

Research Article

A Hybrid Approach by CEEMDAN-Improved PSO-LSTM Model for Network Traffic Prediction

Bilin Shao (),¹ Dan Song (),¹ Genqing Bian (),² and Yu Zhao ()¹

¹School of Management, Xi'an University of Architecture and Technology, Xi'an, China ²School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, China

Correspondence should be addressed to Dan Song; 674355101@qq.com

Received 11 December 2021; Revised 23 June 2022; Accepted 15 July 2022; Published 12 September 2022

Academic Editor: Jinwei Wang

Copyright © 2022 Bilin Shao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an important part of data management, network traffic evaluation and prediction can not only find network anomalies but also judge the future trends of the network. To predict network traffic more accurately, a novel hybrid model, integrating Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) with long short-term memory neural network (LSTM) optimized by the improved particle swarm optimization (IPSO) algorithm, is established for network traffic prediction. Firstly, an LSTM prediction model for the real-time mutation and dependence of network traffic is constructed, and the IPSO is applied to optimize the hyperparameters. Then, CEEMDAN is introduced to decompose sequences of raw network traffic data into several different modal components containing different information to reduce the complexity of the network traffic sequence. Finally, the evaluation of the experiments shows the feasibility and effectiveness of the proposed method by comparing it with other deep neural architectures and regression models. The results show that the proposed model CEEMDAN-IPSO-LSTM produced a significantly superior performance with a reduction of the prediction error.

1. Introduction

As the information storage terminals of the Internet of Things (IoT), and with the continuous expansion of the data center scale, the network structure of the data center is becoming increasingly complex, the network business and network data flow are growing rapidly, and the frequency of network congestion is also getting higher and higher [1-3]. Network traffic monitoring, network resource optimization, network congestion avoidance, and network security strategy are of great significance in the real-time analysis of network traffic [4-8]. When the network is overloaded or congested, accurate prediction can ensure high-quality execution of network services with super importance or priority [9]. In recent years, predictive analysis based on historical network traffic has become a major research topic in the academic field. Establishing an accurate prediction model to describe network traffic characteristics contributes to optimizing network topology structure and route planning, reducing energy consumption, and providing more reliable service quality assurance.

Network links change dynamically with limited node processing resources. Network traffic prediction mainly depends on the statistical characteristics of flow and the strong correlation between time-sequence values. Modeling analysis based on network traffic time series is an effective method for network traffic research, which has been widely used in network traffic prediction and network performance evaluation. How to fully consider the complex characteristics of network traffic, and improve the prediction accuracy and real-time of network traffic has always been a hot and difficult topic of network traffic research [10].

In this study, we propose a novel hybrid model based on the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), the Long Short-Term Memory (LSTM) neural network model, and the improved particle swarm optimization (IPSO) algorithm to predict and analyze the network traffic. Firstly, combined with the real-time mutability, dependence, and highly nonlinear characteristics of network traffic, we establish the LSTM network traffic prediction model to extract the dynamic characteristics of network traffic. Then, the IPSO is utilized for hypermeters optimization. In addition, the CEEMDAN method is employed to decompose the network traffic data into several simplified modes. Finally, we compare the prediction accuracy of different models to evaluate the prediction effects of the CEEMDAN-IPSO-LSTM neural network model.

The main contributions of our work are presented as follows:

- (1) CEEMDAN is introduced to decompose network traffic data into several components, which creates modal confusion and avoids making larger impacts on the original signal during adding the white noise.
- (2) Network traffic prediction model based on LSTM is constructed to pursue the real-time mutation of network traffic.
- (3) The improved PSO algorithm is proposed to optimize the hyperparameters of the LSTM network traffic model. The optimization of hyperparameters of the LSTM prediction model can improve the prediction performance.

The rest of this paper is as follows. In section 2, we review the related research about network traffic prediction. In section 3, we constructed a network traffic prediction model based on LSTM. In section 4, we study the hyperparameter optimization of LSTM by the Improved Particle Swarm Optimization, and network traffic data denoising by CEEMDAN. Section 5 presents our evaluation of the proposed method. Finally, the main conclusions and future work are drawn in section 6.

2. Related Work

Due to the importance of network traffic prediction, there has been much research on network traffic prediction methods in recent years. Generally, network traffic prediction can be divided into short-term prediction, medium-term prediction, and long-term prediction according to the different cycles of network traffic prediction, [11] while network traffic prediction models are mainly divided into two categories: parametric model and nonparametric model [12].

2.1. Parameter model. The parametric model has the advantage of being simple and easy to understand. Moreover, it does not have high standards for training data, and the solving process is easy compared to nonparametric model, which consumes less time. However, the parametric model is suitable for the prediction of small data volumes with obvious features and stable structure, while the network traffic has real-time mutability and dependency characteristics, and the parametric model will lead to higher prediction errors than the nonparametric model.

ARMA network traffic model can effectively analyze network data with a stable flow in a short period, obtain network traffic characteristics at the corresponding scale, and realize data flow decomposition [13]. In the network traffic model based on ARMA, the deployment of the multiscale fitting process can obtain high accuracy under any expiration delay, simplify the ARMA model, and enhance the integration effect of the ARMA framework in network traffic modeling [14].

However, ARMA is not suitable for long-term network traffic data with network anomalies because the premise of ARMA modeling is that the data analyzed is a stationary random process. Most of the actual network traffic data are nonstationary [15], which can be transformed into stationary data after finite-difference. Therefore, some scholars proposed the Autoregressive Integrated Moving Average (ARIMA) model in the research process [16].

2.2. Nonparametric model. Nonparametric model refers to a model with no fixed structure and fixed parameters. Common nonparametric models include Support Vector Machine (SVM), k-Nearest Neighbor (kNN) [8], Artificial Neural Network (ANN), etc. The nonparametric model can automatically fit a variety of function forms without assumption, and the training effect is good, which is suitable for predicting large data volume.

Due to the real-time variability and dependence of network traffic, traditional network traffic prediction models have some disadvantages such as weak generalization ability and limited prediction accuracy. Therefore, more and more researchers use nonparametric models to predict network traffic data. The Support Vector Regression model (SVR) and its variant MK-SVR are first used to predict network traffic [17–19], which effectively predicts the changing trend of network traffic data but lacks the consideration of temporal correlation of time series data leading to a limit of prediction accuracy.

Methods based on the artificial neural network, such as Convolutional Neural Network (CNN) [20], improve the effect of flow classification by autonomous feature learning of data [21]. LSTM neural network and Gated Recurrent Unit (GRU) neural network have a superior effect over existing SVM and ANN models in predicting network traffic, which is more suitable for random nonlinear network traffic prediction [12]. LSTM neural network was originally used for short-term flow prediction, which can better learn the abstract representation of nonlinear flow data and capture the inherent characteristics of long-term dependence relationship in continuous data, thus improving the accuracy of flow prediction [22]. LSTM neural network is used for network traffic prediction, and the auto-correlation coefficient is added to the model to describe the trend of network traffic change better, which improves the accuracy of the prediction model [23]. On this basis, the improved Particle Filter (PF) algorithm is used to optimize the LSTM model, which improves the training rate and overcomes the shortcoming of convergence to local optimal in the traditional LSTM network [24].

The experiments of many neural network methods to predict the network traffic data show that in a real-time network data set, LSTM is of better performance than Recurrent Neural Network (RNN), the Feed-forward Neural Network (FFN), and other classic methods. LSTM neural network can more accurately simulate time series and its long-term dependencies than the traditional RNN, in large network traffic matrix prediction, and obtain a faster convergence rate [25]. The variants of LSTM neural network, GRU neural network, and identity-RNN (IRNN) have comparable performance with LSTM [26]. Minimal Gated Unit (MGU) overcomes the shortcoming of the high computing cost of the LSTM network and achieves relatively predictable performance with less model training time [27]. In addition, LSTM neural network has achieved good prediction results in financial data forecast [28, 29], metal price prediction [30], air quality index prediction [31], modular temperature prediction [32], and bridge health monitoring [33].

In summary, a single parametric or nonparametric model has its problems and defects, while a hybrid prediction model can overcome the shortcomings of a single model by combining two or more models. The hybrid model mainly combines some decomposition algorithms, optimization algorithms, and prediction algorithms, respectively, in the data preprocessing, prediction, and result correction stage of network traffic prediction. Although combinatorial prediction has achieved good results in other researches [34, 35], there are still some problems, such as how to choose the prediction model and its parameters, how to integrate the prediction results reasonably, and how to choose the appropriate decomposition algorithm or optimization algorithm. For network traffic prediction, using the combined prediction model and overcoming the above problems is a research direction worthy of further study.

3. Network Traffic Prediction Based on LSTM

3.1. LSTM Neural Network Model. LSTM neural network (hereinafter referred to as LSTM) is an improvement of the recurrent neural network, which aims to overcome the defects of the recurrent neural network in processing long-term memory [36]. The LSTM introduced the concept of cellular states, which determine which states should be preserved and which should be forgotten. The basic principle of LSTM is shown in Figure 1.

As shown in Figure 1, X_t is the input at time t, h_{t-1} is the output of the hidden layer at time t-1, and C_{t-1} is the output of the historical information at time t-1; f, i, and, o are, respectively, the forgetting gate, input gate, and output gate at time t, and g is the internal hidden state, namely, the transformed new information. LSTM conducts parameter learning for them in the training. C_t is the updated historical information at time t, and h_t is the output of the hidden layer at time t.

Firstly, the input x_t at time t and the output h_{t-1} of the hidden layer are copied into four copies, and different weights are randomly initialized for them, to calculate the forgetting gate f, input gate i, and output gate o, as well as the internal hidden state g. Their calculation methods are shown in formulas (1)–(4), where W is the parameter matrix from the input layer to the hidden layer, U is the self-recurrent

parameter matrix from the hidden layer to the hidden layer, *b* is the bias parameter matrix, and σ is the sigmoid function, so that the output of the three gates remains between 0 and 1.

$$f = \sigma \left(W_f x_t + U_f h_{t-1} + b_f \right), \tag{1}$$

$$i = \sigma \left(W_i x_t + U_i h_{t-1} + b_i \right), \tag{2}$$

$$o = \sigma \left(W_o x_t + U_o h_{t-1} + b_o \right), \tag{3}$$

$$g = \sigma \left(W_g x_t + U_g h_{t-1} + b_g \right). \tag{4}$$

Secondly, forgetting gate f and input gate i are used to control how much historical information C_{t-1} is forgotten and how much new information g is saved, to update the internal memory cell state C_t . The calculation method is shown in formula (5).

$$C_t = f_t \otimes C_{t-1} \oplus i \otimes g. \tag{5}$$

Finally, output gate o is used to control how much C_t information of the internal memory unit is output to the implicit state h_t , and its calculation method is shown in formula (6).

$$h_t = o \otimes \tan h(C_t). \tag{6}$$

3.2. Network Traffic Prediction Model Based on LSTM. Network traffic data are modeled as a nonnegative matrix X of an NxT, where N represents the number of nodes, T represents the number of time slots sampled, and each column in the data matrix represents the network traffic value at different nodes in a specific time interval.

Network traffic prediction can obtain the predicted value of the future time through the historical time series, X(i, j)represents the scale of the NxT flow matrix, and $x_{n,t}$ represents the network traffic value of row n and column t. Network traffic prediction is defined by a series of historical network traffic data $(x_{n,t-1}, x_{n,t-2}, x_{n,t-3}, ..., x_{n,t-1})$ to predict the network traffic at time t in the future. In the network traffic prediction model based on LSTM (Figure 2), it is assumed that the network traffic at a certain point in the tslot is predicted, the input of the model is $(x_{n,t-1}, x_{n,t-2}, x_{n,t-3}, ..., x_{n,t-1})$, and the output is the predicted value \hat{x}_t of the network traffic at the t-slot at this point.

In Figure 2, we summarize the process of network traffic prediction based on LSTM, and it mainly includes network traffic data preparation, data preprocessing (data resampling and null filling), normalization of data, data classification, prediction network building, network compilation, network evaluation, data prediction, and evaluation.

The detailed contents of each process for network traffic prediction are as follows:

 Network traffic data preparation and preprocessing. To meet the time and frequency requirements (second, minute, hour, day, etc.) of network traffic data prediction, the original data are required to resample, namely, the time series from one



FIGURE 2: Process of LSTM network traffic prediction.

frequency is converted to another frequency. And to ensure even data time interval, the uneven time interval data are converted to equal interval data. There are generally two methods of data resampling: downward sampling and upward sampling. The former is to convert high-frequency data into lowfrequency data, while the latter is to convert lowfrequency data into high-frequency data. In addition, if there is a void value in the resampled data sequence, it is necessary to fill the void value. The commonly used methods include the direct deletion method, statistically based filling method, and machine-learning-based filling method. The direct deletion method may discard some important information in the data, and the statistically based filling method ignores the timing information of the data [37]. Therefore, this paper adopts the machinelearning-based filling method-K-Nearest Neighbor (KNN) to fill the void value of network traffic data.

(2) Data normalization. The range standardization method is used to process the network traffic data so that the sample data value is between 0 and 1. The calculation method of the range standardization method is shown in formula (7).

$$X_{\rm nor} = \frac{X - X_{\rm min}}{X_{\rm max} - X_{\rm min}},\tag{7}$$

where X_{max} represents the maximum value of network traffic data and X_{min} represents the minimum value of network traffic data.

(3) Data division. After preprocessing and normalization, the network traffic data are divided into training set and test set according to the simple crossvalidation. Under the condition of keeping the network traffic data time sequence constant, the training set and the test set are divided by fivefold cross-validation [38], which are used for the training and prediction of the LSTM network traffic prediction model.

- (4) LSTM network traffic prediction model construction. An LSTM neural network is defined and its parameters are set, including the values of time step size, number of network layers, number of neurons in each layer, dropout, activation function, type and number of the return value, dimension size of the hidden layer, learning rate, batch processing size, iteration times, etc.
- (5) Network compilation. Set the optimizer, error measurement index, training record parameters, and compile the LSTM network traffic prediction model.
- (6) Network evaluation. Substitute training data into the model for training, and evaluate the error of the established prediction model. According to the results, finetune the parameter setting of the model to get a better prediction effect.
- (7) Prediction and evaluation. The optimized network traffic prediction model is used to make a prediction, and calculate the prediction errors by comparing prediction results with the real data.

4. The LSTM Network Traffic Prediction Model Optimized by IPSO and CEEMDAN

4.1. Improved Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a simple-rule, fast-convergence-speed swarm intelligence optimization algorithm [39, 40]. It regards every individual as a part with no size and no quality in an *n*-dimension search space, which flies at a certain speed. It improves the searching through group cooperation and competition among the particles under the guidance of swarm intelligence.

Particle swarm optimization in n-dimensional continuous search space, for *i*-th (i = 1, 2, ..., m) particle, determines that *n* dimensional current position vector $x^i(k) = [x_1^{i_1}x_2^{i_2}...x_n^{i_n}]^T$ represents the current position of the *i*-th particle in the search space, and *n* dimensional velocity vector $v^i(k) = [v_1^i v_2^i ... v_n^i]^T$ represents the search direction of the particle. The optimal position (pbest) experienced by the *i*-th particle in the group is denoted as $p^i(k) = [p_1^i p_2^{i_2} ... p_n^i]^T$, and the optimal position (gbest) experienced by all particles in the group is denoted as $p^g(k) = [p_1^g p_2^g ... p_n^g]^T$. The basic PSO algorithm is shown in formulas (8) and (9).

$$v_{j}^{i}(k+1) = \omega(k)v_{j}^{i}(k) + c_{1} \operatorname{rand}(0, a_{1})(p_{j}^{i}(k) - x_{j}^{i}(k)) + c_{2} \operatorname{rand}(0, a_{2})(p_{j}^{g}(k) - x_{j}^{i}(k)),$$
(8)

$$x_{j}^{i}(k+1) = x_{j}^{i}(k) + v_{j}^{i}(k+1),$$
(9)

where i = 1, 2, ..., m, j = 1, 2, ..., n, $\omega(k)$ is the inertia weight factor, c_1 and c_2 is acceleration constant, all of which are nonnegative values. rand $(0, a_1)$ and rand $(0, a_2)$ are random numbers with uniform distribution within the range of $[0, a_1]$ and $[0, a_2]$, a_1 and a_2 are corresponding control parameters.

In the PSO algorithm, ω keeps the particle moving inertia so that it tends to expand the search space, the ability to search new areas. The ω value usually adopts the linear inertia weight method, that is, the ω value increases or decreases linearly with the number of iterations. Compared with the fixed ω value, the linear method improves the optimization ability and convergence speed of the PSO algorithm to some extent, but it is far from enough. The nonlinear inertia weight method can further improve the optimization ability and convergence speed of the PSO algorithm [41]. Therefore, the ω calculation in this paper is improved by using the nonlinear inertia weight method, as shown in formula (10).

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * \arcsin \frac{i}{\text{item}_{\max}} * \frac{2}{\pi}.$$
 (10)

In formula (10), ω_{max} and ω_{min} , respectively, represent the maximum inertia weight and the minimum inertia weight, and *i* is the current iteration number. *item_*max is the maximum iteration number.

In the PSO algorithm, c_1 and c_2 are used to adjust the step size of particle movement. In this paper, the sine function is used to improve the acceleration constant [29]. The calculation method is shown in formulas (11) and (12).

$$c_1 = 2\sqrt{1 - \sin\left(\frac{\pi}{2} * \frac{i}{\text{item_max}}\right)},\tag{11}$$

$$c_2 = 2\sqrt{\sin\left(\frac{\pi}{2} * \frac{i}{\text{item}_{\text{max}}}\right)}.$$
 (12)

4.2. LSTM Hyperparameter Optimization Based on Improved PSO. The selection of hyperparameters of the LSTM prediction model has an important influence on prediction accuracy. The current hyperparameter selection method based on the empirical method has randomness, blindness, and nonuniversality in the parameter setting. Therefore, multiple hyperparameters are formed into a multidimensional solution space, and the optimal parameter combination is obtained by traversing the solution space, which can reduce the randomness and blindness of parameter selection. Multiple hyperparameter selections are in a larger scope, which needs a better performance optimization algorithm to obtain the global optimal solution quickly, so we introduce the improved particle swarm algorithm (Improved PSO, IPSO) to optimize LSTM model parameters. With the quick convergence speed, the IPSO promotes the scientific nature of the model parameter selection and further improves the prediction accuracy of the models.

It is assumed that n hyperparameters of the LSTM network traffic prediction model are optimized, each particle represents a set of hyperparameters of solution space. It is

supposed in the *n*-dimensional continuous search space, there are *m* groups of hyperparameter combinations, representing the *i*-th (*i* = 1, 2,...,*m*) hyperparameter. The current position vector $x^i(k) = [x_1^i x_2^i \dots x_n^i]^T$ of *n* dimension represents the current value of an *i*-th group of hyperparameters in the solution space. The velocity vector $v^i(k) = [v_1^i v_2^i \dots v_n^i]^T$ of *n* dimension represents the search direction of this group of hyperparameters.

The goal of network traffic prediction is to make the predicted value close to the actual value, that is, the error between the predicted value and the actual value is as small as possible. Therefore, the Root Mean Square Error (RMSE) of training data in the network traffic prediction model is selected as the objective function. Let fitness = RMSE, then the objective function is to minimize RMSE. The RMSE calculation method is shown in formula (13).

RMSE =
$$\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$
. (13)

In formula (13), \hat{y}_i is the prediction value. $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_i\}, y$ is the real value, $y = \{y_1, y_2, \dots, y_i\}.$

Two important hyperparameters of the LSTM network traffic prediction model are optimized according to IPSO: time step size and the number of neurons in each layer. The single-layer and bilayer LSTM models are taken as the research objects to carry out the hyperparameter optimization. For the single-layer LSTM model, the node is for the number of neurons, and the lookback is for the time step, fitness = RMSE (node, lookback); for a bilayer LSTM model, fitness = RMSE (node1, node2, lookback).

According to the algorithm flow of IPSO, the process of IPSO optimized LSTM network traffic prediction model hyperparameter mainly includes six steps.

Step 1. The IPSO parameter is set. The particle swarm size is set as the number of hyperparameter combinations *m*. Each particle is randomly set as the initial value and speed of each group of hyperparameters within the allowed range. The maximum number of iterations *item_max* and the prediction error *Pre_error*.

Step 2. The fitness of each particle is evaluated, that is, the fitness value of the objective function of each group of hyperparameters is calculated.

Step 3. The optimal objective function value P_i for each set of hyperparameters is set. For the *i*-th group hyperparameter, its current target function value current_fitness is compared with P_i . If it is less than P_i , then current_fitness is used as the best target function value P_i for the ith group hyperparameter, namely, P_i = current_fitness.

Step 4. The global optimal value P_g . For the hyperparameter of *i*-th group, P is compared with P_g . If it is less than P_g , then P_i is taken as the optimal value P_g of the current group, namely, $P_g = P_i$

Step 5. The search direction and value of each set of hyperparameters are updated according to formulas (8) and (9).

Step 6. The termination conditions are checked. If the set condition (default error or the maximum number of iterations) is not met, step 2 is returned to continue execution.

4.3. Network Traffic Data Decomposition by CEEMDAN. The empirical mode decomposition algorithm (EMD) is a data processing method commonly used for nonstationary time series signals [42]. It can decompose the nonstationary signals into a series of intrinsic mode function (IMF) components with different time scales. However, modal confusion exists in this method. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) algorithm improved the EMD algorithm by adding a set of white noise with equal size and opposite signs before decomposing data via the EMD [43]. The CEEMDAN both confuses modal confusion and also avoids making larger impacts on the original signal during adding the white noise. The main steps of CEEMDAN are as follows:

Add a group of Gaussian white noise sequence ε_i (t) with opposite signs to the original sequence x (t), and obtain a new set of time series;

$$\begin{cases} x_{i}^{+}(t) = x(t) + \varepsilon_{i}^{+}(t), \\ x_{i}^{-}(t) = x(t) + \varepsilon_{i}^{-}(t). \end{cases}$$
(14)

(2) Decompose each time series via EMD in formula (15) and obtain n intrinsic mode functions components;

$$\begin{cases} x_i^+(t) = \sum_{j=1}^n c_{ij}^+(t), \\ x_i^-(t) = \sum_{j=1}^n c_{ij}^-(t), \end{cases}$$
(15)

where, c_{ij} is the *j*-th modal component obtained by EMD decomposition after adding white noise for the *i*-th time.

- (3) Add different adaptive noises and repeat steps (14) and (15) for *m* times to obtain the set of *m* groups of intrinsic modal components (IMF), in which the last group is the trend term (Res);
- (4) Calculate the ensemble average of all components to obtain the final modal component group c_i (*t*).

$$c_i(t) = \frac{1}{2m} \sum_{j=1}^m \left(c_{ij}^+(t) + c_{ij}^-(t) \right).$$
(16)

The process of network traffic prediction based on IPSO-LSTM combined with CEEMDAN is shown in Figure 3.

The process of data decomposition and prediction includes three main steps.

- (1) Network traffic data preparation and preprocessing
- (2) Decompose the raw data into several different modal components and * obtain some subsequences of IMF1, IMF2, IMF3, ..., IMFn
- (3) Divide each subsequence into a training set and a test set
- (4) Construct the LSTM network traffic prediction model. Set partial parameters and fix the number n of the optimized parameter
- (5) IPSO parameter initialization (particle swarm size *m*, solving space dimension *d*, the maximum number of iterations iter_max, learning factor φ_1 , φ_2 , weight ω)
- (6) Initialize the values of n-dimensional parameter combinations of m groups randomly in the solution space
- (7) Initialize the global optimal parameter combination gbest_parameters, the partial optimal parameter combination pbest_parameters and the best fitness function value Pg
- (8) While the end condition is False
- (9) Apply the n-dimensional parameter combinations of *m* groups, respectively, to the LSTM network traffic prediction model for training, and calculate the current fitness function value;
- (10) Get the current best fitness value Pi and the corresponding parameter combination pbest_parameters;
- (11) if Pi < Pg;
- (12) $P_a = P_i$;//Update the best fitness value
- (13) gbest_parameters = pbest_parameters;//Update the global optimal parameter combination
- (14) end if;
- (15) for each parameter combination
- (16) Calculate the search direction and position of the new parameter combination according to equations (8) and (9)
- (17) Fix the updated parameter in the selected values;
- (18) end for;
- (19) The number of iterations + 1;
- (20) end while;
- (21) Return to gbest_parameters;
- (22) Introduce gbest_parameters into the LSTM network traffic prediction model;
- (23) Predict test data of each subsequence and gain results1, result2, result3, ..., resultsn;
- (24) Superpose the subsequence prediction results of results₁, result₂, result₃,..., results_n and output the network traffic prediction result.



ALGORITHM 1: CEEMDAN-IPSO-LSTM network traffic prediction algorithm.

- (2) Use the IPSO-LSTM model to predict each subsequence and gain results₁, result₂, result₃, ..., results_n;
- (3) Superpose the subsequence prediction results of results₁, result₂, result₃, ..., results_n and output the network traffic prediction result.

4.4. Network Traffic Prediction Algorithm Based on CEEM-DAN-IPSO-LSTM. According to the process of IPSO for hyperparameter optimization and data de-composition by CEEMDAN, based on the network traffic prediction steps of LSTM, the CEEMDAN-IPSO-LSTM network traffic prediction algorithm is obtained. The pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1 firstly prepares network traffic data and decomposes the raw data into several subsequences, and then divides each subsequence into a training set and a test set. Then, it uses the IPSO-LSTM network traffic model to obtain the optimal parameter combination. Finally, the optimal parameters are substituted into the LSTM model to complete the prediction of each subsequence and output the network traffic prediction result by superposing subsequence prediction results.

CEEMDAN-IPSO-LSTM network traffic prediction algorithm contains three processes, the time complexity of data decomposition is $O(k^2)$, k is the size of the predicted data set; the time complexity of hyperparameter optimization process is O(n!); and the time complexity of the

FIGURE 3: Process of data decomposition and prediction.

 The network traffic data are decomposed by CEEMDAN into serval different modal components and obtain some subsequences of IMF₁, IMF₂, IMF₃, ..., IMF_n; prediction process is $O(hp + h^2 + h)$, *h* is the hidden_size, *p* is the input_size. Therefore, the time complexity of CEEMDAN-IPSO-LSTM is

$$T(n) = O(k^{2}) + O(n!) + O(hp + h^{2} + h).$$
(17)

In the running process of the algorithm, the parameter optimization process consumes the most time with the highest computational complexity, but its time cost is acceptable because this process needs to run only once to obtain the optimal combination of hyperparameters. Once the hyperparameters are determined, the main time complexity is reflected in the prediction process. The time of the prediction process is mainly spent in the training. As long as the training is completed, the prediction can be finished by substituting the input data into the equation.

5. Experiment Evaluation and Discussion

5.1. Experimental Environment Configuration and Parameter Setting. This experiment completed under the measured flow data of BC-Oct89Ext provided by Bell Laboratory is selected. The flow data were Ethernet data detected in The Bell Morristown study, containing one million packets. This paper selects some data segments of BC-Oct89Ext flow data for model analysis.

For the prediction results of the network model, three error analysis indicators were used to verify the prediction accuracy, which were Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), respectively. MAE and MAPE calculation methods are shown in equations (18) and (19).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \widehat{y}_i|, \qquad (18)$$

MAPE =
$$\frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%.$$
 (19)

According to Equation (13), the smaller the RMSE value, the smaller the average error between the prediction results and the actual data, the higher the prediction accuracy of the model, and the better the prediction performance of the model. Similarly, it can be seen from equations (18) and (19) that the MAE and MAPE values tend to 0, the better the prediction effect of the model is and the more perfect the model is. On the contrary, the greater the value is, the greater the error is, and the worse the prediction effect of the model is.

5.2. Network Traffic Prediction Results Based on LSTM

5.2.1. Data Processing

(1) Data resampling. As the original network traffic data in BC-OCT89Ext were collected multiple times per second with unequal time intervals, the data collected multiple times per second were preprocessed with the mean value method, and then the K-Nearest Neighbor (KNN) algorithm was used to fill the void value. Figure 4 shows 1800 pieces of flow data after packet resampling and null value processing.

(2) Data decomposition. It can be seen from Figure 5 that network traffic data have obvious nonlinearity and non-stationarity, which makes prediction difficult. Then the original time series is decomposed by the CEEMDAN method into several more predictable time subseries, and six groups of modal subsequences were obtained from high frequency to low frequency. Decomposition results are shown in Figure 5. It can be seen that the fluctuation of IMF1 to Res subsequence gradually flattens out and the frequency becomes lower and lower.

(3) Data division. The data after normalization was divided into a training set and a test set according to simple crossvalidation. The first 80% of the data were used as training data for LSTM network model training. The remaining 20% of the data were used as prediction data to verify the efficiency of the model.

5.2.2. Network Traffic Prediction Based on Basic LSTM

(1) Network definition. In this forecast, the network structures of three-layer LSTM (one input layer, one hidden layer, and one output layer) and four-layer LSTM (one input layer, two hidden layers, and one output layer) are, respectively, adopted.

The specific connection mode of the three-layer LSTM is as follows: the timesteps of LSTM in the first layer are 1. The input of the data dimension is 3, and the number of neurons is 64. The second layer hidden layer (dense) takes the output of the first layer LSTM as input; the output layer of the third layer takes the output of the second hidden layer as the input and connects to a full connection layer. A one-dimensional vector with a length of 360 output from the full connection layer is the final output result, which represents the value of the predicted future 360 data points. To prevent overfitting, a dropout layer was added between the first layer and the hidden layer for regularization. After many tests in this experiment, it was concluded that when the dropout is 0.3, the training set had the highest accuracy.

Compared with the three-layer LSTM network, a hidden layer is added to the four-layer LSTM network structure. The hidden layer uses the results of the first layer as the input for training and transmits its output to the next hidden layer. The number of neurons is the same as that of the first layer. The dropout = 0.3 layer is added in both the first and second layers to prevent overfitting.

(2) Network compilation. LSTM network compilation uses the adaptive moment estimation (Adam) algorithm as the optimizer and the mean square error loss function as the objective function.

(3) Network fitness. The LSTM network was trained on 1440 samples and 360 samples were used for testing. The number of iterations epochs equals 50, look_back is made of 1, 5, and 10, respectively, and batch_size equals 128.

(4) Network evaluation. When look_back takes 1, 5, and 10, respectively, and the number of hidden layers (LN) is 1 and



FIGURE 4: Network traffic data after null filling.



FIGURE 5: Network traffic data after decomposition.

2, respectively, the loss data of the model training process is shown in Figure 6.

(5) Network traffic forecast. 360 test data were predicted, and the first 100 predicted results were shown in Figure 7. TestOriginal_result represents the original data, and test-Predict_result_101, testPredict_result_105, and testpredict_110, respectively, represent the prediction results when LN = 1, look_back takes 1, 5, and 10, respectively. Test-Predict_result_201, testPredict_result_205, and testPredict_210, respectively, represent the prediction results when LN = 2, and look_back takes 1, 5, and 10, respectively.

(6) Evaluate the prediction error of the model. The LSTM model under different parameter combinations was executed for network traffic prediction, and the indexes of RMSE, MAE, and MAPE for each validation set were calculated. The results were shown in Table 1.

It can be seen from Table 1 that the prediction error of the model changes with the look_back increases, and the prediction error of single-layer LSTM and double-layer LSTM is different under the same look_back. Based on the above experiments, four groups of experiments were added, namely, when look_back takes 15 and 20, and multiple predictions were made in the case of single-layer network







FIGURE 6: Model training loss under different parameter combinations.



test data and test predict data

FIGURE 7: Test results of partial data.

TABLE 1: Errors of LSTM prediction models under different parameter combinations.

Model	Parameters	RMSE	MAE	MAPE (%)
LSTM101	$(look_back, LN) = (1,1)$	71.82	41.68	32.17
LSTM105	$(look_back, LN) = (5, 1)$	66.81	40.39	32.77
LSTM110	$(look_back, LN) = (10,1)$	67.08	41.58	34.11
LSTM201	$(look_back, LN) = (1,2)$	71.65	42.28	33.29
LSTM205	$(look_back, LN) = (5,2)$	69.58	40.57	31.19
LSTM210	$(look_back, LN) = (10,2)$	70.43	43.01	34.97



FIGURE 8: Errors of LSTM prediction model under different parameter combinations.

and double-layer network, respectively, and corresponding error values were calculated. The test results are shown in Figure 8.

It can be seen from Figure 8 that the setting of the number of hidden layers and the time step has a great impact on the fitting effect of LSTM. When a hidden layer is added, the prediction error changes, and the increase or decrease of

the prediction error is not fixed at different timesteps. When the time step is changed, that is, the look_back value is changed from small to large, and the trend of prediction error is also not fixed. For example, when the look_back value changes from 5 to 10, the prediction error of the singlelayer LSTM model decreases, while the prediction error of the double-layer LSTM model increases.



FIGURE 9: The change in fitness value.



FIGURE 10: The change of parameter in LSTM12 during IPSO iteration (a) The change of node number. (b) The change of time step.

Therefore, for network traffic data, the prediction effect of the parameter combination set by the empirical method is unstable and cannot achieve the optimal prediction performance. Therefore, the Improved Particle Swarm Optimization (IPSO) will be adopted to carry out model optimization, that is, the intelligent algorithm will be used to efficiently obtain the parameter combination with the optimal prediction effect.

5.2.3. Parameter Optimization of LSTM Network Traffic Prediction Model Based on IPSO. The IPSO algorithm was used to optimize the LSTM network traffic prediction model, and parameters were optimized for single-layer LSTM and double-layer LSTM, respectively. The fitness value of the LSTM prediction model changed as the number of iterations increased during the experiment, as shown in Figure 9.

In Figure 9, fitness12, fitness23, and fitness22 correspond to the fitness values of the model IPSO-LSTM12 (2 parameters node1, lookback of the single layer), IPSO-LSTM23 (3 parameters node1, node2, lookback of the double layer), and IPSO-LSTM32 (2 parameters node1, lookback of double layer), respectively. The second parameter of IPSO-LSTM22 is set as node2 = 4 according to the optimization results of LSTM23.

It can be seen from Figure 9 that the final convergence value of fitness12 is less than fitness23 and fitness22, the convergence rate is faster than fitness23, and the fitness22 final convergence value is only slightly smaller than the fitness of 23. This shows that for the long-term prediction of network traffic data if the fitness value from a single hidden layer LSTM optimized by the particle swarm algorithm is slightly smaller than that from a two-layer hidden layer



FIGURE 11: The change of parameter in LSTM23 during IPSO iteration. (a) The change of node number. (b) The change of time step.



FIGURE 12: The comparison of prediction results of the four models.

LSTM optimized by the particle swarm algorithm, convergence speed is faster.

It can be seen that compared with the empirical method of setting LSTM parameters, the RMSE of the IPSO for setting LSTM parameters is reduced by 20%, which means that the IPSO algorithm can effectively find the optimal parameter combination of LSTM network traffic prediction and reduce the prediction error.

In addition, Figure 10 shows the changes in node number and time step size during the IPSO-LSTM12 model optimization that shows the process of the optimal parameter value of the LSTM network traffic model determined by the improved PSO algorithm. It can be seen from Figure 10 that the optimal parameters of the LSTM12 model are set as node1 = 8 and look back = 1. Therefore, in the prediction of network traffic data used in this paper, the optimal configuration of the single-layer LSTM model is to set the number of neurons to 8 and the time step to 1.

The changes in node number and time step size in IPSO-LSTM23 model optimization are shown in Figure 11.

It can be seen from Figure 11 that the optimal parameters of LSTM23 model are set as node1 = 16, node2 = 4, and look back = 1. Therefore, in the prediction of network traffic data used in this paper, the optimal configuration of the two-layer LSTM model is to set the number of network neurons in the

Security and Communication Networks

TABLE 2: Errors of LSTM prediction models under different parameter combinations.

Index	IPSO-LSTM12	IPSO-LSTM22	IPSO-LSTM23-1	IPSO-LSTM23-2
RMSE	46.93	46.77	47.20	46.93
MAE	21.61	40.71	42.47	41.14
MAPE (%)	20.67	31.28	33.07	32.58



FIGURE 13: The predictive performance of each IMF.

first layer to 16, the number of neurons in the second layer to 4, and the time step length to 1.

To evaluate the prediction performance of the LSTM model after parameter optimization by IPSO, network traffic data samples at 180 time points are used for verification. In this paper, the IPSO optimized single-layer LSTM IPSO—LSTM12, double parameter optimization model IPSO - LSTM22 of double-layer LSTM, three parameters optimization model IPSO LSTM23-1 (no dropout in training) of double-layer LSTM, three-parameter optimization model of IPSO LSTM23-2 (dropout in training) of double-layer LSTM are compared, and Figure 12 shows the model prediction results for the last 180 test data.

It can be seen from Figure 12 that the prediction results of the LSTM model with different parameter combinations have a good fitting effect, and the prediction results of the single-layer LSTM dual-parameter optimization model IPSO-LSTM12 are better than those of other parameter configuration models. To compare the predictive performance of the four models more clearly, the predictive performance evaluation index values of the four models in Figure 12 are obtained, respectively, and the results are shown in Table 2.

As it can be seen from Table 2, compared with singlelayer LSTM12, two-layer LSTM22 has slightly fewer prediction errors in RMSE and MAE, while MAPE is slightly bigger. If only RMSE or MAE evaluation index is considered, LSTM22 is better than LSTM12, while only MAPE evaluation indicators are considered, LSTM12 is considered better than LSTM22. On the whole, the prediction error of LSTM12 and LSTM22 is less than that of the other three prediction models, that is, the prediction effect of LSTM12 and LSTM22 on network traffic data is better than that of the other three models. The prediction error of LSTM23-2 is less than that of LSTM23-1, which indicates that the optimization of dropout parameters added in the training reduces the prediction error of the model and improves the prediction performance of the model.

5.2.4. Network Traffic Prediction Based on CEEMDAN-IPSO-LSTM. Through testing on a 500-time data set, the predictive performance of each IMF is shown in Figure 13.



FIGURE 14: The CEEMDAN-IPSO-LSTM forecasting flowchart.



FIGURE 15: The comparison of prediction results of the eight models (a) Prediction results of CEEMDAN-IPSO-LSTM and other neural network methods (b) Prediction results of CEEMDAN-IPSO-LSTM and other regression methods.

TABLE 3: The prediction error with five classical models.

Index	CEEMDAN-IPSO-LSTM	CEEMDAN-LSTM	IPSO-LSTM	LSTM	ARIMA	SVR	DTR	MLR
RMSE	46.93	51.77	47.20	52.53	90.22	65.59	89.78	71.74
MAE	21.61	27.61	42.47	41.14	53.98	37.58	51.42	41.52
MAPE (%)	20.67	27.94	33.07	32.58	43.79	25.61	49.15	32.19



CEEMDAN for Network Traffic Decomposition



FIGURE 16: The decomposition results of three decomposition methods.



FIGURE 17: The predictions by LSTM combining the three decomposition methods.

Figure 13 shows the prediction results and training loss of eight IMFs and it has a better prediction effect. IMF0 and IMF7 are a little poor, in which the loss of the training set is very high during the whole training process. Especially, the loss of IMF0 is relatively large. For the remaining IMFs, LSTM predicts them well. Despite this problem, the overall results were excellent when the predictions were integrated.

After finishing predicting all IMFs, the final prediction result is integrated by superimposing the predicted results of each IMF. Figure 14 shows the forecasting flowchart of CEEMDAN-IPSO-LSTM.

5.3. Result Analysis. To evaluate the prediction effect of the proposed hybrid method CEEMDAN-IPSO-LSTM, it is compared with other neural network prediction methods like CEEMDAN-LSTM, IPSO-LSTM, and LSTM, and other predictive models like ARIMA, Support Vector Regression (SVR), Decision Tree Regressor (DTR), and Multivariate Linear Regression (MLR). Similarly, the network traffic data samples at 180-time points were used for verification, and the prediction results of the eight models are shown in Figure 15.

Figure 15 shows that the prediction effects of different models and the hybrid prediction model have a better fitting effect which indicates that the prediction results of the CEEMDAN-IPSO-LSTM model are better than those of other models. To compare the prediction performance of the eight models more clearly, their predictive performance evaluation index values were obtained, respectively, and shown in Table 3.

It can be seen from Table 3, that the prediction errors of the LSTM-based model are all less than regression

prediction models, which indicates that the LSTM network traffic prediction model has a better prediction effect than other regression network traffic prediction models. In other words, the LSTM is more suitable for solving long-term network traffic data prediction and processing real-time variability of network traffic data. In addition, the RMSE, MAE, and MAPE index values of the CEEMDAN-IPSO-LSTM prediction model are all smaller than other neural network prediction models, indicating that the proposed hybrid model CEEMDAN-IPSO-LSTM is better than other prediction models in network traffic prediction.

Besides, we make comparisons of decomposition methods like EMD, EEMD, and CEEMDAN. Firstly, based on a 500-time network flow data, we decompose the original data into several IMFs and compare the decomposition results of three decomposition methods. Then, we make predictions by LSTM methods combining the three decomposition methods to explain which method works better.

In Figure 16, there are seven IMFs of EMD, eight of EEMD, and eight of CEEMDAN, including residue. We only know that different decomposition results make the prediction accuracy different, but it is hard to see which one produces the better prediction. So, we make predictions by LSTM combining the three decomposition methods and the results are in Figure 17.

Figure 17 shows the red lines fit the raw data more closely, which shows that the predicted result of CEEM-DAN-LSTM is closer to the real value. To further verify the effect of different decomposition methods, Table 4 gives the prediction error of CEEDAN-LSTM, EEMD-LSTM, and EMD-LSTM.

Security and Communication Networks

Index	CEEMDAN-LSTM	EEMD-LSTM	EMD-LSTM
RMSE	14.12	22.74	36.66
MAE	10.94	17.46	30.65
MAPE (%)	15.69	16.35	26.05

TABLE 4: The prediction error of LSTM combining different decomposition methods.



FIGURE 18: The prediction results of the state-of-the-art models.

TABLE 5: The prediction error of the state-of-the-art models.

Index	CEEMDAN-IPSO-LSTM	SA-ARIMA-LSTM	ST-LSTM	INGARCH
RMSE	8.59	10.79	13.45	14.12
MAE	6.9	8.23	10.00	10.94
MAPE (%)	8.02	9.18	12.20	15.69

In Table 4, the prediction error of CEEMDAN-LSTM is significantly less than the other two methods, which indicates CEEMDAN can decompose data more effectively so that LSTM can predict better. That is to say, the results verify the superiority of CEEMDAN for data decomposition.

Also, based on the same 500-time network flow data, we compare CEEMDAN-IPSO-LSTM with *s* three state-of-theart prediction models to verify the effectiveness of the proposed network traffic prediction model, like ST-LSTM, SA-ARIMA-BPNN, and INGARCH. The last 100-time prediction data of the four methods are in Figure 18.

In Figure 18, the four prediction methods do a good job of forecasting network traffic. Figure 18 shows that the purple

and green lines match the raw data represented by the blue lines better, which demonstrates that the proposed method and the SA-ARIMA-LSTM make more effective predictions close to reality. To compare the prediction accuracy of the four methods more clearly, the prediction error of the four methods is calculated similarly and shown in Table 5.

In the same appearance as Figure 18, CEEMDAN-LSTM has the lowest prediction error. The appearances of Figure 18 and Table 5 prove the superiority of the CEEMDAN-IPSO-LSTM in this paper once again.

Above all, the CEEMDAN-IPSO-LSTM has a better prediction effect and higher reliability for the future prediction of network traffic.

6. Conclusion and Future Work

Network traffic prediction can be applied to network resource optimization and network congestion avoidance, which makes great significance for network business planning, data management, fault detection, resource allocation, and other operations. In this paper, a hybrid deep interval prediction model has been proposed for network traffic forecasting to improve the prediction accuracy. Firstly, the nonparametric LSTM neural network is used to establish the network traffic prediction model, and the Improved Particle Swarm Optimization algorithm is used to optimize the hyperparameters of the established LSTM prediction model, and further obtain the optimized LSTM network prediction model-IPSO-LSTM12,IPSO-LSTM23 and IPSO-LSTM32-which reduces the RMSE by 20% compared to the Experience-based LSTM. Besides, the prediction performance of single-layer LSTM is better than double-layer LSTM in network traffic prediction. Then the CEEMDAN is introduced to decompose the network traffic time series into different modes to reduce the complexity of the network traffic sequence. To verify the effectiveness of the proposed models, the proposed CEEMDAN-IPSO-LSTM model is applied to network traffic prediction and compared with other neural network prediction methods and regression methods. The experimental results show that compared with other prediction models and the traditional LSTM model, the CEEMDAN-IPSO-LSTM model reduces the prediction error and obtains a better fitting effect, which demonstrates that the proposed hybrid method improves network traffic prediction accuracy.

In future work, we plan to enhance the prediction model from two aspects to further improve the prediction accuracy of network traffic. On the one hand, in the data preprocessing stage, we will try other data decomposition methods, such as Variational Mode Decomposition (VMD), wavelet packet, and combination method, to improve the stability and regularity of network traffic data decomposition. On the other hand, we will focus more on the error correction strategy of the hybrid model of network traffic forecasting, such as analysis of different error correction strategies, or re-decompose the IMF data, to enhance the prediction performance.

Data Availability

The 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.

Acknowledgments

This work was funded by the National Natural Science Foundation of China under Grant No. 62072363, the National Natural Science Foundation of China under Grant No. 61672416, Industrialization projects of the Education Department of Shaanxi Province under Grant No. 21JC017, Industrial research project of Science and Technology Department of Shaanxi Province under Grant No. 2020GY-012, and Science and Technology Plan Project of Yulin under Grant No. 2019-175.

References

- S. Iqbal, K. N. Qureshi, F. Shoaib, A. Ahmad, and G. Jeon, "Minimize the delays in software defined network switch controller communication," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 13, p. e5940, 2020.
- [2] L. Liu, S. Guo, G. Liu, and Y. Zeng, "Priority based online flow scheduling for network throughput maximization in software defined networking," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 9, p. e5633, 2020.
- [3] M. Gupta and A. Sinha, "Distributed Temporal Data Prediction Model for Wireless Sensor Network," *Wireless Personal Communications*, vol. 119, no. 4, pp. 3699–3717, 2021.
- [4] W. Jiang, "Cellular traffic prediction with machine learning: A survey," *Expert Systems with Applications*, vol. 201, Article ID 117163, 2022.
- [5] C. Pan, Y. Wang, H. Shi, J. Shi, and R. Cai, "Network Traffic Prediction Incorporating Prior Knowledge for an Intelligent Network," *Sensors*, vol. 22, no. 7, p. 2674, 2022.
- [6] N. Jiang, Y. Deng, O. Simeone, and A. Nallanathan, "Online Supervised Learning for Traffic Load Prediction in Framed-ALOHA Networks," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1778–1782, 2019.
- [7] Y. Ji, Y. Wu, D. Zhang et al., "A Novel Flash P2P Network Traffic Prediction Algorithm based on ELMD and Garch," *International Journal of Information Technology and Decision Making*, vol. 19, no. 01, pp. 127–141, 2020.
- [8] C. Katris and S. Daskalaki, "Dynamic bandwidth allocation for video traffic using FARIMA-based forecasting models," *Journal of Network and Systems Management*, vol. 27, no. 1, pp. 39–65, 2019.
- [9] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting Inter-Data-Center Network Traffic Using Elephant Flow and Sublink Information," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 1–792, 2016.
- [10] L. Nie, X. Wang, S. Wang et al., "Network Traffic Prediction in Industrial Internet of Things Backbone Networks: A Multitask Learning Mechanism," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7123–7132, 2021.
- [11] M. Kim, "Network traffic prediction based on INGARCH model," Wireless Networks, vol. 26, no. 8, pp. 6189–6202, 2020.
- [12] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, Wuhan, China, November 2016.
- [13] L. Mingyan and T. Sucheng, "Large Scale Network Data Flow Parallel Forecasting Method Simulation," *Computer Simulation*, vol. 35, no. 08, pp. 206–209, 2018.
- [14] H. Yang, X. Li, W. Qiang, Y. Zhao, W. Zhang, and C. Tang, "A network traffic forecasting method based on SA optimized ARIMA-BP neural network," *Computer Networks*, vol. 193, Article ID 108102, 2021.
- [15] S. Hu and Z. Yuxue, "Network Traffic Modeling and Forecasting based on ARIMA," *Communications Technology*, vol. 52, no. 04, pp. 903–907, 2019.

- [16] C. Katris and S. Daskalaki, "Comparing forecasting approaches for Internet traffic," *Comparing forecasting approaches for Internet traffic Expert Systems with Applications*, vol. 42, no. 21, pp. 8172–8183, 2015.
- [17] C. Xiang, Q. Peixin, and Q. Xilong, "Network Traffic Prediction Based on MK-SVR," *Journal of Information and Computational Science*, vol. 12, no. 8, pp. 3185–3197, 2015.
- [18] X. Chen, Y. Liu, and J. Zhang, "Traffic prediction for Internet of Things through support vector regression model," *Internet Technology Letters*, vol. 5, no. 3, p. e336, 2022.
- [19] X. Zheng, W. Lai, H. Chen, and S. Fang, "Data Prediction of Mobile Network Traffic in Public Scenes by SOS-vSVR Method," *Sensors*, vol. 20, no. 3, pp. 603–619, 2020.
- [20] T. De Schepper, M. Camelo, J. Famaey, and S. Latre, "Traffic classification at the radio spectrum level using deep learning models trained with synthetic data," *International Journal of Network Management*, vol. 17, no. 4, p. 30, 2020.
- [21] W. Yong, Z. Huiyi, F. Hao, Y. Miao, and E. Wenlong, "Network traffic classification method basing on CNN," *Journal on Communications*, vol. 39, no. 01, pp. 14–23, 2018.
- [22] S. Nihale, S. Sharma, L. Parashar, and S. Upendra, "Network traffic prediction using long short-term memory," in *Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 338– 343, IEEE, Coimbatore, India, July 2020.
- [23] W. Shihao, Z. Qinzheng, Y. Han, and L. Qiamnu, "A Network Traffic Prediction Method Based on LSTM," *ZTE Communications*, vol. 17, no. 2, pp. 19–25, 2019.
- [24] L. Xiaolin and W. Teng, "Efficient network traffic prediction method based on PF-LSTM network," *Application Research of Computers*, vol. 36, no. 2, pp. 3833–3836, 2019.
- [25] Z. Tian and P. Song, "A novel network traffic combination prediction model," *International Journal of Communication Systems*, vol. 35, no. 7, p. e5097, 2022.
- [26] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (*ICACCI*), pp. 2353–2358, IEEE, Udupi, India, September 2017.
- [27] G. Fang, C. Lei, and Y. Ziwen, "Real-time traffic prediction based on MGU for large-scale IP backbone networks," *Journal* of Shandong University, vol. 49, no. 2, pp. 88–95, 2019.
- [28] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *The Journal of Supercomputing*, vol. 76, no. 3, pp. 2098–2118, 2020.
- [29] Y. L. Zhou, R. J. Han, Q. Xu, Q. Jiang, and W. Zhang, "Long short term memory networks for CSI300 volatility prediction with Baidu search volume," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 10, Article ID e4721, 2019.
- [30] B. Shao, M. Li, Y. Zhao, and G. Bian, "Nickel Price Forecast Based on the LSTM Neural Network Optimized by the Improved PSO Algorithm," *Mathematical Problems in Engineering*, vol. 2019, Article ID 1934796, 15 pages, 2019.
- [31] Z. Jianhui, D. Ting, and C. Bo, "AQI Prediction Based on Long Short-Term Memory Model with.Spatial-Temporal Optimizations and Fireworks Algorithm," *Journal of Wuhan University (Natural Science Edition)*, vol. 65, no. 3, pp. 250–262, 2019.
- [32] Z. Dianfan, L. Zihao, and C. Shuhong, "Research on Model Temperature Prediction Method Based on Fireworks Algorithm and Long and Short Time Memory Network," Acta Metrologica Sinica, vol. 41, no. 006, pp. 640–645, 2020.

- [33] A. Guo, A. Jiang, J. Lin, and X. Li, "Data mining algorithms for bridge health monitoring: Kohonen clustering and LSTM prediction approaches," *The Journal of Supercomputing*, vol. 76, no. 2, pp. 932–947, 2020.
- [34] L. Lian and Z. Tian, "Network traffic prediction model based on ensemble empirical mode decomposition and multiple models," *International Journal of Communication Systems*, vol. 34, no. 17, Article ID e4966, 2021.
- [35] J. Bi, X. Zhang, H. Yuan, J. Zhang, and M. Zhou, "A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM," *IEEE Transactions* on Automation Science and Engineering, vol. 19, pp. 1869– 1879, 2021.
- [36] G. Zhang, L. Xu, and Y. Xue, "The time dependency predictive model on the basis of community detection and long short term memory," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 24, Article ID e4184, 2017.
- [37] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A Novel K-Means Clustering Algorithm with a Noise Algorithm for Capturing Urban Hotspots," *Applied Sciences*, vol. 11, no. 23, Article ID 11202, 2021.
- [38] Z. H. Zhang, F. Min, G. S. Chen, S. Peng, and Z. Cheng, "Tripartition state alphabet-based sequential pattern for multivariate time series," *Cognitive Computation*, vol. 18, pp. 1–19, 2021.
- [39] W. Deng, X. Zhang, Y. Zhou et al., "An enhanced fast nondominated solution sorting genetic algorithm for multi-objective problems," *Information Sciences*, vol. 585, pp. 441–453, 2022.
- [40] H. Cui, Y. Guan, H. Chen, and W. Deng, "A Novel Advancing Signal Processing Method Based on Coupled Multi-Stable Stochastic Resonance for fault detection," *Applied Sciences*, vol. 11, no. 12, p. 5385, 2021.
- [41] L. Wan, F. Fenling, and J. Qiwei, "Prediction for railway passenger volume based on modified PSO optimized LSTM neural network," *Journal of Railway Science and Engineering*, vol. 15, no. 12, pp. 3274–3280, 2018.
- [42] H. Cui, Y. Guan, and H. Chen, "Rolling Element Fault Diagnosis Based on VMD and Sensitivity MCKD," *IEEE Access*, vol. 9, Article ID 120308, 2021.
- [43] F. Zhou, Z. Huang, and C. Zhang, "Carbon price forecasting based on CEEMDAN and LSTM," *Applied Energy*, vol. 311, Article ID 118601, 2022.