSTM-Informer Model

The procedure for predicting highway traffic flow using the LSTM-Informer model is outlined as follows:

1. Initially, the feature vector sequence generated by IHPO-VMD undergoes Kalman filtering to remove redundancy, followed by normalization preprocessing.
2. Next, the data is fed into the LSTM model to extract temporal features. These features are then input into the Informer model's encoder, where positional encoding is applied. The model employs a multi-criteria probabilistic sparse self-attention mechanism to focus on the evolving characteristics, followed by maximum pooling and one-dimensional convolution to eliminate redundant combinations from the final output feature maps.
3. Lastly, the multi-head probabilistic sparse self-attention mechanism is applied again to concentrate on the feature vectors. Additionally, the decoder of the Informer model is modified to incorporate a fully connected layer, which produces the final predictions.

The pseudo-code of the LSTM-Informer is shown in Table 1.

Table 1
The pseudo-code of the LSTM-Informer

Model 1: The LSTM-Informer
Input: Time series data X
Initialize parameters:
LSTM parameters (hidden layer size, learning rate, number of layers, etc.)
Informer pa rameters (sequence length, number of attention heads, etc.)
Preprocess data:
Normalize time series data X
Create training and testing datasets
Generate sequences of fixed length for LSTM input
Define LSTM model:
Initialize LSTM layer(s)
Initialize the output layer (Dense layer)
Train LSTM model:
For each epoch:
For each batch in training data:
Forward pass:
Input sequence into LSTM layers
Obtain LSTM output (hidden states)
Compute loss (e.g., Mean Squared Error) between predicted and actual values
Backward pass:
Compute gradients
Update weights using an optimizer (e.g., Adam)
(Optional) Validate on the validation set
Prepare LSTM output for Informer:
Extract the final hidden states from the LSTM
(Optional) Concatenate LSTM outputs with additional features
Define Informer model:
Initialize Informer encoder with multi-head attention layers
Initialize decoder layers for prediction
Train Informer model:
For each epoch:
For each batch in training data:
Forward pass:
Input LSTM outputs into the Informer encoder
Apply attention mechanism to extract features
Pass through decoder layers to generate predictions
Compute loss (e.g., Mean Squared Error) between predicted and actual values
Backward pass:
Compute gradients
Update weights using an optimizer (e.g., Adam)
(Optional) Validate on the validation set
Make predictions:
For test data:
Input test sequences into the trained LSTM
Obtain the LSTM outputs
Input the LSTM outputs into the trained Informer model
Generate final predictions
Output: Predicted values Y
Evaluate model performance:
Calculate prediction accuracy (MAPE, RMSE)
Visualize a comparison between predicted and actual results

5. Experimental Analysis

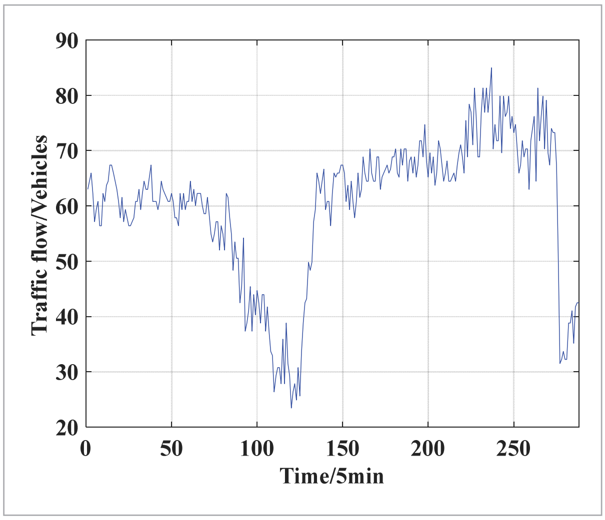
5.1 Experimental Environment and Data Processing

The dataset utilized in the experiments includes traffic flow data, vehicle trajectory information, weather conditions, and attributes related to road and network connectivity at a highway toll station, as provided by KDD CUP 2017. The data is sampled at 5-minute intervals, resulting in daily generated 288 entries. The vehicle trajectory data encompasses travel information from intersections A to C leading to the toll station. The traffic flow records document vehicles passing through the toll gates, with only those entering the highway allowed at toll gate No. 2; here, "0" indicates outbound vehicles while "1" signifies inbound ones. Weather data comprises features such as humidity levels and rainfall. The data sampling period spans from September 19, 2016, to October 24, 2016. To maintain higher prediction accuracy, holiday data was excluded, as traffic patterns during the National Day holiday significantly differ from regular periods. Missing data was addressed using linear interpolation, while the Min-Max normalization method was applied to standardize the dataset.

Figure 4 shows the map of the traffic flow data for a certain day. The curve illustrates that the daily traffic flow at a tollgate shows a strong correlation between work and rest, showing a clear morning and evening peak, with a relatively large flow during the day and an overall smaller flow at night. The NPCA was performed on the correlation factors of highway traffic flow data to reduce the dimensionality of the data to filter out the 3 main factors of adjacent toll intersections, vehicle models, and air temperature.

Figure 4

A map of traffic flow data for a certain day



In this paper, the experiments are performed on a computer model Intel(R) Core(TM) i9-12900H CPU @ 5.0GHz with 16GB of RAM, and the simulation software used is Matlab 2023A.

5.2 Evaluation Metrics

To assess the accuracy of the predictions, the mean absolute percentage error (MAPE) and root mean square error (RMSE) are utilized for evaluation. Equations (23)-(24) provide formulas.

$$MAPE = \frac{\sum_{i=1}^{N} \frac{\left| X_{\text{predicted}} - X_{\text{real}} \right|}{X_{\text{real}}} \times 100\%}{N} \tag{23}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(X_{\text{predicted}} - X_{\text{real}} \right)^2}{N}}, \tag{24}$$

where $X_{\text{predicted}}$ and X_{real} represent the predicted value and the actual score of highway traffic flow, respectively.

5.3 Analysis of Experimental Results

Due to the randomness and volatility of the highway traffic flow data, when using the VMD method to decompose its data, the IHPO is selected to optimize the system to determine the optimal score of the number of decomposition models K=7. The scores of the other parameters are set as follows: Penalty Factor α=92867, so the highway traffic flow in Figure 4 will be decomposed by the VMD decomposition to attain the decomposition results as shown in Figure 5. The sample entropy of IMFs obtained by the VMD decomposition is resolved and the feature vectors are formed together with other principal components, which are employed to input into the LSTM-Informer model.

The highway traffic flow for a specific day in the test set is now predicted. The feature vector, derived from the extracted data, serves as the input for the LSTM Informer prediction model. This process results in the freeway traffic flow prediction curve illustrated in Figure 6.

Figure 6 depicts the predictions by the proposed method closely aligned with the actual scores, exhibiting only minor deviations in the peak and through values of the highway traffic flow. To demonstrate the efficacy of the proposed approach, the MAPE and RMSE scores are compared with those of VMD-LSTM, VMD-Informer, VMD-LSTM-Informer, and HPO-VMD-LSTM-Informer. Table 2 presents the results.

Table 2

The comparison of the experimental results

Models	MAPE (%)	RMSE
VMD-LSTM	28.76	19.84
VMD-Informer	23.45	17.65
VMD-LSTM-Informer	19.57	12.64
HPO-VMD-LSTM-Informer	14.64	8.64
IHPO-VMD-LSTM-Informer	8.09	2.84

Table 2 presents that it is evident that the IHPO-VMD-LSTM-Informer model achieves the highest prediction accuracy among the 4 assessed models. Specifically, the results of the VMD-LSTM-Informer outperform those of both VMD-LSTM and VMD-Informer. This suggests that the LSTM-Informer demonstrates superior feature classification capabilities for highway traffic flow data. Moreover, the incorporation of the Informer effectively addresses the limitations of the LSTM in extracting features from time-series data, leading to a higher prediction accuracy for the LSTM-Informer when compared to employing the LSTM or Informer alone. Additionally, the IHPO approach introduced in this paper enhances the learning performance of the LSTM-Informer, addressing the drawbacks of the conventional HPO methods and significantly boosting prediction accuracy. In conclusion, the IHPO-VMD-LSTM-Informer is well-suited for predicting highway traffic flow, offering both higher accuracy and more robust practical applications in engineering settings.

Figure 5

The decomposition of the highway traffic flow data by the VMD

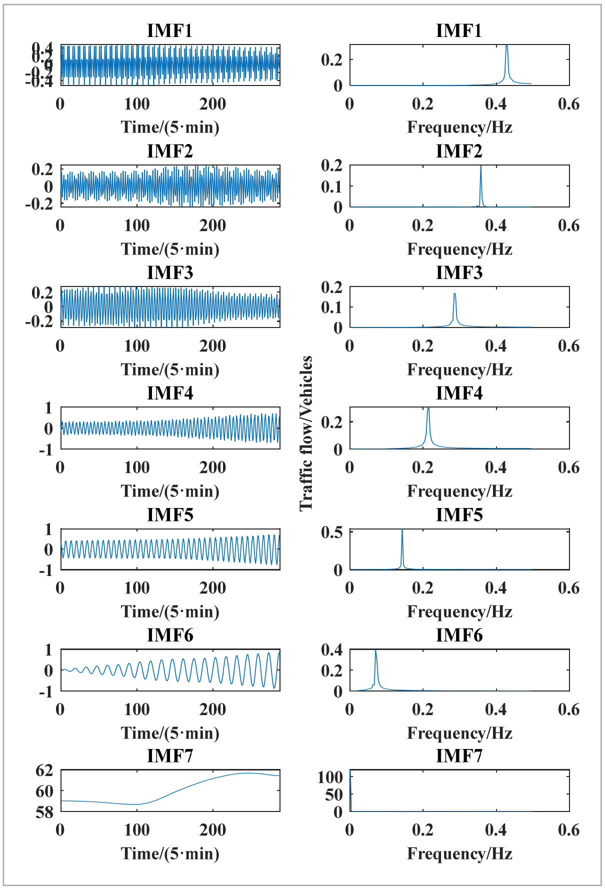
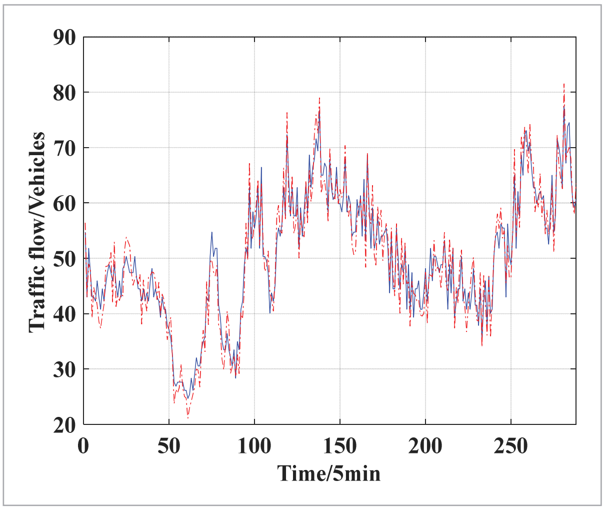


Figure 6

The predictions of traffic flow in 5-minute intervals



6. Discussion

Decomposition methods, commonly implemented in time series analysis, often encounter 2 significant challenges: mode mixing and the boundary problem [4, 24, 16, 27]. Mode mixing occurs when distinct frequency components are inadequately separated within the intrinsic mode functions (IMFs), leading to the mixing of high-frequency fluctuations such as sudden vehicle surges in traffic flow with long-term trends like average daily traffic patterns. This amalgamation complicates the analysis, as it becomes challenging to isolate and interpret these critical components effectively. Also, the resulting IMFs can misrepresent the underlying dynamics of the system, potentially leading to inaccurate predictions.

On the other hand, the boundary problem arises from the inherent limitations of decomposition methods that rely heavily on local signal properties. These methods often struggle with the edges of the time series, resulting in artifacts or distortions at the beginning and end of the decomposed signals. In the context of highway flow traffic prediction, such inaccuracies can adversely affect the reliability of predictions at critical points, such as the start and end of a forecasting period. If the boundary effects are not managed properly, they can introduce significant errors, undermining the overall model's efficacy.

To address these issues, the proposed approach incorporates several strategies. First, we employ advanced decomposition techniques that utilize adaptive algorithms, which are designed to mitigate mode mixing by enhancing the separation of frequency components. For instance, using the VMD can help enhance the distinction between IMFs, ensuring that each mode captures a specific frequency range without overlap.

In addition, to resolve the boundary problem, we adopt the IHPO to realize the adaptive optimization of the VMD and reduce the influence of the boundary effect, which also makes the learning of the LSTM-Informer more thorough and the training more adequate. These strategies help ensure that the decomposed components are less prone to distortion, resulting in more accurate and reliable predictions of the entire time series. By effectively addressing these common problems, the proposed approach improves the robustness and accuracy of traffic flow prediction, which ultimately leads to better decision-making and resource allocation in traffic management.

7. Conclusion

To address the challenges of fluctuating highway traffic flow and inadequate prediction accuracy, we have developed a new prediction method called the IHPO-VMD-LSTM-Informer to predict highway traffic flow. Highways are critical transportation means and a smooth traffic flow is required especially for short periods covering an interval from 1 to 15 minutes when abrupt changes occur. To manage those short fluctuations, more data-oriented approaches are required to deal with the intrinsic characteristics of highway traffic flow such as nonlinearity, high uncertainty, and non-stationarity.

To predict more accurately, the proposed method IHPO-VMD-LSTM-Informer is composed of 4 different approaches. Initially, we perform dimensionality reduction using the NPCA, apply the VMD to decompose highway traffic flow data, and incorporate the IHPO algorithm to effectively tackle the issues of both excessive and insufficient decomposition, which are often the result of manual parameter tuning. This approach facilitates the feature extraction from time-series data. Furthermore, the proposed model combines the LSTM network with the Informer model, ultimately achieving accurate predictions for freeway traffic flow. Experimental results demonstrate that the proposed method outperforms other prediction models in accuracy, thereby offering valuable insights for the intelligent management of highways.

The limitations of the research can be summarized as follows: 1. the optimization method of the IHPO is heuristic. The performance of other heuristic methods should be checked. 2. In addition to using MAPE and RMSE, other prediction metrics such as R^2 should be used. 3. A benchmark data set or more data sets should be used to better assess the proposed model.

In our future work, we plan to implement several robustness checks to enhance the reliability of the proposed model. Specifically, we will introduce synthetic noise and run simulations with missing data to assess how the model performs under uncertain conditions. This will allow us to gauge its robustness and generalizability in practical applications. Additionally, we will conduct a comprehensive sensitivity analysis of the model's hyperparameters, including the number of layers and learning rates. By systematically varying these parameters and observing the resulting impacts on model performance, we aim to verify the consistency and stability of the proposed model across different configurations. This dual approach of robustness testing and sensitivity analysis will provide valuable insights into the model's resilience and guide future refinements.

Funding

No funding was received for this study.

References

1. Abadi, A., Rajabioun, T., Ioannou, P. A. Traffic Flow Prediction for Road Transportation Networks with Limited Traffic Data. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2), 653-662.
2. Bellini, P., Bilotta, S., Palesi, A. L. I., Nesi, P., Pantaleo, G. Vehicular Traffic Flow Reconstruction Analysis to Mitigate Scenarios with Large City Changes. IEEE Access, 2022, 10, 131061-131075. https://doi.org/10.1109/ACCESS.2022.3229183
3. Chan, K. Y., Dillon, T. S., Singh, J., Chang, E. Traffic Flow Forecasting Neural Networks Based on Exponential Smoothing Method. 2011 6th IEEE Conference on Industrial Electronics and Applications, Beijing, China, 2011, 376-381. https://doi.org/10.1109/ICIEA.2011.5975612
4. Fearn, D. R. Eigensolutions of Boundary Value Problems Using Inverse Iteration. Journal of Computational and Applied Mathematics, 1991, 34(2), 201-209. https://doi.org/10.1016/0377-0427(91)90042-I
5. Feng, X., Ling, X., Zheng, H., Chen, Z., Xu, Y. Adaptive Multi-Kernel SVM With Spatial-Temporal Correlation for Short-Term Traffic Flow Prediction. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(6), 2001-2013. https://doi.org/10.1109/TITS.2018.2854913
6. Gao, Y., Zhou, C., Rong, J., Wang, Y., Liu, S. Short-Term Traffic Speed Forecasting Using a Deep Learning Method Based on Multitemporal Traffic Flow Volume. IEEE Access, 2022, 10, 82384-82395. https://doi.org/10.1109/ACCESS.2022.3195353
7. Jian, W., Li, J. P., Akbar, M. A., Haq, A. U., Khan, S., Alotaibi, R. M. SA-Bi-LSTM: Self Attention with Bi-Directional LSTM-Based Intelligent Model for Accurate Fake News Detection to Ensured Information Integrity on Social Media Platforms. IEEE Access, 2024, 12, 48436-48452. https://doi.org/10.1109/ACCESS.2024.3382832
8. Jiang, W., Luo, J., He, M., Gu, W. Graph Neural Network for Traffic Forecasting: The Research Progress. ISPRS International Journal of Geo-Information, 2023, 12(3), 100. https://doi.org/10.3390/ijgi12030100
9. Kashinath, S. A., Mostafa, S. A., Mustapha, A., Mahdin, H., Lim, D., Mahmoud, M. A. Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. IEEE Access, 2021, 9, 51258-51276. https://doi.org/10.1109/ACCESS.2021.3069770
10. Koller, D. R., et al. Reinforcement Learning for Traffic Signal Control: A Review. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(2), 1983-1998.
11. Kushchenko, L. E., Kushchenko, S. V., Novikov, A. N. The Application of Wavelet Analysis to Study the Characteristics of the Traffic Flow. 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2020, 1-4. https://doi.org/10.1109/FarEastCon50210.2020.9271410
12. Li, R., et al. A Survey on Reinforcement Learning for Traffic Control. Journal of Intelligent Transportation Systems, 2021, 25(5), 490-504.
13. Li, S., Zhao, J., Wu, Y., Bian, S., Zhai, G. Prior Frequency Information Assisted VMD Method for SBP Sonar Data Noise Removal. IEEE Geoscience and Remote Sensing Letters, 2023, 20, 1-5, Art. no. 7503905. https://doi.org/10.1109/LGRS.2023.3291714
14. Lihua, N., Xiaorong, C., Qian, H. ARIMA Model for Traffic Flow Prediction Based on Wavelet Analysis. The 2nd International Conference on Information Science and Engineering, Hangzhou, 2010, 1028-1031. https://doi.org/10.1109/ICISE.2010.5690910

15. Linting, M., Meulman, J. J., Groenen, P. J., van der Kooij, A. J. Nonlinear Principal Components Analysis: Introduction and Application. Psychological Methods, 2007, 12(3), 336-358. https://doi.org/10.1037/1082-989X.12.3.336

16. Madathil, S., Padannayil, S. K. MC-DMD: A Data-Driven Method for Blood Vessel Enhancement in Retinal Images Using Morphological Closing and Dynamic Mode Decomposition. Journal of King Saud University - Computer and Information Sciences, 2022, 34(8, Part A), 5223-5239. https://doi.org/10.1016/j.jksuci.2022.06.002

17. Selim, T., Elkabani, I., Abdou, M. A. Students Engagement Level Detection in Online E-Learning Using Hybrid EfficientNetB7 Together with TCN, LSTM, and Bi-LSTM. IEEE Access, 2022, 10, 99573-99583. https://doi.org/10.1109/ACCESS.2022.3206779

18. Stanulov, A., Yassine, S. A Comparative Analysis of Machine Learning Algorithms for the Purpose of Predicting Norwegian Air Passenger Traffic. International Journal of Mathematics, Statistics, and Computer Science, 2023, 2, 28-43. https://doi.org/10.59543/ijmscs.v2i.7851

19. Wang, T., Zhang, B., Wei, W., Damasevicius, R., Scherer, R. Traffic Flow Prediction Based on BP Neural Network. 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 2021, 15-19. https://doi.org/10.1109/AIID51893.2021.9456479

20. Wei, X., Feng, G., Qi, T., Guo, J., Li, Z., Zhao, D. Reduce the Noise of Transient Electromagnetic Signal Based on the Method of SMA-VMD-WTD. IEEE Sensors Journal, 2022, 22(15), 14959-14969. https://doi.org/10.1109/JSEN.2022.3184697

21. Wu, G., et al. Graph Neural Network for Traffic Forecasting: A Review. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(4), 1880-1893.

22. Yang, H., Yu, W., Zhang, G., Du, L. Network-Wide Traffic Flow Dynamics Prediction Leveraging Macroscopic Traffic Flow Model and Deep Neural Networks. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(5), 4443-4457. https://doi.org/10.1109/TITS.2023.3329489

23. Yu, X., et al. Spatial-Temporal Graph Convolutional Networks for Traffic Forecasting. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(5), 1930-1940.

24. Zhang, K., Lu, R., Zhang, L., Zhang, X., Yao, J., Li, R., Zhao, H. A Two-Stage Efficient History Matching Procedure of Non-Gaussian Fields. Journal of Petroleum Science and Engineering, 2016, 38, 189-200. https://doi.org/10.1016/j.petrol.2015.11.038

25. Zhang, S., Song, Y., Jiang, D., Zhou, T., Qin, J. Noise-Identified Kalman Filter for Short-Term Traffic Flow Forecasting. 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Shenzhen, China, 2019, 462-466. https://doi.org/10.1109/MSN48538.2019.00093

26. Zhang, Y., Xu, S., Zhang, L., Jiang, W., Alam, S., Xue, D. Short-Term Multi-Step-Ahead Sector-Based Traffic Flow Prediction Based on the Attention-Enhanced Graph Convolutional LSTM Network (AGC-LSTM). Neural Computing and Applications, 2024. https://doi.org/10.1007/s00521-024-09827-3

27. Zhou, B., Zhao, L. A Multi-Objective Decomposition Evolutionary Algorithm Based on the Double-Faced Mirror Boundary for a Milk-Run Material Feeding Scheduling Optimization Problem. Computers & Industrial Engineering, 2022, 177, 108385. https://doi.org/10.1016/j.cie.2022.108385