Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 5070504, 15 pages
https://doi.org/10.1155/2023/5070504

Research Article

A Hybrid Deep Learning Model for Link Dynamic Vehicle Count Forecasting with Bayesian Optimization

Chunguang He,[1,2] **Dianhai Wang**,[1] **Yi Yu**,[1,3] **and Zhengyi Cai**[1]

[1]*College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China*
[2]*School of Transportation and Logistics Engineering, Xinjiang Agricultural University, Urumqi, China*
[3]*Shanghai AI Laboratory, Shanghai 200232, China*

Correspondence should be addressed to Zhengyi Cai; caizhengyi@zju.edu.cn

Received 30 March 2022; Revised 7 September 2022; Accepted 24 November 2022; Published 7 February 2023

Academic Editor: Young-Jae Lee

The link dynamic vehicle count is a spatial variable that measures the traffic state of road sections, which reflects the actual traffic demand. This paper presents a hybrid deep learning method that combines the gated recurrent unit (GRU) neural network model with automatic hyperparameter tuning based on Bayesian optimization (BO) and the improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) model. There are four steps in this hybrid approach. First, the ICEEMDAN is employed to decompose the link dynamic vehicle count time series data into several intrinsic components. Second, the components are predicted by the GRU model. At the same time, the Bayesian optimization method is utilized to automatically optimize the hyperparameters of the GRU model. Finally, the predicted subcomponents are reconstructed to obtain the final prediction results. The proposed hybrid deep learning method is tested on two roads of Hangzhou, China. Results show that, compared with the 12 benchmark models, the proposed hybrid deep learning model achieves the best performance in link dynamic vehicle count forecasting.

1. Introduction

With the development of the social economy and urbanization, travellers' traffic demands increased rapidly. Traffic problems such as traffic congestion, environmental pollution, and economic losses have brought challenges to urban transportation management. Intelligent transportation technology promises to deal with these problems, and accurate and effective forecasting of traffic demands is a key step. Traditionally, indicators such as traffic flow are used to represent traffic demands. While in seriously congested areas, those indicators can hardly reflect actual traffic demands [1]. Traffic signal control systems, such as SCOOT (Split Cycle Offset Optimization Technique) and SCATS (Sydney Coordinated Adaptive Traffic System), failed to work properly due to the inaccurate traffic demand estimation in the congested period [2, 3].

Compared with the indicator of traffic flow, the link dynamic vehicle count (LDVC) refers to the vehicle number on a specific road that describes the space occupancy rate of roads [4–6] and can reflect traffic demand more precisely [7, 8]. Accurate and real-time prediction of the link dynamic vehicle count can also provide a reliable basis for the network-wide traffic signal control strategy and optimization [1, 2, 9]. In recent years, thanks to continuous investments in intelligent transportation systems, various sensors have been deployed and a large amount of real-time traffic data can be collected. The link dynamic vehicle count data are effectively calculated under the new data environment with mature technology [10, 11]. However, the critical issues and challenges unaddressed in forecasting link dynamic vehicle count in the following aspects: (a) LDVC is often disturbed by stochastic factors. For instance, LDVC experiences a sudden increase and decrease when traffic flow becomes congested or traffic incidents occur, such as accidents and temporary traffic control measures. (b) Although selecting the "best" model among a set of baselines is significant, a better alternative is to consider the strength and robustness

of the prediction results. By decomposing traffic data into subsequences, it can help capture both the common tendency and some changes in traffic flow to improve the prediction accuracy. (c) Most machine learning-based methods can mine the nonlinear profile of link vehicle count, but overfitting issues may occur. It indicates that the model extracts noise in the training data as a feature of the data itself, which in turn degrades performance in the test dataset.

Machine learning approaches are widely used in traffic forecasting. Researchers obtained some progress in prediction algorithms, model fusion, and temporal and spatial characteristics of traffic data. However, there are still some crucial issues and challenges. (a) In traffic research field, most forecasting models focus on traffic flow [12–15], traffic speed [16–18], or travel time [19–21]. Few forecasting models focus on the prediction of the LDVC, let alone considering the data characteristics of the LDVC in the prediction model. (b) The LDVC has nonlinear and stochastic characteristics, such as the random change due to traffic congestion and weather factors. Simultaneously, due to the signal light control, weekday commuting, the LDVC shows the characteristics of long short-term periodic changes in signal cycle, day, and week pattern. (c) The excellent performance of deep learning methods in traffic forecasting is inseparable from the model hyperparameter tuning efficiently and appropriately. The prediction model represented by deep learning requires many hyperparameter tuning, which is time consuming and laborious but hard to obtain suitable hyperparameters.

In response to these challenges, this paper proposes a hybrid ICEEMDAN-GRU-BO forecasting model for the link dynamic vehicle count. The hybrid model fuses the ICEEMDAN based on hyperparameter tuning with Bayesian optimization (BO) and the improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN). First, the ICEEMDAN is utilized to decompose the link dynamic vehicle count data into subcomponents. Thus, the decomposed components reduce the stochastic characteristics and become more regular and suitable for prediction. Second, the GRU models are applied to predict those components by considering the feature of long short-term periods. Simultaneously, the BO is used to automatically optimize the hyperparameters of the GRU models to deal with the challenge problem of the hyperparameter tuning. Finally, the predicted subcomponents are reconstructed to obtain the final prediction results.

The contributions of this paper can be summarized as follows: (1) For the first time, a hybrid ensemble decomposition deep learning prediction framework is proposed to predict the link dynamic vehicle count to improve the prediction accuracy and reduce the prediction time. (2) Aiming at tackling the challenges mentioned above, we present a novel approach by integrating the GRU with ICEEMDAN. The data decomposition method ICEEMDAN is used to reveal the nonlinearity and stochastic characteristic of the link dynamic vehicle count. We propose a short-term prediction method based on GRU model to effectively capture the long short-term period features. (3) To avoid overfitting issues and address the considerable time consuming and laborious in hyperparameter tuning, the hyperparameters in our model are automatically and efficiently tuned using BO. (4) The test results show that the proposed hybrid deep learning framework achieved the best performance in the aspect of improvement in prediction performance and reduction in training time compared with a variety of benchmark models. Considering the LDVC is one of the best inputs for real-time control applications in urban areas, the proposed model could provide the accurate real-time LDVC forecast data for real-time traffic control.

The rest of the paper is organized as follows: Section 2 reviews the relevant literature about traffic prediction model, hybrid model, and hyperparameter optimization. Section 3 proposes the framework of the prediction model, including the logical relationship between data decomposition, model training, parameter optimization, and prediction component reconstruction. The fundamental methods are explained, including the sequence data decomposition method ICEEMDAN, the basic model GRU, and the hyperparameter tuning algorithm BO. Section 4 validates the proposed model using actual link dynamic vehicle count data collected in Hangzhou, China. We further study the influence of hyperparameter tuning and data decomposition on model performance. Besides, we compare the prediction accuracy and computing time of our model with 12 benchmark models. Section 5 concludes the paper.

2. Literature Review

In the past few decades, there has been a lot of research on traffic system forecasting. This paper focus on the traffic prediction model, hybrid model, and hyperparameter optimization. The literature review of the three research lines is summarized as follows.

This section first reviews the literature on traffic information prediction models, such as parametric models [12, 13, 22] and nonparametric models [20, 23–28].

A parametric model mainly considers some unsteady time series data to establish prediction models with limited parameters. Autoregressive integrated moving average (ARIMA) or seasonal autoregressive integrated moving average (SARIMA) is commonly used in time series data forecasting in transportation. For example, Williams et al. developed a SARIMA model to identify seasonal patterns to capture periodic changes in traffic states [13]. Van Der Voort et al. used a self-organizing neural network graph as the initial classifier associated with an individually ARIMA model to predict the half hour traffic flow on French highways [12]. Kumar et al. selected a three-lane arterial road with limited three days of traffic flow data in Chennai, India, to establish a SARIMA model for traffic flow forecasting [14].

The nonparametric models mainly include machine learning and deep learning approaches. Machine learning has been widely used to predict traffic information. For example, researchers used the k-nearest neighbor (KNN) model to predict short-term traffic flow [23, 26, 29]. Support vector regression (SVR) was utilized in traffic flow prediction [27, 29, 30] and travel time forecasting [20]. Random

forest regression (RF) was applied for traffic flow prediction [31]. Early neural network modeling such as multilayer perceptron (MLP) [24, 32], back propagation neural network (BPN) [33], and artificial neutral networks (ANN) were widely employed in the prediction of traffic systems. For example, Kumar et al. operated ANN for the short-term prediction of traffic volume [14]. Ruiz Aguilar et al. proposed a hybrid prediction method based on the combination of the ARIMA and ANN models to predict the number of goods inspected at European border checkpoints [34].

However, the traditional artificial neural network cannot capture time-series data features because it does not consider time dependence. To overcome this shortcoming, researchers have explored a large number of novel neural network models. The deep learning models are the fastest growing algorithms in recent years. In terms of sequence data modeling, RNN (Recurrent Neural Network) is one of the representatives. Van Lint et al. proposed a nonlinear state space method using RNN to predict short-term highway travel time [19].

RNN's variant LSTM (Long-Short Term Memory) solves the shortcoming that RNN cannot store long-term memory of information. LSTM has successful applications in traffic system prediction. For example, Ma et al. wielded remote microwave sensor data to establish LSTM models to predict traffic speed [35]. Zhao et al. proposed an LSTM prediction model for short-term traffic flow prediction [36]. Yang et al. applied LSTM to predict urban rail transit passenger flow [37].

As an improved algorithm of LSTM, GRU [38] was first proposed by Cho et al. in 2014. In most cases, the prediction performance of GRU is similar to LSTM, but the training time is reduced. Zhang et al. predict network-wide traffic speed with a deep learning model, and the results show that GRU obtains even better performance than LSTM [39].

The hybrid model that combines data decomposition and machine learning or deep learning approach can efficiently improve the prediction performance. The following part reviews the literature on hybrid models that combine the data decomposition methods, machine learning, and deep learning methods. Both machine learning and deep learning models require stable inputs, and data decomposition methods can effectively improve the quality of model input data and make the decomposed data more regular [40]. Choosing a reliable data sequence decomposition method is critical for the stable and effective input required in the forecasting model. Traditional data decomposition methods, such as wavelet transform (WT), have been successfully applied in transportation. For example, Wang and Shi established a short-term traffic speed prediction model based on chaotic wavelet transform and support vector machine [16]. However, the traditional wavelet transforms and Fourier transform techniques have disadvantages. For example, it is difficult to choose the mother wavelet, while empirical mode decomposition (EMD) and ensemble empirical mode decomposition (EEMD) are more effective. The decomposition method can decompose the data into intrinsic mode components. Many researchers have greatly improved the accuracy of the prediction model based on the decomposition of EMD and EEMD [41]. For example, Wei and Chen combined the EMD and back propagation neural network (BPN) model to predict short-term passenger flow in the subway system [33]. Yang and Chen combined EMD and stacked autoencoder (SAE) for passenger flow prediction in urban rail transit [42]. Jiang et al. combined the EEMD and grey support vector machine model to develop a hybrid short-term demand forecasting method for short-term high-speed rail passenger flow forecasting [43]. The particle swarm optimization algorithm was used to optimize the grey support vector machine, and the results show that the model performs well in terms of prediction accuracy. Zhang et al. proposed a hybrid deep learning prediction model that combined 3D convolutional neural network (3D CNN) and EEMD to predict the network-wide speed of Beijing, and the results showed that the EEMD method effectively improves the input data, and 3D CNN can consider the temporal and spatial characteristics of the road network [40]. As advantages in data decomposition, other improved algorithms were proposed based on EMD and EEMD, such as complementary EEMD (CEEMD) [44], a complete EEMD with adaptive noise (CEEMDAN) [45], and ICEEMDAN [46].

The excellent performance of the neural network prediction model is inseparable from the parameter optimization. The following literature reviews focus on the relevant parameter optimization methods. Suitable parameter setting shows an enormous impact on the performance of the neural network prediction model [43]. For parameter optimization in machine learning and deep learning models, manual tuning relies on experience and vulnerable to bias, and the tuning process is very time-consuming [47, 48]. Commonly used automatic parameter tuning algorithms [49] include grid search, random search, and Bayesian optimization. Grid and random search have shortcomings, in which the new search may separate from the previous search information and cannot make full use of prior knowledge. Bayesian optimization utilizes the prior distribution information of parameters [50]. It can auto effectively search for hyperparameters with fewer iteration steps. Bayesian optimization has become the most practical tool for parameter optimization in predictive systems, which is successfully applied in the deep learning model hyperparameter tuning recently, such as references [51–53].

In response to these challenges, we propose a hybrid deep learning model for link dynamic vehicle count forecasting. The data decomposition method ICEEMDAN is adopted to decompose the irregular traffic demand data to more simple IMFs components. The GRU model is used to predict the IMFs' components considering long- and short-term periodic characteristics of traffic demand, and Bayesian optimization is utilized for automatically tuning multiple hyperparameters of the deep learning models.

3. Link Dynamic Vehicle Count Forecasting Model

This paper proposes a hybrid deep learning model that combines ICEEMDAN and GRU with Bayesian optimization for link dynamic vehicle count forecasting, called

ICEEMDAN-GRU-BO. Figure 1 shows the framework and the main steps are as follows:

(1) Data processing, including data cleaning, normalization, and completion. For example, we fill in the missing data according to the average value of the previous three steps.

(2) Data decomposition. ICEEMDAN is adopted to decompose the link dynamic vehicle count into several intrinsic mode functions (IMFs) and a residual. These mode components are simpler and more regular, which can improve the accuracy of the deep learning model.

(3) Subcomponents prediction and hyperparameters optimization. GRU is used to predict subcomponents of different frequencies as the basic prediction model. This framework employs a Bayesian optimization algorithm to optimize the hyperparameters of each GRU model. These hyperparameters include the initial learning rate, number of hidden units, L2 regularization coefficient, and number of GRU layers.

(4) Mode reconstruction and results evaluation. The final prediction result can be obtained by summing up the predicted subcomponents and evaluated by the test dataset.

The following describes the model details.

3.1. Data Normalization. The data normalization is adopted in data processing to reduce data redundancy and improve data usability. After normalizing, the original data are converted into a pure dimensionless values. The training data are transformed into standardized data with zero mean and unit variance to better fit and prevent training divergence. The standardized formula is as follows:

$$X' = \frac{X - \overline{X}}{s(X)}, \tag{1}$$

where X' is the normalized data, X is the original data, $\overline{X}$ is the mean of the original data, and $s(X)$ is the standard deviation of the original data.

In the prediction stage, the mean and variance parameters are denormalized for the predicted data.

$$\widehat{X} = s(X)\widehat{X}' + \overline{X}, \tag{2}$$

where $\widehat{X}'$ is the predicted value after normalization and $\widehat{X}$ is the final predicted value.

3.2. Data Decomposition Method. The EMD [54] is an adaptive method for the analysis of nonstationary and nonlinear signals. EMD can decompose the original signal into the sum of amplitude and frequency modulation functions, called "Intrinsic Mode Function" (IMF), and the final monotonic trend. However, EMD has the problem of "mode mixing," which is very similar oscillations in different IMFs. Mode mixing reduces the EMD's ability to recognize different amplitudes in the actual data of the IMF components and affects the prediction accuracy of the hybrid model [43]. To overcome the problem of mode mixing, researchers have proposed a new data decomposition method that adds Gaussian white noise, called "Ensemble Empirical Mode Decomposition" (EEMD) [55–58]. EEMD is a noise-assisted data analysis method that aims to overcome the shortcomings of the EMD method. The steps of EEMD are as follows:

Step (1): Before EMD decomposition, Gaussian white noise is added to the original sequence data each time, and the construction sequence after addition is as follows:

$$x^i(t) = x(t) + w^i(t), \tag{3}$$

where $x^i(t)$ is the construction sequence data, $x(t)$ is the original sequence data, and $w^i(t) \sim N(0, \sigma^2)$ is the added white noise sequence data.

Step (2): EMD is adopted to decompose the construction sequence $x^i(t)$ into n IMFs.

$$x^i(t) = \sum_{j=1}^{n} c_j^i(t) + r_n^i(t), \tag{4}$$

where $c_j^i(t)$ is the ith decompose of jth IMF and $r_n^i(t)$ is the ith residual data.

Step (3): Repeat steps (1) and (2) M times, and add different white noise each time to obtain M groups of corresponding IMFs.

Step (4): Calculate the average value of the corresponding IMFs of the M groups as the final IMFs.

$$c_j(t) = \frac{1}{M}\sum_{i=1}^{M} c_j^i(t), \tag{5}$$

where $c_j^i(t)$ is ith decompose of jth IMF.

When the EEMD decomposition is completed, the original sequence data can be expressed as n IMFs and a residual.

$$x(t) = \sum_{j=1}^{n} c_j(t) + r_n(t), \tag{6}$$

where $c_j(t)$, $(t = 1, 2, \ldots, T)$ is the jth IMF component decomposed at time t, $r_n(t)$ is the final residual, and n is the number of IMFs.

The main problem of the EEMD is the high computing time and the residue of added noise present in the IMFs. In the EEMD, it can be recognized that every $x^i(t)$ is decomposed independently from the other realizations, and the reconstructed signal contains residual noise and different realizations of signal plus noise that may produce a different number of modes. To overcome this limitation, the CEEMDAN algorithm was first proposed by Torres et al. in 2011 [45]. The main idea of the CEEMDAN is to add white noise at each phase of decomposition and calculate a unique residue to obtain each mode.

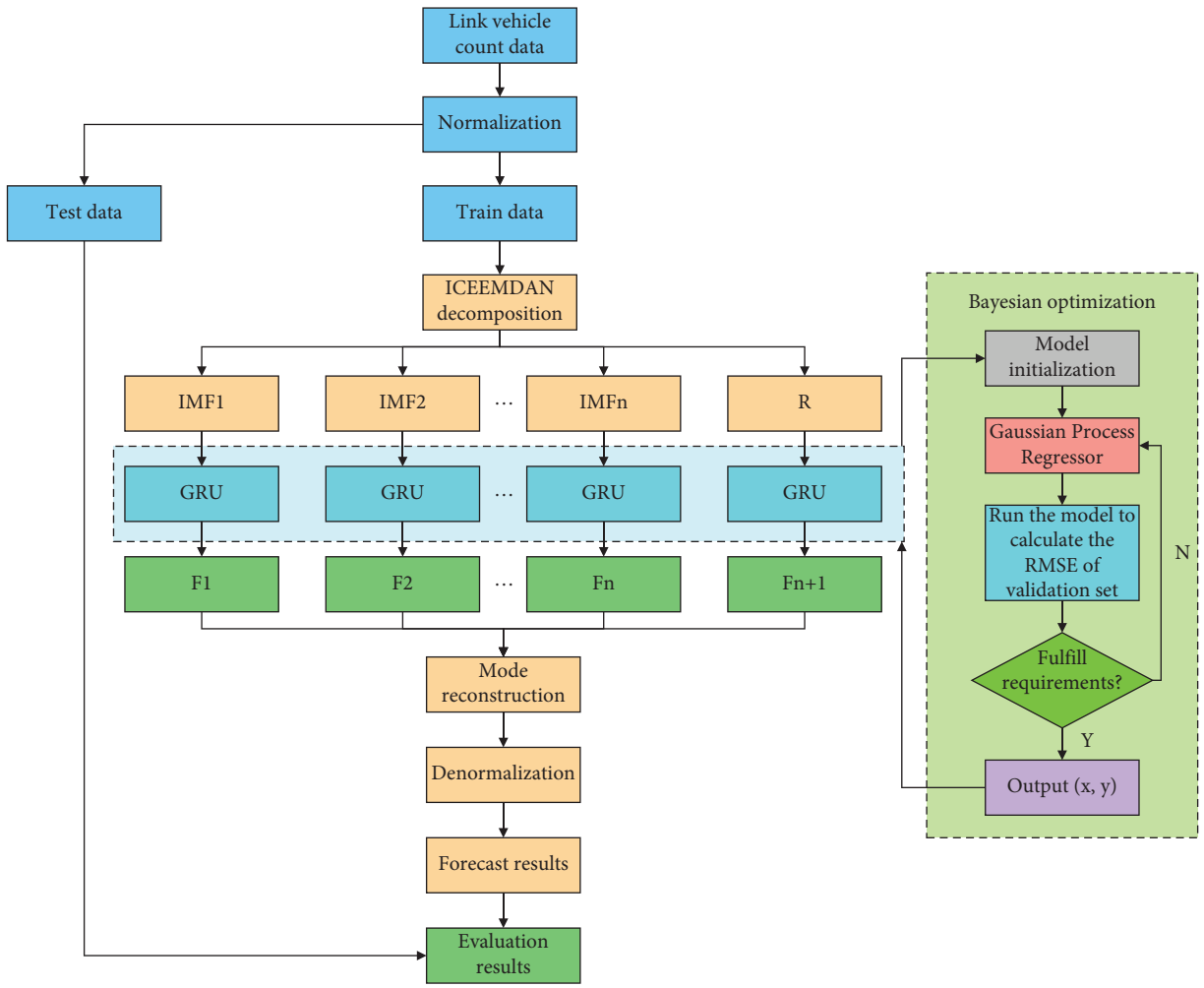

Figure 1: Framework of the link vehicle dynamic count forecasting model.

The resulting dissociation is complemented by a numerically negligible error. However, there are still problems with some residual noise and "spurious" modes. The ICEEMDAN technique is developed to improve the problems with some residual noise and "spurious" modes by Colominas et al. [46].

Given a composite signal $x(t)$, where t is the sampling sequence of the signal, and let $E_K(\cdot)$ be the kth IMF obtained by EMD, and define $M(\cdot)$ as the operator to calculate the local mean of the signal, then, the ICEEMDAN algorithm is described as follows:

Step 1: Calculate the local means of I realizations using the EMD algorithm:

$$x^i = x + \beta_0 E_1\left(w^i\right), \quad i = 1, \ldots, I, \tag{7}$$

where $\beta_0 = \varepsilon_0 \mathrm{std}(x)/E_1(w^i)$ and ε_0 is the reciprocal of the desired signal-to-noise ratio between the first added noise and the analyzed signal.

Step 2: Calculate the first residue R1:

$$R_1 = \langle M\left(x^i\right)\rangle, \tag{8}$$

where $\langle\cdot\rangle$ is the action of averaging throughout the realizations.

Step 3: Compute the first mode at the first stage ($k = 1$) as $d_1 = x - R_1$.

Step 4: Estimate the second residue as the average of local means of the realizations $R_1 + \beta_1 E_2(w^i)$ and define the second mode as follows:

$$d_2 = R_1 - R_2 = R_1 - \langle M\left(R_1 + \beta_1 E_2\left(w^i\right)\right)\rangle. \tag{9}$$

Step 5: For $k = 3, \ldots, K$; calculate the kth residue:

$$\begin{aligned} R_k &= \langle M\left(R_{k-1} + \beta_{k-1} E_k\left(w^i\right)\right)\rangle, \\ \beta_k &= \varepsilon_0 \mathrm{std}\left(r_k\right), k \geq 1. \end{aligned} \tag{10}$$

Step 6: Compute the kth mode:

$$d_k = R_{k-1} - R_k. \tag{11}$$

Step 7: Go back to step 4 for the next k.

Compared with EEMD and CEEMD, the ICEEMDAN can not only reduce the noise in the mode but also decrease

the residual spurious pattern problems caused by signal overlap, providing an accurate reconstruction of the original signal.

3.3. GRU Model. As a special RNN structure, LSTM solves the problems of vanishing gradient and explosive gradient by changing the cell structure and adding storage cells to determine whether it is necessary to remember information. GRU [38] improved LSTM by reducing the number of gates to decrease the training time. As shown in Figure 2, the GRU units transfer the input vector x_t to the output vector h_t through time t iteration. GRUs consist of two gates: the reset gate and the update gate. The main process in a GRU unit can be described as follows:

$$\begin{aligned} r_t &= \sigma\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right), \\ u_t &= \sigma\left(W_{xu}x_t + W_{hu}h_{t-1} + b_u\right), \\ y_t &= \tanh\left(W_{xh}x_t + W_{hy}\left(r_t \odot h_{t-1}\right) + b_y\right), \\ h_t &= \left(1 - u_t\right) \odot h_{t-1} + u_t \odot y_t, \end{aligned} \tag{12}$$

where u_t and r_t represent the update and reset gates of the GRU, respectively, y_t means the candidate activation, h_t represents the current activation, and h_{t-1} is the previous activation. W_{xr}, W_{hr}, W_{xu}, W_{hu}, W_{xh}, W_{hy} is the corresponding weight parameter matrices; b_r, b_u, b_y is the corresponding bias vector; and σ and tanh are the activation functions. For more GRU information, please see the work of Chung et al. [59].

3.4. Bayesian Optimization Hyperparameters. Hyperparameters are the parameters of the training algorithm itself, not directly learned from the training process. Each model has different hyperparameters, and a good choice of hyperparameters can get the best performance. For example, there are four key hyperparameters in the GRU model, such as the number of hidden units, the learning rate, the number of GRU layers, and the number of hidden units. However, manual tuning is inefficient and often affected by human subjective factors.

The basic idea of Bayesian optimization is to use Bayes' theorem to estimate the posterior distribution of the objective function based on the data and then select the hyperparameter combination of the following samples according to the distribution. The Bayesian optimization algorithm makes full use of the information of the previous sampling points. The algorithm optimizes by learning the shape of the objective function. It will find the hyperparameters that maximize the result to the global optimal.

Bayesian optimization is generally used to minimize the objective function $f(x)$ as follows:

$$x^* = \arg\min_{x \in \chi} f(x), \tag{13}$$

where x is a decision variable, χ is the decision space. In general, the objective function $f(x)$ is unknown, so it cannot use gradient descent to solve $f(x)$. Bayesian optimization utilizes a surrogate model to deal with the optimization problem of the unknown objective function. The model used to approximate the objective function is called the surrogate model. The surrogate model commonly used in Bayesian optimization is the Gaussian process to obtain its posterior distribution.

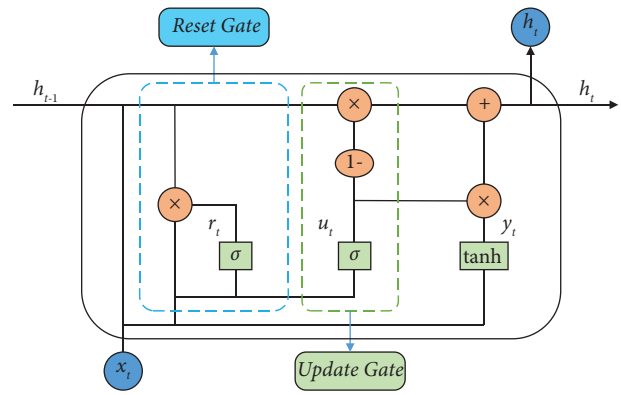

Figure 2: The structure of GRU.

Bayesian optimization also applies an acquisition function to direct sampling to an area that may improve the current best observation for searching for the next suitable sampling point. The carefully designed acquisition functions balance the exploration of the search space and existing fields [48]. The types of acquisition functions include PI (Probability of Improvement) and EI (Expected Improvement) [50, 60].

The steps of Bayesian optimization of hyperparameter tuning include:

(1) Model initialization. Prepare variables, such as the initial learning rate, the number of hidden units, and the L2 regularization coefficient.

(2) Use the Gaussian process to optimize the objective function.

(3) Perform Bayesian optimization and calculate the RMSE of the test set.

(4) Check the optimization results. If the results meet the requirements, Bayesian optimization will output the hyperparameters. Otherwise, the Bayesian optimization will be restarted or modified for the optimization options to continue.

4. Experiments and Results

4.1. Data Description. The license plate recognition (LPR) data collected in Jinji Road and Airport City Avenue in Hangzhou, China, are used to verify the proposed model. The selected section of Jinji Road is about 280 m long with three northbound lanes without entrance and exit in the middle. The selected section of Airport Avenue is about 560 m long. Figure 3 shows the location of the license plate recognition detector on the Jinji Road. The data were collected from December 1, 2018 to December 24, 2018. The original LPR record data contains the key information, such as license plate of the car, the timestamp of the car passing the detected line, lane number, location information, and

Figure 3: Detectors' location in Jinji road.

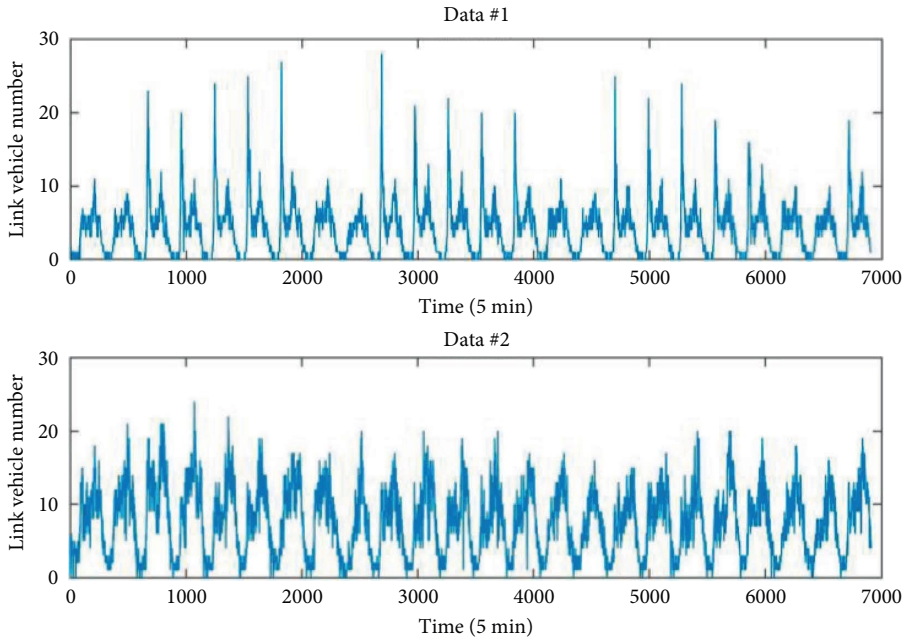

Figure 4: Link dynamic vehicle count data.

other information, such as car type, car color, and car length. Then, the raw LPR data are employed to extract the corresponding LDVC data with the 5 min time window by the cumulative curve model of upstream and downstream vehicles [10, 11, 61]. Thus, the model provides 288 LDVC data points each day. We divide the data into training set and test set with the ratio of 90 per cent training set and 10 per cent test set. The obtained LDVC data of Jinji Road (Data 1) and Airport Avenue (Data 2) are shown in Figure 4. The Data 1 on the Jinji Road reflects the periodic characteristics of the data in days and weeks. Data 1 has more obvious morning peak characteristics, while Data 2 has a slightly higher evening peak traffic demand. The proposed model is applied to the one-step ahead LDVC forecasting problem. The length of historical time window is set as 288. The prediction horizon is set as 1, that is to say, we use one day historical data to predict 5-min LDVC.

4.2. IMF Components Extraction. The quality of input data will affect the prediction performance, and ICEEMDAN improves the quality of input data by decomposing the original data into more regular pattern components. The calculation of the decomposition quantity m is determined by $m = \text{fix}(\log 2(N)) - 1$ [56], where N is the length of the input data. We used ICEEMDAN to decompose the original Data #1 into 11 IMF components and a residual, as illustrated in Figure 5. The periods of these components range from short to long periods and have different amplitudes. The first subgraph represents the original LDVC data with noise and the residual shows the trend term of the LDVC data. The ICEEMDAN algorithm overcomes the mode mixing problem of EMD. Taking IMF7 as an example, after decomposing by ICEEMDAN, the periodic characteristics of Data #1 in days and weeks are more obvious and more regular.

4.3. Benchmarks and Measures of Effectiveness. The real-world link dynamic vehicle data were divided into training data set and test data set. We selected 12 benchmark models, including the SVR (support vector regression) [25], RF

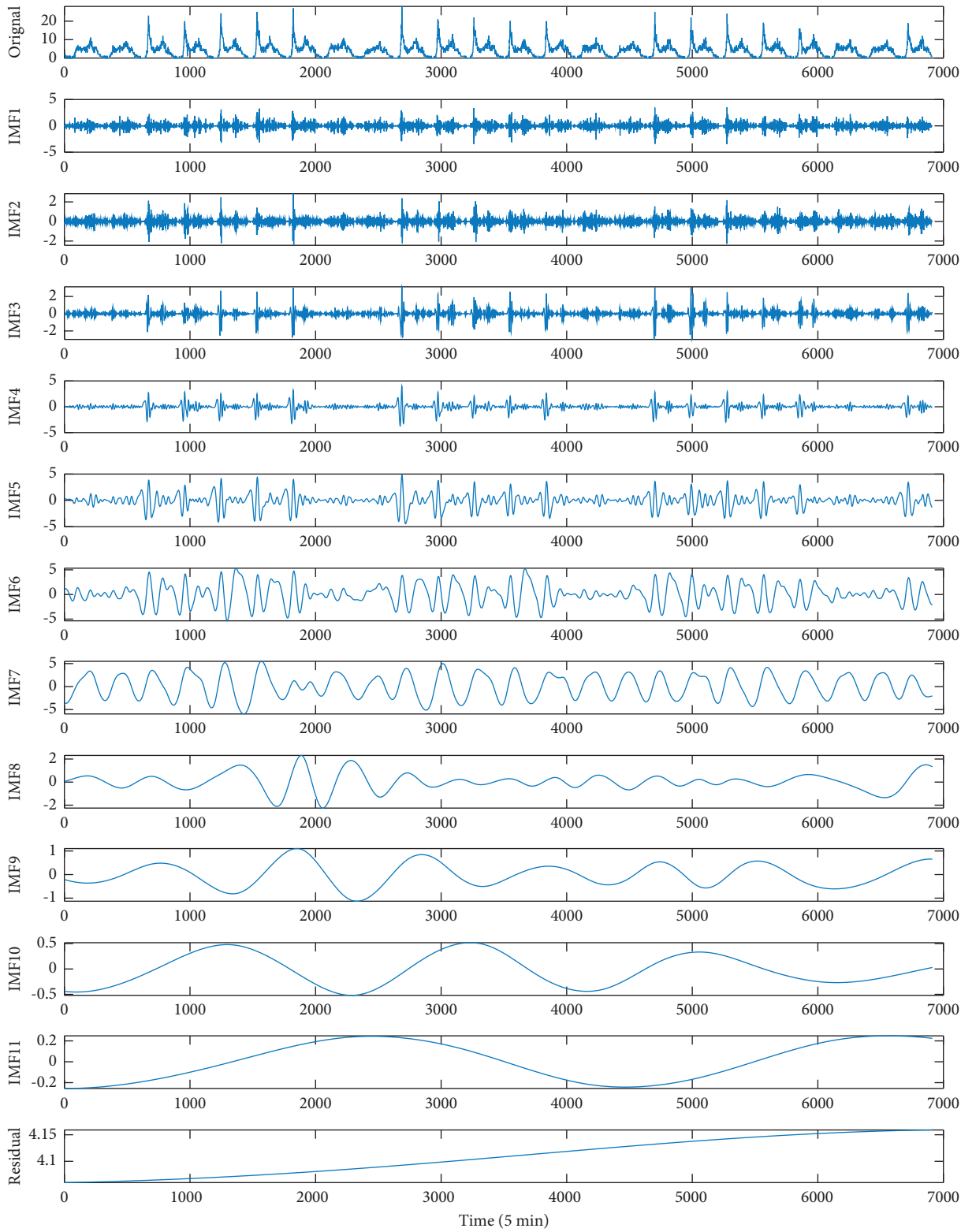

Figure 5: IMF components extraction by ICEEMDAN.

(random forest) [62], XGBoost (extreme gradient boosting tree regression) [63], GRU-BO, LSTM-BO, EMD-GRU, VMD-GRU (variational mode decomposition) [64], MVMD-GRU (multivariate variational mode decomposition) [65], EEMD-GRU, CEEMD-GRU (a complete ensemble empirical mode decomposition) [44], ICEEMDAN-GRU, and ICEEMDAN-LSTM-BO for comparison. ICEEMDAN-GRU-BO is the proposed prediction method, which considers the data decomposition and Bayesian optimization parameters. We choose the baseline

model for the following reasons: the first reason is to compare with commonly used machine learning and deep learning models. The second reason is to obtain the prediction performance using Bayesian optimization alone. The third reason compares the performance of different data decomposition methods.

This paper applied 3 evaluation indicators to evaluate the model, namely, mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2). The calculation formulas are as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|x^{(i)} - \widehat{x}^{(i)}\right|, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - \widehat{x}^{(i)}\right)^2}, \\ R^2 &= 1 - \frac{\sum_{i=1}^{n}\left(x^{(i)} - \widehat{x}^{(i)}\right)^2}{\sum_{i=1}^{n}\left(x^{(i)} - \overline{x}\right)^2}, \end{aligned} \tag{14}$$

where $x^{(i)}$ is the observed value of link dynamic vehicle count, $\widehat{x}^{(i)}$ is the predicted value of vehicle count, and $\overline{x}$ is the mean value of vehicle count.

A well-performing deep learning model is inseparable from an effective hyperparameter tuning process. In the following section, we first analyze the impact of model hyperparameters on the prediction performance. Then, we compare the prediction performance of the proposed hybrid depth model and the benchmark method.

4.4. Hyperparameter Tuning with Bayesian Optimization. This section verifies the superiority of the BO in tuning hyperparameters by the effect analysis of hyperparameters on the prediction performance. The hyperparameters of the GRU model mainly include the number of hidden units (HS), the initial learning rate (LR), the regularization coefficient (L2), and the number of GRU layers (NL) [51, 53]. When we compare the prediction performance of one hyperparameter, we fix the other hyperparameters.

The number of hidden units has a strong influence on the model. If the number of hidden units is too small, the net will not learn well, while too many hidden units will affect the efficiency and increase the risk of overfitting. The effects of hidden layer number on the GRU model are shown in Table 1. As the HS approaches the optimal value 80, the prediction accuracy of the model gradually increases. When the HS continues to increase, the RMSE and R^2 gradually decrease, reflecting the possibility of overfitting. The same rule also fits in the Data #2. Furthermore, we could conclude that the optimal number of hidden units for different data is different.

Then, we tested the effects of different initial learning rates on the GRU model. The initial learning rates have significant effects on the model: a too high learning rate may cause the model to fail to converge; a too small learning rate will cause the model to converge slowly or fail to learn. As shown in Table 2, when the initial learning rate is set to 0.1, the model cannot converge. In addition, the optimal initial LR for Data #1 is **0.007**, and for Data #2 is **0.003**, showing that it is hard to manually tune an appropriate initial LR for different data. Therefore, it is essential to determine a suitable initial learning rate automatically.

Table 1: Predictive performance comparison with different number of hidden units.

	HS	MAE	RMSE	R^2
		Data #1		
	40	0.598	0.968	0.897
LR = 0.005	**80**	**0.588**	**0.959**	**0.899**
NL = 1	120	0.601	0.975	0.895
L2 = 0.001	160	0.601	0.969	0.897
	200	0.598	0.969	0.897
		Data #2		
	40	1.189	1.585	0.862
LR = 0.005	80	1.171	1.582	0.863
NL = 1	**120**	**1.171**	**1.559**	**0.873**
L2 = 0.001	160	1.183	1.587	0.860
	200	1.185	1.592	0.858

The bold values means the best results in the data set.

Table 2: Predictive performance comparison with different initial learning rates.

	LR	MAE	RMSE	R^2
		Data #1		
	0.001	0.609	0.972	0.896
	0.003	0.601	0.969	0.897
NL = 1	0.005	0.599	0.977	0.895
HS = 80	**0.007**	**0.586**	**0.957**	**0.899**
L2 = 0.001	0.010	0.583	0.959	0.899
	0.015	0.634	1.011	0.887
	0.1	NaN	NaN	NaN
		Data #2		
	0.001	1.158	1.524	0.871
	0.003	**1.156**	**1.516**	**0.875**
NL = 1	0.005	1.203	1.565	0.852
HS = 80	0.007	1.151	1.525	0.871
L2 = 0.001	0.010	1.169	1.535	0.866
	0.015	1.254	1.676	0.798
	0.1	NaN	NaN	NaN

The bold values means the best results in the data set.

The Bayesian optimization model based on the Gaussian process can effectively search the candidate hyperparameter interval and determine the appropriate HS and initial LR, as shown in Figure 6(a). Moreover, there exist interactions between different hyperparameters. The superiority of Bayesian optimization is optimizing multiple hyperparameters at the same time, as illustrated in Figure 6(b). The Bayesian optimization can simultaneously optimize the initial LR and HS and select the appropriate combination.

In terms of the L2 regularization coefficient, it helps to improve the overfitting issues and the generalization level of the model. However, a too large regularization coefficient may lead to the underfitting of the model. We manually adjust the L2 regularization coefficient to examine the prediction performance of the model, as shown in Table 3.

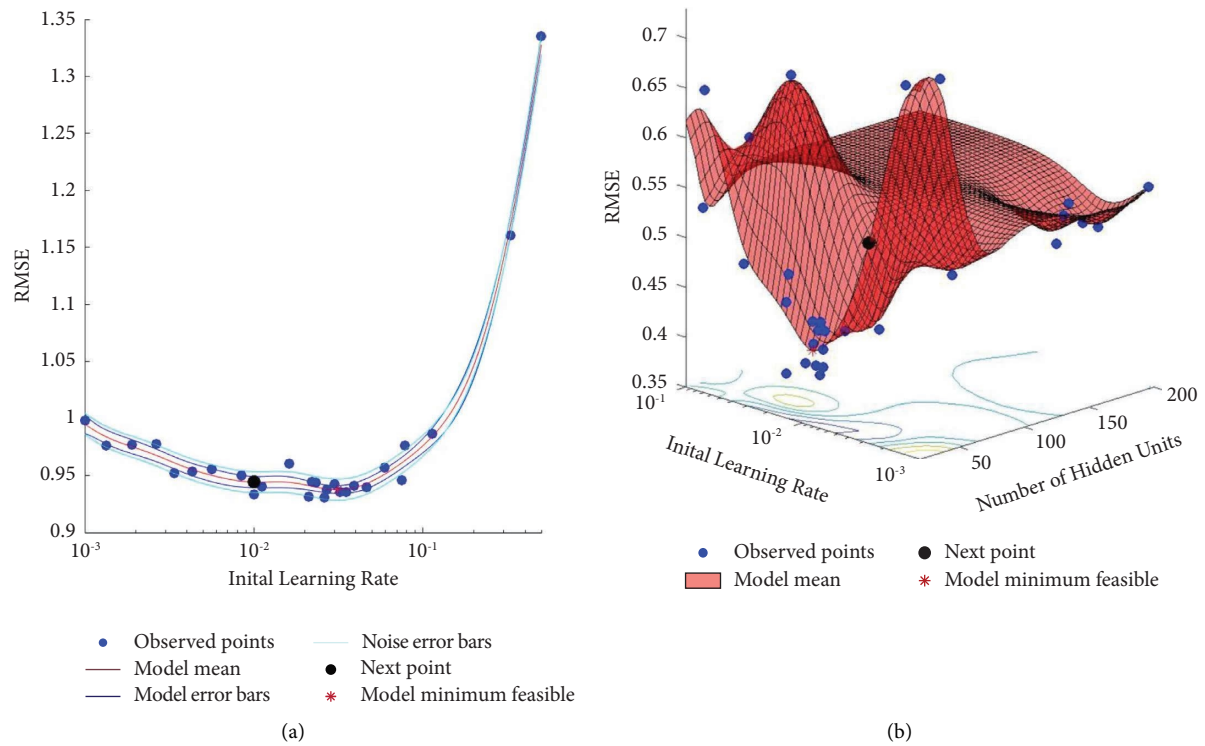

Figure 6: Adjusting model hyperparameters with Bayesian optimization. (a) Bayesian optimization adjusts the initial learning rate. (b) Bayesian optimization adjusts the initial learning rate and hidden unit size.

The results in Table 3 show that different regularization coefficients impact the model prediction accuracy. Data #1 corresponds to the optimal regularization coefficient equal to **0.007** and Data #2 corresponds to the optimal L2 regularization coefficient equal to **0.005**, showing that the model of different data suits its own optimal L2 regularization coefficients.

In Table 4, we look into the impact of multilayer GRU on the prediction performance of the model. In a multilayer GRU, the number of hidden units in each layer is equally distributed. Second, a dropout layer with a dropout probability equal to 0.2 is added after each GRU layer to avoid overfitting.

The results also show RMSE increases as the number of GRU layers increases. The optimal number of GRU layers corresponding to Data #1 and Data #2 is one layer. It shows that as the number of layers increases, the model may overfit.

In Table 5, we use the Bayesian optimization algorithm to optimize the four hyperparameters of the model. Compared with the manually tuning method, Bayesian optimization could obtain the best-fitted hyperparameters when all evaluation indicator results are optimal.

This section shows the hyperparameters have a significant impact on the performance of the GRU model. The setting of various hyperparameters also presents the phenomenon of trade-offs. Manual tuning is not only time-consuming and labor-intensive but also arduous to obtain better results. Bayesian optimization could achieve the best performance in tuning hyperparameters.

Table 3: Influence of different L2 regularization coefficients on the model.

	L2	MAE	RMSE	R^2
		Data #1		
LR = 0.005 NL = 1 HS = 80	0.001	0.614	0.989	0.892
	0.003	0.593	0.962	0.898
	0.005	0.599	0.980	0.894
	0.007	**0.582**	**0.955**	**0.900**
		Data #2		
LR = 0.00 NL = 1 HS = 80	0.001	1.180	1.599	0.840
	0.003	1.174	1.585	0.846
	0.005	**1.173**	**1.580**	**0.848**
	0.007	1.183	1.606	0.837

Table 4: Influence of different GRU layers on the model.

	NL	MAE	RMSE	R^2
		Data #1		
LR = 0.005 HS = 80/NL L2 = 0.001	**1**	**0.653**	**0.931**	**0.905**
	2	0.657	0.941	0.902
	3	0.675	0.946	0.901
	4	0.677	0.950	0.901
		Data #2		
LR = 0.005 HS = 80/NL L2 = 0.001	**1**	**1.122**	**1.603**	**0.833**
	2	1.129	1.600	0.830
	3	1.125	1.601	0.831
	4	1.133	1.601	0.831

The bold values means the best results in the data set.

Table 5: Effects of Bayesian optimization super parameters on the model.

	MAE	RMSE	R^2
	Data #1		
LR = 6.281e − 04; S = 173 L2 = 1.548e − 05; NL = 1	0.580	0.835	0.923
	Data #2		
LR = 0.0054; HS = 184 L2 = 0.046; NL = 1	1.089	1.297	0.882

Table 6: Performance comparison of different prediction models.

	Data #1					*Data #2*				
	MAE	RMSE	R^2	Training time (min)	Prediction time (s)	MAE	RMSE	R^2	Training time (min)	Prediction time (s)
SVR	0.724	1.048	0.879	3.4	0.02	1.182	1.683	0.842	2.7	0.02
RF	0.683	1.028	0.884	0.5	0.06	1.158	1.661	0.846	0.2	0.02
XGBoost	0.684	1.030	0.883	0.2	0.02	1.158	1.660	0.846	0.1	0.01
GRU-BO	0.586	0.835	0.923	12.3	3.94	1.089	1.297	0.882	11.3	3.47
LSTM-BO	0.528	0.749	0.938	19.2	4.67	1.022	1.390	0.892	17.1	4.32
EMD-GRU	0.537	0.705	0.945	11.4	5.17	0.524	0.712	0.906	9.0	5.80
VMD-GRU	0.383	0.536	0.968	4.8	3.26	0.407	0.550	0.944	4.8	2.85
MVMD-GRU	0.601	0.833	0.924	4.0	2.74	0.531	0.721	0.904	4.4	3.19
EEMD-GRU	0.335	0.457	0.977	9.6	5.48	0.374	0.505	0.953	8.9	5.69
CEEMD-GRU	0.322	0.453	0.977	9.7	5.68	0.363	0.510	0.952	9.1	5.61
ICEEMDAN-GRU	0.257	0.352	0.986	10.4	6.74	0.289	0.420	0.967	10.8	5.86
ICEEMDAN-LSTM-BO	0.217	0.307	0.990	221.4	7.47	0.223	0.311	0.982	201.2	6.14
ICEEMDAN-GRU-BO	0.203	0.298	0.990	146.0	6.13	0.233	0.324	0.981	136.6	6.43

4.5. Benchmarks and Model Comparisons. This part studies the effect of data decomposition and Bayesian optimization on model performance. We compared the performance of 12 benchmark models with the proposed ICEEMDAN-GRU-BO model. We used a computer with a 2.9 Ghz, dual-core processor and 8G memory in our experiments.

The parameters setting of the models are set as follows. The parameters of SVR, RF, and XGBoost are optimized by the grid search method. According to the results of grid search, the kernel coefficient, regularization parameter, and epsilon are three essential parameters in the SVR model, which are set as 288.83, 29.31, and 0.0017 in Data #1 and 60.2, 775.97, and 0.0061 in Data #2, respectively. The number of trees and max depth are two essential parameters in the RF model, which are set as 138 and 15 in Data #1 and 22, 9 in Data #2, respectively. The number of estimators, learning rate, and max depth are three key parameters in XGBoost, which are set as 117 and 0.3, 6 in Data #1 and 102 and 0.4, 5 in Data #2, respectively. The hyperparameters of the GRU model such as the number of hidden units, the initial learning rate, the regularization coefficient, and the number of GRU layers are tuning with Bayesian optimization. According to the results of section 4.4, we set the initial learning rate to 0.005, the number of hidden units to 80, the L2 regularization coefficient to 0.001, and the number of GRU layers to one.

Table 6 shows the performance comparison of different models and Figure 7 shows the prediction error box plot. The results of Data #1 show that the performance of the ICEEMDAN-GRU-BO model is the best. The evaluation indicators of the proposed model are similar to the ICEEMDAN-LSTM-BO model in Data #2. However, the training time of the proposed model is drastically reduced.

XGBoost and RF show similar predictive performance, and both perform better than SVR. By comparing with XGBoost and RF models, the GRU-BO and the LSTM-BO models improves prediction accuracy. By comparing with GRU-BO, EMD-GRU achieves better prediction accuracy due to the effect of EMD, which shows that data decomposition obtains a better outcome.

The effect of different data decomposition on model performance shows that, compared with EMD, VMD, MVDM, EEMD, and CEEMD, the ICEEMDAN data decomposition has the most significant improvement in model prediction.

Although Bayesian optimization increases the training time of ICEEMDAN-GRU-BO, the training time is within an acceptable range for the improvement of prediction performance.

Bayesian optimization effectively and simultaneously optimizes the four hyperparameters of the model; avoiding manual tuning that only relies on the empirical

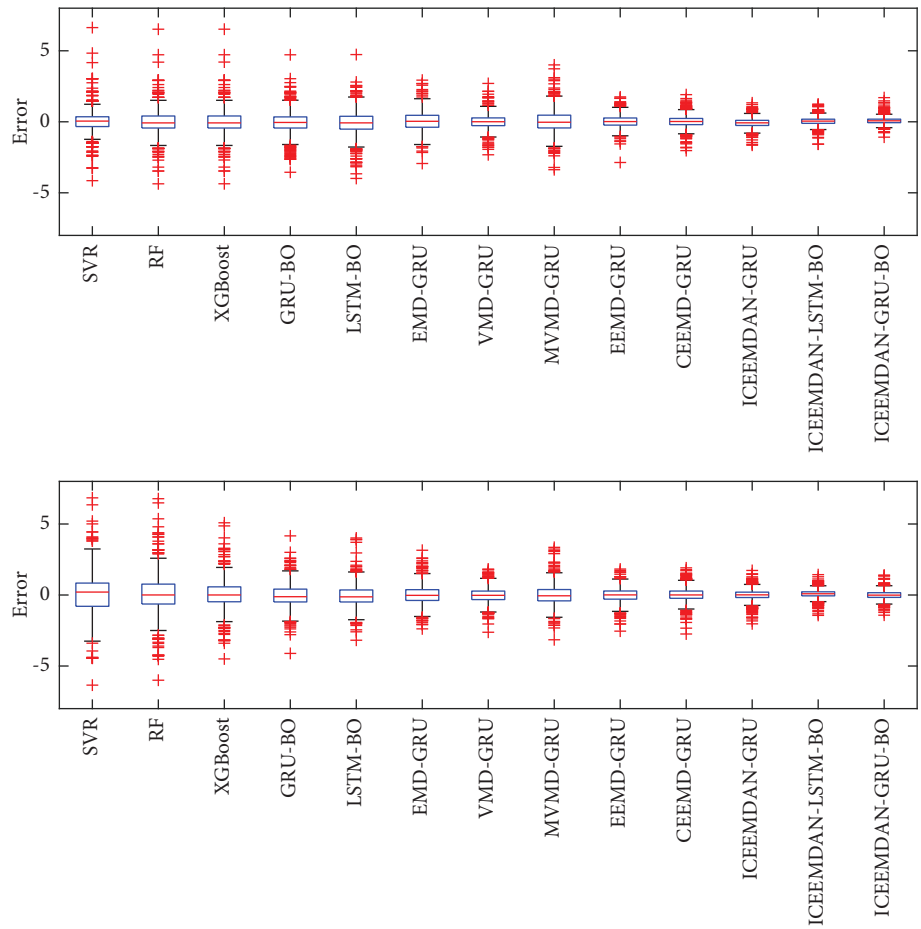

Figure 7: Box plots of prediction errors of different models.

methods. In addition, with the development of computing technology, such as the technology represented by cloud computing, higher-performance computing resources will become cheaper and more convenient, and the training time of the model will not be a limiting factor.

5. Conclusions and Discussion

In this paper, we propose a hybrid deep learning model that combines the data decomposition method, ICEEMDAN, and GRU model with Bayesian optimization for link dynamic vehicle count forecasting. ICEEMDAN is used to process and decompose the original data into specific IMFs. Considering the nonlinear characteristics of IMFs of link dynamic vehicle data, GRU is used as the basic forecast model of each IMF. BO is adapted to auto-tune hyperparameters of the deep learning approach. The presented model is compared with 12 benchmark models including SVR, RF, XGBoost, GRU-BO, LSTM-BO, EMD-GRU, VMD-GRU, MVMD-GRU, EEMD-GRU, CEEMD-GRU, ICEEMDAN-GRU, and ICEEMD-LSTM-BO. Models are validated with the real-world data collected in Hangzhou, China. Three evaluation indicators (MAE, RMSE, and R^2), training time, and prediction time are used to measure the performance. Results indicated that ICEEMDAN, GRU, and BO all contribute to the improvement of prediction accuracy and efficiency. Combining ICEEMDAN-GRU-BO can obtain the best performance and least calculation complexity.

The prediction model is based on the ICEEMDAN-GRU-BO for link dynamic vehicle count. It is necessary to establish different prediction models for different road section scenarios. In the future, we will further explore whether we can make full use of the training information of the previous models and train the deep learning prediction model by transfer learning to improve the modeling efficiency and prediction accuracy. Besides, LDVC is a crucial parameter in developing efficient adaptive traffic signal controllers as traffic-responsive control systems require reliable real-time demand information on prevailing traffic conditions to make sensible control decisions. The proposed model can be used to predict the basic inputs for real-time control applications in urban areas and further extended to network-wide vehicle counts forecasting for network-wide traffic signal control optimization in urban traffic management.

Data Availability

The link vehicle count data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research study was financially supported by National Natural Science Foundation of China under Grant nos. 52131202, 71901193, and 52072340; National Key Research and Development Program of China under Grant no. 2019YFB1600303; and China Postdoctoral Science Foundation under Grant no. 2020M671724.

References

[1] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.

[2] K. M. A. E. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos, "A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 680–694, 2010.

[3] S. Chen and D. J. Sun, "An improved adaptive signal control method for isolated signalized intersection based on dynamic programming," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 4–14, 2016.

[4] G. Vigos, M. Papageorgiou, and Y. Wang, "Real-time estimation of vehicle-count within signalized links," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 18–35, 2008.

[5] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "Real-time measurement of link vehicle count and travel time in a road network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 814–825, 2010.

[6] G. Vigos and M. Papageorgiou, "A simplified estimation scheme for the number of vehicles in signalized links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 312–321, 2010.

[7] M. Papageorgiou and G. Vigos, "Relating time-occupancy measurements to space-occupancy and link vehicle-count," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 1–17, 2008.

[8] M. Rostami Shahrbabaki, A. A. Safavi, M. Papageorgiou, and I. Papamichail, "A data fusion approach for real-time traffic state estimation in urban signalized links," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 525–548, 2018.

[9] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.

[10] X. Zhan, R. Li, and S. V. Ukkusuri, "Lane-based real-time queue length estimation using license plate recognition data," *Transportation Research Part C: Emerging Technologies*, vol. 57, pp. 85–102, 2015.

[11] C. He, D. Wang, M. Chen, G. Qian, and Z. Cai, "Link dynamic vehicle count estimation based on travel time distribution using license plate recognition data," *Transportmetrica: Transportation Science*, pp. 1–22, 2021.

[12] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[13] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record*, vol. 1644, no. 1, pp. 132–141, 1998.

[14] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction in heterogeneous condition using artificial neural network," *Transport*, vol. 30, no. 4, pp. 397–405, 2013.

[15] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6910–6920, 2021.

[16] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on Chaos–Wavelet Analysis-Support Vector Machine theory," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 219–232, 2013.

[17] L. Zhou, S. Zhang, J. Yu, and X. Chen, "Spatial–temporal deep tensor neural networks for large-scale urban network speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3718–3729, 2020.

[18] Y. Chen, C. Tao, Q. Bai, F. Liu, X. Qi, and R. Zhuo, "Short-term speed prediction for expressway considering adaptive selection of spatiotemporal dimensions and similar traffic features," *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 10, Article ID 04020114, 2020.

[19] J. W. C. Van Lint, S. P. Hooqendoorn, and H. J. Van Zuvlen, "Freeway travel time prediction with state-space neural networks modeling state-space dynamics with recurrent neural networks," *Transportation Research Record*, vol. 1811, pp. 30–39, 2002.

[20] C.-H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[21] J. van Lint, S. Hoogendoorn, and H. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5-6, pp. 347–369, 2005.

[22] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.

[23] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[24] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches," in *Proceedings of the IEEE international conference on systems, man and cybernetics*, San Antonio, TX, USA, October 1994.

[25] H. Drucker, C. J. C. Burges, L. Kaufman, S. Alex, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.

[26] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[27] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019.

[28] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.

[29] Y. S. Jeong, Y. J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.

[30] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Applied Mathematics and Computation*, vol. 217, no. 15, pp. 6733–6747, 2011.

[31] G. Leshem and Y. Ritov, "Traffic flow prediction using adaboost algorithm with random forests as a weak learner," *International Journal of Mathematical and Computational Sciences*, vol. 1, no. 1, pp. 1–6, 2007.

[32] H. Chen, S. Grant-Muller, L. Mussone, and F. Montgomery, "A study of hybrid neural network approaches and the effects of missing data on traffic forecasting," *Neural Computing & Applications*, vol. 10, no. 3, pp. 277–286, 2001.

[33] Yu Wei and Mu-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.

[34] J. J. Ruiz-Aguilar, I. J. Turias, and M. J. Jiménez-Come, "Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting," *Transportation Research Part E: Logistics and Transportation Review*, vol. 67, pp. 1–13, 2014.

[35] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[36] Z. Zhao, W. H. Chen, X. M. Wu, P. C. Y. Chen, and J. M. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[37] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features," *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1475–1482, 2019.

[38] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[39] K. Zhang, L. Zheng, Z. Liu, and N. Jia, "A deep learning based multitask model for network-wide traffic speed prediction," *Neurocomputing*, vol. 396, pp. 438–450, 2020.

[40] S. Zhang, L. Zhou, X. M. Chen, L. Zhang, L. Li, and M. Li, "Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 10, pp. 1132–1147, 2020.

[41] L. Li, X. Qu, J. Zhang, H. Li, and B. Ran, "Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network," *Applied Soft Computing*, vol. 73, pp. 921–932, 2018.

[42] H. F. Yang and Y. P. P. Chen, "Hybrid deep learning and empirical mode decomposition model for time series applications," *Expert Systems with Applications*, vol. 120, pp. 128–138, 2019.

[43] X. Jiang, L. Zhang, and X. Michael Chen, "Short-term forecasting of high-speed rail demand: a hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China," *Transportation Research Part C: Emerging Technologies*, vol. 44, pp. 110–127, 2014.

[44] J.-R. Yeh, J. S. Shieh, and N. E. Huang, "Complementary ensemble empirical mode decomposition: a novel noise enhanced data analysis method," *Advances in Adaptive Data Analysis*, vol. 02, no. 02, pp. 135–156, 2010.

[45] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proceedings of the 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Prague, Czech Republic, May 2011.

[46] M. A. Colominas, G. Schlotthauer, and M. E. Torres, "Improved complete ensemble EMD: a suitable tool for biomedical signal processing," *Biomedical Signal Processing and Control*, vol. 14, pp. 19–29, 2014.

[47] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI-Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, 2015.

[48] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: a review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[49] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 25th annual conference on neural information processing systems (NIPS 2011)*, Granada, Spain, January 2011.

[50] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[51] H. Cheng, X. Ding, W. Zhou, and R. Ding, "A hybrid electricity price forecasting model with Bayesian optimization for German energy exchange," *International Journal of Electrical Power & Energy Systems*, vol. 110, pp. 653–666, 2019.

[52] F. He, J. Zhou, Z.-K. Feng, G. Liu, and Y. Yang, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," *Applied Energy*, vol. 237, pp. 103–116, 2019.

[53] H. Yi and K. H. N. Bui, "An automated hyperparameter search-based deep learning model for highway traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5486–5495, 2021.

[54] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[55] N. E. Huang, M. L. C. Wu, S. R. Long et al., "A confidence limit for the empirical mode decomposition and hilbert spectral analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 459, pp. 2317–2345, 2003.

[56] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 1, no. 1, pp. 1–41, 2009.

[57] Y. X. Wu, Q.-B. Wu, and J.-Q. Zhu, "Improved EEMD-based crude oil price forecasting using LSTM networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 516, pp. 114–124, 2019.

[58] Y. Shrivastava and B. Singh, "A comparative study of EMD and EEMD approaches for identifying chatter frequency in CNC turning," *European Journal of Mechanics - A: Solids*, vol. 73, pp. 381–393, 2019.

[59] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[60] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: the Bayesian optimization algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, Orlando, Florida, USA, July 1999.

[61] A. Bhaskar, T. Tsubota, L. M. Kieu, and E. Chung, "Urban traffic state estimation: fusing point and zone based data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 120–142, 2014.

[62] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[63] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, San Francisco, CA, USA, August 2016.

[64] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[65] N. U. Rehman and H. Aftab, "Multivariate variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 6039–6052, 2019.