

Research Article

A Hybridly Optimized LSTM-Based Data Flow Prediction Model for Dependable Online Ticketing

Chunmei Fan,¹ Jiansheng Zhu,² Haroon Elahi ,³ Lipeng Yang,² and Beibei Li²

¹Graduate Department, China Academy of Railway Sciences, Beijing 100081, China

²Institute of Computing Technologies, China Academy of Railway Sciences, Beijing 100081, China

³School of Computer Science, Guangzhou University, Guangzhou 510006, China

Correspondence should be addressed to Haroon Elahi; haroon.elahi@egzhu.edu.cn

Received 10 March 2021; Accepted 16 May 2021; Published 8 June 2021

Academic Editor: Xiao Zhang

Copyright © 2021 Chunmei Fan 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.

Fifth-generation (5G) communication technologies and artificial intelligence enable the design and deployment of sophisticated solutions for enhanced user experience and superior network-based service delivery. However, the performance of the systems offering 5G-based services depends on various factors. In this paper, we consider the case of the online railway ticketing system in China that serves the needs of hundreds of millions of people daily. This system's online access rates vary over time, and fluctuations are experienced, affecting its overall dependability and service quality. We use long short-term memory network, particle swarm optimization, and differential evolution to construct DP-LSTM—a hybridly optimized model to predict network flow for dependable and quality-enhanced service delivery. We evaluate the proposed model using real data collected over six months from the “12306 online ticketing” system. We compare the performance of the proposed model with mainstream network traffic prediction models. We use mean absolute percentage error, mean absolute error, and root mean square error for performance evaluation. Experimental results show the superiority of the proposed model.

1. Introduction

Fifth-generation (5G) communication technologies and artificial intelligence (AI) enable the design and deployment of sophisticated solutions for enhanced user experience and superior mobile service delivery meeting diverse critical requirements [1–4]. However, the performance of 5G-based services is dependent on many factors [5, 6]. 5G-based mobile services face challenges such as transferring of high data rates, rapid response requirements, dynamic coupling and decoupling of new devices, and their remote configuration [7]. Consequently, the overall data traffic associated with the service delivery systems grows tremendously [8]. Likewise, while 5G infrastructures support high data throughput and AI-based knowledge driven data layers try to address corresponding resource allocation and optimization challenges, Internet is a limited resource and new situations can affect its performance (<https://www.nytimes.com/2020/03/26/business/coronavirus-internet-traffic-speed.html>), which can lead to deteriorated service delivery.

Here, we need to remember that the performance of Internet-based services is not entirely dependent on the telecommunication infrastructure, and the scalability of servers entertaining service requests also plays a critical role [9]. Network congestion can slow down the service. Unexpected fluctuations in service request rates and corresponding changes in Internet traffic can directly affect the availability of web-based systems, and a substantial change in traffic may even result in crashing various application services (<https://techcrunch.com/2018/12/26/alexa-crashed-on-christmas-day/>). Consequently, we see that special measures and strategic approaches are used for load balancing, enhanced stability, and detecting and mitigating malicious actors affecting the availability of Internet-based services and systems [9–11].

Moreover, the requirements and challenges of Internet-based services vary for different application scenarios [12–14]. As a result, recent research has focused on application-specific issues investigating underlying factors affecting the performance of corresponding Internet-based services and proposing different optimization methods for improved

availability and Quality of Service (QoS) [15, 16]. In this paper, we focus on Chinese railway system that serves the needs of hundreds of millions of people daily. A key component of the Chinese railway system is the “12306 ticketing system (<https://www.highspeed.mtr.com.hk/en/ticket/buy-ticket-12306.html>)” that serves as the main channel for passengers to check the availability of tickets and make online bookings. This system can experience occasional incoming traffic fluctuations [17]. Particularly, with the wide-scale availability of high-speed Internet over 5G infrastructures, the number of people seeking the services through “12306 ticketing system” has grown significantly and ensuring its scalability and dependability is a significant challenge in its own.

This research observes and predicts the Internet traffic generated by the incoming service requests in the “12306 ticketing system” to propose a model that enables scalability and resilience in the event of Internet traffic surges. The main goal of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the 12306 ticketing system. Since, AI and data analytics can play an important role to control and efficiently operate networks and for efficient service delivery [18–21], we use AI-based data analytics to achieve our goals.

There service requests and related network access flows are encountered at successive time intervals. Therefore, it can be treated as time series data. In other words, the problem of predicting network access flow for the “12306 ticketing system” resembles traffic flow forecasting and stock forecasting [22–24]. Conventionally, there are three main approaches used to solve these problems [25, 26]: traditional algorithms such as exponential smoothing and autoregressive integrated moving average (ARIMA) [27, 28]; traditional machine learning algorithms such as support vector regression (SVR) [28], eXtreme Gradient Boosting (XGBoost) [29, 30], and Random Forest; and deep learning algorithms, such as Deep Autoregressive Networks (DARNs) [31] and long short-term memory (LSTM) [32, 33]. Among these algorithms, the ARIMA requires stable data. It predicts network traffic flows considering the variations in the historical data. Therefore, it cannot predict nonlinear patterns, and its generalization ability is weak [34, 35].

Likewise, due to network flow’s complex nature and the lack of typical behavior [36], traditional models cannot handle such data well. The LSTM network [32] has achieved good results in nonperiodic event detection [37], traffic load balancing [38], and other fields [8, 39]. It has shown promising performance in time series data trend prediction [34]. With its nonlinear approximation function and self-learning and self-adaptive features, LSTM can better describe the characteristics of time series data and achieve high prediction accuracy and strong generalization [33]. It is mainly used to describe the relationship between current data and previous input data and uses its memory ability to save the state information before the data is fed in the network and use the previous state information to influence the exact value and development trend of subsequent data. In LSTM, appropriate number of layers of and the number of hidden neurons in the feedforward network layer play key role in its performance. Research shows that increasing the number of network layers does not necessarily improve the effect, and

selecting the appropriate network layers can train a highly accurate model [40, 41].

However, in the actual application of LSTM network, determining the network structure and parameter selection are challenging tasks. It is generally achieved by hit and trial method or based on experience. This is also a bottleneck in the development of neural network [41]. Particle swarm optimization (PSO) is a heuristic random search algorithm, which has a lower number of setting parameters, no update and mutation involved, and it can find the extreme function values faster. Some LSTM models use particle swarm optimization (PSO) algorithm to find the optimal super parameters and achieve good results [34, 38, 39]. However, the PSO algorithm converges faster in the early stage of the optimization process and is easy to fall into the local optimum in the later stage. In order to be able to establish an optimal model, it is proposed to realize the parameter selection of the LSTM traffic prediction model by fusing the particle swarm optimization (PSO) and the differential evolution (DE) algorithm [42]. Using DE to optimize the evolution of PSO can improve the results of PSO [43] and greatly reduce the probability of obtaining a local optimal solution.

In this paper, we use LSTM—optimizing it through a fusion of PSO and DE to construct the DP-LSTM model that we use for the access flow prediction of 12306 ticketing system. The purpose of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the 12306 ticketing system.

Specifically, the contributions of this research can be listed as follows.

- (1) We focus on “the 12306 ticketing system” to learn its flow data patterns and predict future traffic to minimize the impact of sudden fluctuations for improving its dependability and overall QoS
- (2) We use long short-term memory (LSTM) network and hybrid optimization algorithm to construct the DP-LSTM model to predict network access traffic
- (3) We evaluate the proposed model using real data collected over six months from the “12306 ticketing system” and compare its performance with mainstream time series data forecasting methods. We use mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE) for performance evaluation. Experimental results show the superiority of the proposed model over the benchmarks

2. Access Flow Data in “12306 Ticketing System”

As mentioned in the previous section, service requests and related network access flows are encountered at successive time intervals. Such data exhibits secular trend, cyclical fluctuations, and irregular variations [44]. Therefore, it can be expected that operations of the “12306 ticketing system”

can be affected by the time-varying cyclic and irregular variations. The system serves from 7 to 23, and we focused on the hourly peak flows and collected the data from March 1, 2020, to August 31, 2020. We conducted an exploratory analysis on the collected data to identify timing sequence and outliers. The results of this analysis are provided underneath.

2.1. Timing Sequence. By selecting the one-month data from March 1st, 2020, to April 1st, 2020, and observing the variation law of request flow, the timing sequence diagram obtained is shown in Figure 1. The timing diagram indicates that the flow rate fluctuations had evident periodicity and trend. It is basically in accord with the Internet ticketing system's daily operation schedule of suspending ticket booking from 0:00 am to 5:00 am and opening from 6:00 am to 11:00 pm. On March 5th, the flow showed an upward trend, because the tickets can be prebooked for April 4th, the Tomb Sweeping Day holiday. Every morning, the Internet ticketing system becomes available at 6 o'clock for ticket booking, resulting in a daily peak at 6 am. The stability of access flow data is further tested through autocorrelogram and unit root detection. The P value of the unit root check was 0.91. From the autocorrelogram shown in Figure 2, it can be found that it is without truncation. Hence, the request flow is not a stationary sequence.

2.2. Outlier Detection. We detected abnormal points of the data flow through the boxplots. The flows during the ticketing period (6:00 am-11:00 pm) and the nonticketing period (0:00-5:00am) were quite different. Therefore, we divided the data into the ticketing period and the nonticketing period to be tested. The test results are shown in Figure 3.

These red points represent abnormal data which are greater than $Q3+2IQR$. $Q1$ is the first quantile, $Q3$ is the third quantile, and $IQR = Q3 - Q1$. The abnormal data during the nonticketing period were the data at 5:00 am every day, which were the normal flow fluctuation before the ticket release. The abnormal data during the ticketing period appeared in the ticket prebooking for the Qingming Festival (from March 4 to March 9) and 6:00 am. The data were also normal business traffic fluctuations. Through outlier detection, there is no extreme abnormal data in the collected data. There is a big difference in traffic during ticketing and nonticketing periods, so the prediction models need to be built separately.

The analysis results indicated that the access flow of the Internet ticketing system was not only unsteady and periodic but also affected by holidays and ticket release timing, which was random and complicated. For the traditional timing sequence models, it is difficult for them to fit such time sequence data.

3. High-Level Design of DP-LSTM

In this section, we introduce different components that we use to construct the DP-LSTM model for predicting flow data for the "12306 ticketing system."

3.1. LSTM. The LSTM is a recurrent neural network (RNN), which effectively learns the long-term dependency relation-

ship with well-designed "gated" architectures. It consists of memory cells and gate units. An LSTM neuron has input (multiplicative input), output (multiplicative output), and a forget gate. As the name suggests, the input gate handles input data stored in a given memory cell and protects it from perturbation by other irrelevant inputs. The output gate contains the output representations, and the forget gate handles the retention of historical information or, in other words, decides when to forget retained historical information. A typical LSTM network can have at least one input layer, one output layer, and a hidden layer. Memory cells and gate units are located in the hidden layer. When using LSTM, determining the network structure is a challenge, and it is often based on experience. Since single-layer LSTM is limited by the number of convolution kernels, multilayer LSTM can be a better choice [34]. Therefore, we use a multilayer LSTM in the network structure of the access flow prediction model of the railway ticketing system. At the same time, the dropout layer is added to improve the generalization ability of the model. Moreover, different studies have used different optimization algorithms to find the optimal LSTM prediction model [45–48]. We use a fusion of particle swarm optimization (PSO) and differential evolution (DE). The basic network structure is shown in Figure 4.

3.2. Particle Swarm Optimization (PSO). The PSO is a population-based stochastic algorithm for optimization. It searches for the optimal zone through the continuous interactions of population members (data), leading to an iterative improvement in the algorithm's performance [49]. It moves on its own in the search space and tests different parameters. It uses the group's optimal fitness to change the direction and distance of movement to complete the global search space's optimization process. Consider a group $X = \{x_1, x_2, \dots, x_n\}$, consisting of m particles in a d -dimensional search space, and $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ is a group of training parameters of the DP-LSTM model. If x_i is featured with location $X_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{id}^t]^T$ and velocity $V_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{id}^t]^T$ individual optimal location: $p_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{id}^t]^T$ and global optimal location: $p_g^t = [p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t]^T$ at the time t , the global-optimal location is the optimal parameter combination of the current training model; then, velocity and location of the particle at the time $t + 1$ can be updated to

$$\begin{cases} v_{id}^{t+1} = wv_{id}^t + c_1r_1^t(p_{id}^t - x_{id}^t) + c_2r_2^t(p_{gd}^t - x_{id}^t), \\ x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \end{cases} \quad (1)$$

where w is inertia weight which controls the effective equilibrium between global detection and local mining of the particle; c_1 and c_2 are learning factors which, respectively, adjust the step size flying to its own and global-optimal location; r_1 and r_2 are random numbers uniformly distributed within $[0,1]$.

While PSO has its advantages, there is a possibility that after iteratively changing the optimal parameter combination with Equation (1), subsequent particle update is stuck into local

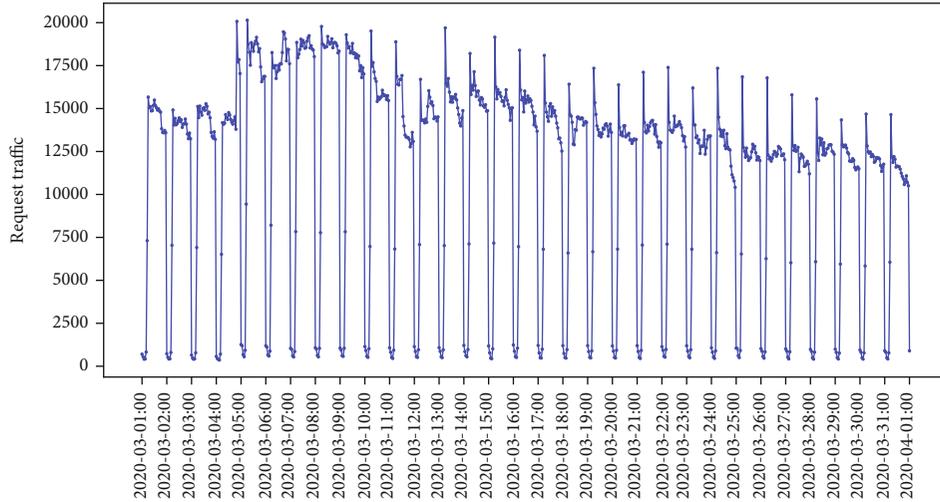


FIGURE 1: The timing sequence diagram.

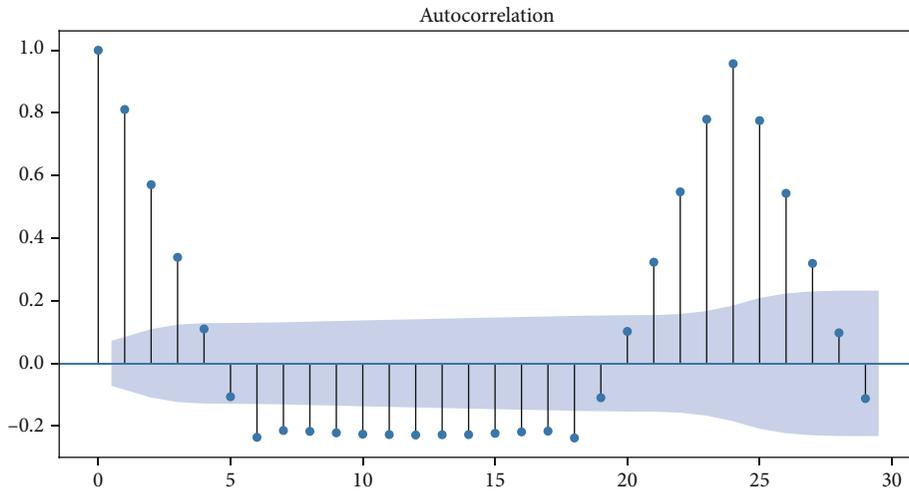


FIGURE 2: The autocorrelogram.

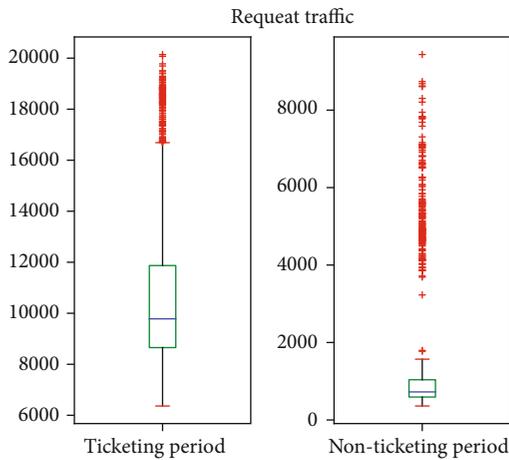


FIGURE 3: The boxplots of ticketing and nonticketing period data flows.

optimum. A differential evolution (DE) algorithm [20, 21] can be used to optimize the location update of the particle swarm with the optimal individual from the differential evolution swarm. In Equation (1), p^t_{gd} uses the optimal value among differential individuals and particle swarm. Sharing the global optimum of two swarms can accelerate optimization velocity of the particle swarm, reduce the risk of falling into local optimum, and output the optimal parameter combination.

3.3. *Differential Evolution.* The DE algorithm is a parallel direct search method utilizing NP d-dimensional parameter vectors for optimization [42]. It is a simple yet effective technique based on group-random search, designed to solve the global optimization problem. Initially, the search vector population is randomly chosen, and it should cover the whole parameter space. New parameters are generated through the mutation operation that adds the differential weights of two parameter vectors to a third vector called a mutated vector. Mutated vector parameters “crossover” with those in a vector called target vector. The resulting vector is called a trial vector. Suppose the value of

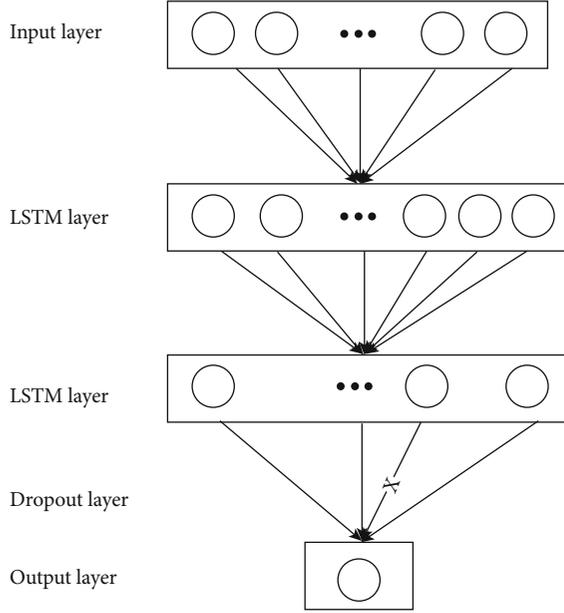


FIGURE 4: The fundamental network structure.

the trial vector's cost function is lower than the target vector. In that case, the trial vector replaces the target vector in the next generation. This process of replacing the target vector with the trial vector after comparing the cost function's value is called selection. The evolution of DE generates new descendants from the parental parameter vectors through mutation, crossover, and selection operation. In our design, the steps of DE evolution are as follows.

- (1) *Swarm Initialization.* The initial swarm consists of a vector $P_i(t) = [P_{i,1}(t), P_{i,2}(t), \dots, P_{i,d}(t)]$ that consists of N parameters, which are randomly selected from the overall search space. d is the dimension of individual vector; $i \in \{1, 2, \dots, N\}$ present the i^{th} chromosome.
- (2) *Mutation Operation.* Different strategies can be used for mutation operation. In order to share current optimal parameter combination with the particle swarm and accelerate the optimization velocity, Equation (2) represents an individual mutation operation.

$$v_i(t+1) = P_{\text{best}}(t) + F^*(P_{r_1}(t) - P_{r_2}(t)), \quad (2)$$

where i is the serial number of the current swarm; r_1 and r_2 are two unequal numbers randomly selected from $1 - N$ ($r_1 \neq r_2 \neq i$); $P_{\text{best}}(t)$ is the optimal individual in the t -generation particle swarm and differential swarm; F is the scaling factor.

- (3) *Crossover Operation.* For the trial vectors generated from mutated and target vectors, Equation (3) is used for crossover selection to get the trial vector.

$$u_{i,j}(t+1) = \begin{cases} v_{i,j}(t+1), & \text{if } \text{rand}(0, 1) < \text{CR} \text{ or } \text{rand}(j) = j, \\ P_{i,j}(t), & \text{otherwise,} \end{cases} \quad (3)$$

where CR is the crossover probability, $\text{rand}(0, 1)$ is a random number uniformly distributed within $[0, 1]$, and $\text{rand}(j)$ is a randomly selected dimension.

- (4) *Selection Operation.* The parameter vector with high fitness (low-cost function value) is selected using Equation (4). This helps to select the optimal parameters for the next-generation swarm.

$$P_i(t+1) = \begin{cases} u_i(t+1), & \text{if } f(u_i(t+1)) < f(P_i(t)), \\ P_i(t), & \text{otherwise.} \end{cases} \quad (4)$$

The mixed optimization is to select the optimal value for p_{gd}^t in Formula (1) and $P_{\text{best}}(t)$ in Formula (2) after iteration. The training here is to find the minimum loss value. Therefore, the optimal individual is selected with $\min(f(p_{gd}^t), f(P_{\text{best}}(t)))$ as the basis of particle swarm and differential swarm during the next-generation evolution.

4. Implementation Details

In this section, we present the implementation details for the proposed DP-LSTM model. The model is constructed using LSTM—optimizing it through a fusion of PSO and DE. We use this model for the access flow prediction of the “12306 ticketing system.” The purpose of proposing this model is to minimize the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the system. As mentioned in the previous sections, the flow data under consideration is time recurrent. We select several time points during ticketing and nonticketing periods. Figure 5 shows the flow trend. A cyclic flow variation can be noted with a significant difference in the access traffic flow depending on whether the ticketing is open or not. Therefore, we train different models for the ticketing period and nonticketing period. The overall structure of the network used for training this model is shown in Figure 4.

4.1. Input and Output. The real flow data for the past 24 hours is used as the model input. This is because the change period of the flow data is 24 hours. If i is the current hour, then the corresponding flow peak data within $[i - 24, i - 23, \dots, i - 3, i - 2, i - 1]$ hours is input, and the model will output the prediction for the peak traffic flow for the next hour. We also tried training the model using input comprising the flow data for more than 24 hours (multiple days). For example, Figure 6 shows some results generated by using traffic flow data for the past seven. We discovered that using long-term data for forecasting can better predict the trend of traffic changes. However, it introduces a serious lag and low prediction accuracy. Simultaneously, the increase in the number of days increases the input dimensions by seven

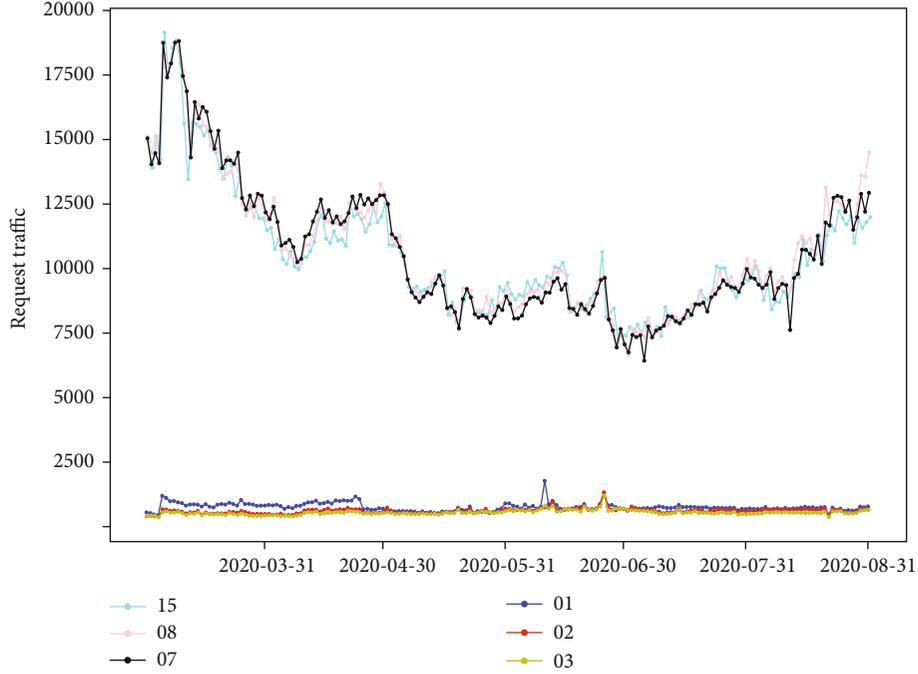


FIGURE 5: Service request flow variations at the same time, but on different dates.

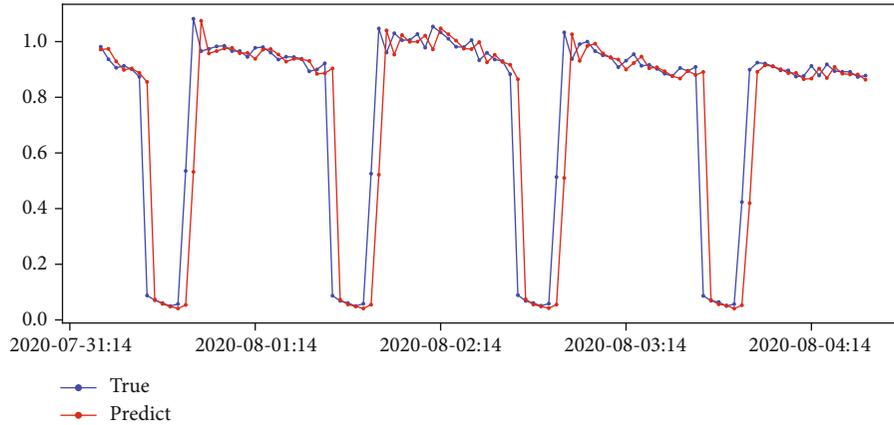


FIGURE 6: Model prediction results.

TABLE 1: Basic parameters.

Parameter	Description
NP_{PSO}	The number of particles
dim	Individual dimension
NP_{DE}	Population number
$iter_{max}$	Maximum number of iterations
w	Inertia weight of particles
CR	Crossover probability
c_1, c_2	Learning factors
F	Scaling factor

times. This significantly increased the time cost of model training. Therefore, selecting longer period data to predict the access traffic in the next hour is costly and does not generate optimal results.

4.2. Strategy Setting for Model Optimization

(a) Fitness function

The MAE between forecasted flow and real flow is considered as the fitness indicator. According to the experiment, the trained model's verified loss value has an inevitable fluctuation but tends to decline as a whole. Therefore, the average value of the final three verified loss values during training is used as a fitness measure.

(b) Boundary strategy

TABLE 2: Fitness of different models.

Iterations	PSO-LSTM (N)	DP-LSTM (N)
0	0.0117	0.0116
1	0.0113	0.0116
2	0.0109	0.0113
3	0.0108	0.0112
4	0.0108	0.0108
5	0.0106	0.0107
6	0.0106	0.0104
7	0.0106	0.0104
8	0.0106	0.0104
9	0.0106	0.0103
10	0.0106	0.0103

TABLE 3: Evaluation index results.

Model	MAE	MAPE	RMSE
LightGBM (N)	0.0193	8.5326	0.0480
RN-LSTM (N)	0.0191	10.9957	0.0391
PSO-LSTM (N)	0.0122	7.4350	0.0277
DP-LSTM (N)	0.0116	6.9252	0.0272

The boundary treatment is performed at each dimension of an individual observation, and maximum and minimum limits are set in each dimension. If an individual variation exceeds maximum or minimum, corresponding treatment is required. Equation (5) shows the treatment function.

$$\begin{cases} x_j = x_j^H & (\text{if } x_j > x_j^H, 0 \leq j \leq n), \\ x_j = x_j^L & (\text{if } x_j < x_j^L, 0 \leq j \leq n), \end{cases} \quad (5)$$

where j is the dimension of an individual variation, x_j^H , x_j^L is the maximum and minimum value of j , respectively, and n is the number of dimensions.

4.3. Hybrid Optimization Algorithm Flow. We fuse the evolution processes of PSO and DE to optimize the parameters of the model. This fusion helps identify the flow prediction model's optimal network structure during the ticketing and nonticketing period. The optimal parameters learned in this process are used for training to obtain the final flow prediction model. Underneath, we describe the detailed steps of algorithm flow.

Step 1: construct the sample sets for the nonticketing period and ticketing period, respectively. Generate the training sample set, validation sample set, and test sample set according to the ratio of 6:2:2. Then, perform model training for the nonticketing period flow prediction model DP-LSTM (N).

Step 2: initialize the basic PSO and DE parameters (as given in Table 1), and obtain the initial population. Generate a particle/individual randomly. Then, generate a correspond-

ing flow prediction model according to the value of the particle, and calculate the fitness value of each model is using the fitness function. Finally, repeat the above operation to generate NP_{DE} individuals and NP_{PSO} particles.

Step 3: update the speed and position of parameters with the number of NP_{PSO} and record the optimal fitness value experienced by each particle using Equation (1). Also, update the overall optimal fitness value and optimal particle position of the PSO. Furthermore, perform the mutation, hybridization, and selection operations using Equations (2)-(4) on all individuals in the DE population. And update the overall optimal fitness value and optimal individuals of the DE population.

Step 4: select the overall optimal set of DE and PSO output as the evolutionary basis for the next epoch. If the optimal fitness value of the DE output is less than the optimal fitness value of PSO, the optimal fitness value and particle position of PSO should be updated and vice versa. Otherwise, the optimal value and individual corresponding to DE should be updated.

Step 5: test whether the algorithm reaches the maximum number of iterations $iter_{max}$ or the loss value is less than los_{min} . If yes, obtain the optimal parameter combination and execute step 6; if not, update the iteration counter $t = t + 1$ and execute step 3.

Step 6: perform training according to the optimal parameter combination to obtain the optimal nonticketing period flow prediction model DP-LSTM (N).

Step 7: use the flow sample data set during the ticketing period to repeat step 2-step 6, obtain the optimal parameter combination model DP-LSTM (S) of the ticketing period, and combine the two optimal models to be the final flow prediction model DP-LSTM.

5. Experimental Evaluation

In this section, we provide the details of the experimental evaluation of the DP-LSTM model constructed to predict the flow data in the "12306 ticketing system" for warning and minimizing the impact of sudden fluctuations in online traffic for improving the dependability and overall Quality of Service (QoS) of the system. We describe the experimental setup's details. We explain the data preprocessing procedures, measure the effect of fused optimization used in our design, compare the proposed model's performance with mainstream methods, and present the results of error analysis.

5.1. Experimental Setup. The proposed model was implemented using Python (3.7.3) and the TensorFlow2 framework. The experiments were performed on a 40-core machine with 125 Giga byte memory. Operating system is Red Hat 7.4. Python API Matplotlib (3.0.3) was used for generating plots.

5.2. Data Set. We collected real data from the "12306 ticketing system" for this research. The peak flow per hour from 1 March to 31 August 2020 (i.e., data of 4393 samples in total) was used to generate the sample set. The data set is a

TABLE 4: Important model parameters.

Model	Important parameters
LightGBM	Boosting type: GBDT, evaluation metric: RMSE, learning rate: 0.3, min child weight: 3.0, number of iterations: 152
XGBoost	Booster: GBTree, evaluation metric: RMSE, gamma: 0.55, max depth: 19, min child weight: 1.0, number of estimators: 26.0
Random forest	Bootstrap: true, max depth: 5, min sample leaf: 2, min sample split:2, number of estimators: 30
SVR	Kernel: RBF, C: 1, epsilon: 0.1
SARIMA	Autoregressive model (P), difference (d), moving average (q): (2, 0, 24)

TABLE 5: Evaluation index results.

Model	MAE	MAPE	RMSE
SARIMA	0.039	5.169	0.057
SVR	0.038	23.431	0.049
Random forest	0.029	4.376	0.047
XGBoost	0.030	7.970	0.043
LightGBM	0.024	4.329	0.035
DP-LSTM	0.021	3.483	0.033

time series data with a period of 24 hours. Through the analysis of “access flow data in 12306 ticketing system,” it can be seen that the collected traffic belongs to nonstationary time series, and the change of access traffic has great fluctuation due to the influence of holidays and business changes. The data set was divided as per the “training set : verification set : test set = 6 : 2 : 2.” The training set comprised the data collected from March to June 2020. The verification and test data set comprised the data collected in June and August 2020.

5.3. Performance Measures. We used mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE) to measure the model’s performance. MAPE provides an effective determination method for forecasting the accuracy rate, which is usually expressed with percentage. The smaller the value is, the better the effect will be, as shown in Equation (6). The index MAE directly provides an average deviation between model output and real data. The larger the error is, the larger this value will be, as shown in Equation (7). RMSE is the standard deviation of the forecasting value, as shown in Equation (8). The smaller the value is, the better the model performance will be. MAPE, MAE, and RMSE are widely used evaluation indexes for neural network-based models [50].

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right|, \quad (6)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (7)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (8)$$

where y_i is the real value and \hat{y}_i the predicted value.

5.4. Data Preprocessing. The internal covariate shift or change in the distribution of input variables used for training and testing the model obstructs a neural network-based model’s training due to nonlinearities. Applying normalization on input data can offer an easy starting condition for the training and reduce the overall training time [51]. It is particularly the case for data sets collected over an extended period of time. Normalizing input data also reduces the risk of overfitting [52]. We test the input data to apply a suitable normalization technique. The Kolmogorov–Smirnov test (K-S test) is conducted on the collected data with the output result (statistic = 0.156 and P value = $6.31e-94$), and the P value is far lower than 0.05. Therefore, the data cannot accommodate normal distribution. We cannot use the normalization based on mean variation for the sample data. Moreover, the maximum and minimum flows during different periods are different, and the training set’s extreme values cannot be used for future data. Therefore, the normalization of maximum and minimum does not apply to the training data. The data distribution shows that the data is distributed within the scope of $[0, 2] \times 1e7$. Therefore, normalization is conducted by dividing by $1e7$.

5.5. Impact of Hybrid Optimization. We built three models: RN-LSTM (N), PSO-LSTM (N), and DP-LSTM (N) for forecasting flow data during the nonticketing period to measure the impact of fusing PSO and DE. It is important to test that randomness does not play any role in the performance of a deep learning model [53, 54]. RN-LSTM (N) was constructed to serve this need. All RN-LSTM (N) parameters were set based randomly. The PSO-LSTM (N) uses the particle swarm optimization algorithm to select parameters for the LSTM model. The DP-LSTM (N) combines DE and PSO algorithms to select parameters for the LSTM model. The three models all use the network structure in Figure 4. The parameters to be optimized are the number of output units of LSTM in the first layer (out1) and the number of output units of LSTM in the second layer (out2), the number of training execution cycles (epoch), and the proportion of dropout (rate). The parameter sequence is [out1, out2, epoch, rate]. The boundary condition for PSO-LSTM (N) and DP-LSTM (N) is set to $L = [5, 5, 40, 0.01]$ and $R = [70, 30, 100, 0.4]$, respectively. Here, L and R refer to the left boundary and right boundary, respectively. The optimal parameter sequences determined by the RN-LSTM (N) model, the PSO-LSTM (N) model, and the DP-LSTM (N) model are [60,10,38,0.1], [41,14,81,0.02], and [70,11,97,0.01], respectively. The fitness values in each iteration of the PSO-LSTM (N) model and DP-LSTM (N) model are shown in Table 2. It can be seen

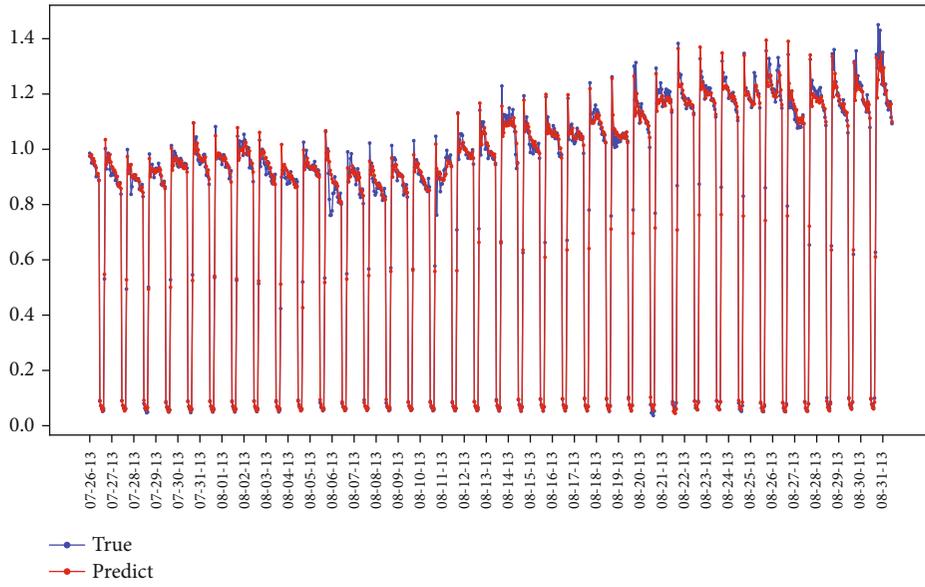


FIGURE 7: Overall forecasting effect.

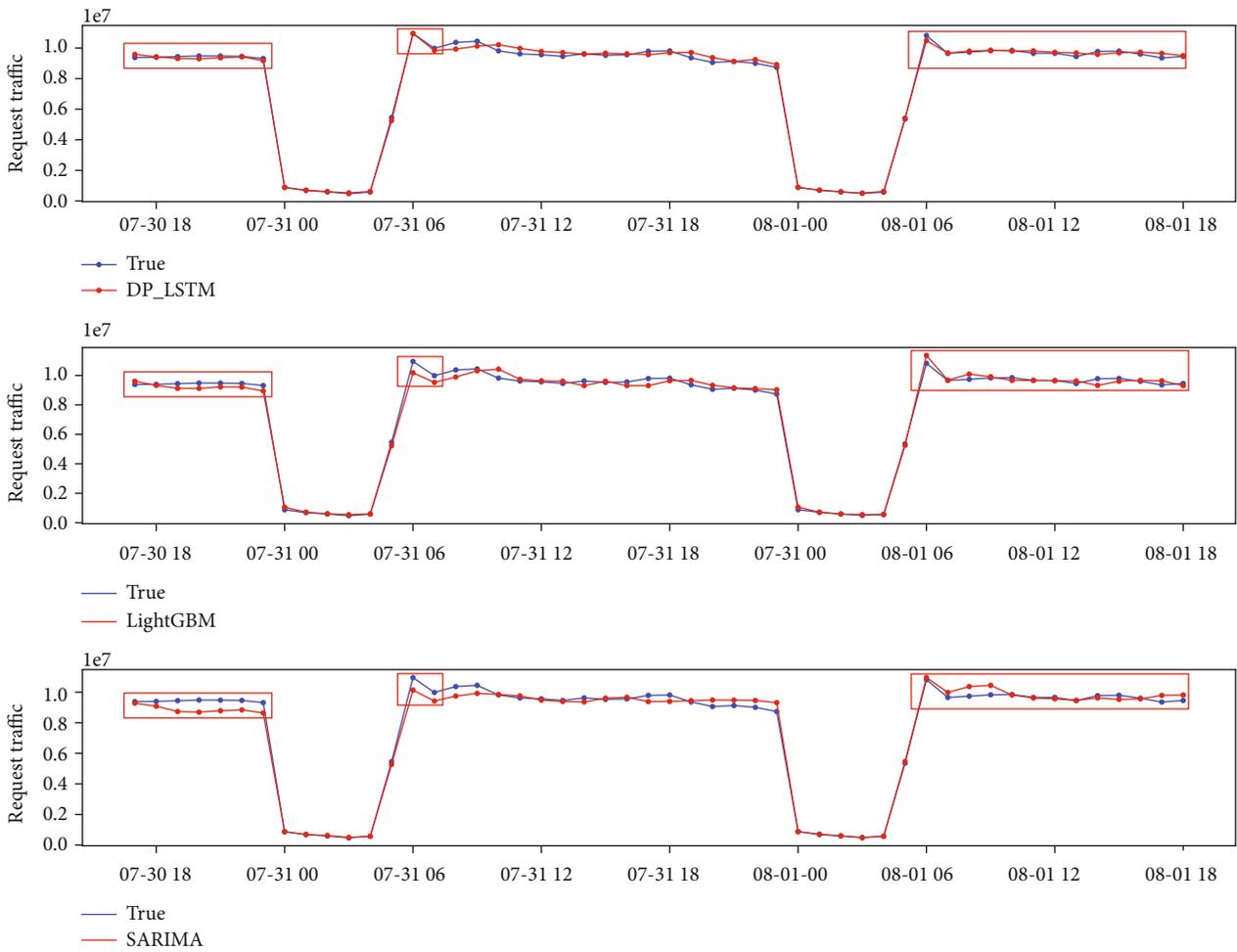


FIGURE 8: Prediction effect of different models.

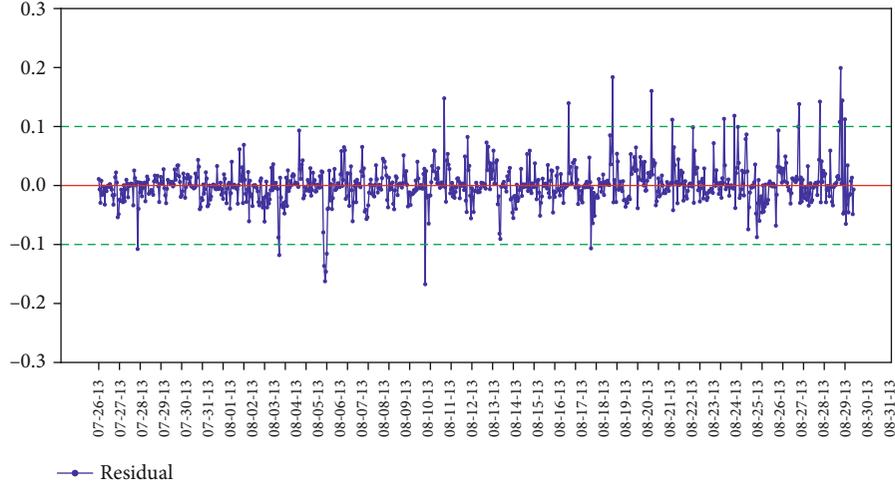


FIGURE 9: Residual sequence.

from the fifth iteration that PSO-LSTM (N) falls into the local optimum and cannot update the optimal value anymore. But PSO-LSTM (N) uses DE to guide the evolution of PSO, and the two populations develop and explore coordinately, which reduces the risk of falling into local optimum and speeds up the optimization.

To further compare parameter influence on the model, the optimal model LightGBM (N) was built using LightGBM for benchmarking. The test set was forecasted with RN-LSTM (N), PSO-LSTM (N), DP-LSTM (N), and LightGBM (N). The MAE, MAPE, and RMSE evaluation indexes were calculated, respectively, with the Formulas (5)–(7). Table 3 shows the results. It can be seen that MAE, MAPE, and RMSE of the DP-LSTM (N) model are lower than RN-LSTM (N) and PSO-LSTM (N). Therefore, it is possible to search optimal model parameters with mixed optimization based on DE and PSO. In contrast, the model with randomly selected parameters has worse effects for the same network structure, and some indexes are worse than LightGBM (N). Consequently, parameter optimization is crucial for the LSTM model results. The proposed mixed optimization method can achieve better effects.

5.6. Comparison with Mainstream Approach. We compare the DP-LSTM model with the mainstream time series models to verify the forecasting effect on the access flow of the “12306 ticketing system.” The Light Gradient Boosting Machine (LightGBM), XGBoost, Random forest, support vector regression (SVR), and Seasonal Autoregressive Integrated Moving Average (SARIMA) algorithm are used to build models. These methods have been widely used in recent studies [55–59]. The parameter optimization is conducted for each algorithm model for building the optimal models. For example, LightGBM and XGBoost use Bayesian optimization for parameter optimization, and SARIMA uses Akaike Information Criterion (AIC) for parameter selection. Table 4 presents related parameters for different models.

Each model makes predictions for the same test set, and then forecasting results are compared. Table 5 presents the evaluation results of each model for given indexes. The DP-LSTM generated the best value for the MAPE index. Contrarily, SVR and SARIMA have the worst values. Each index of LightGBM in the machine learning algorithm is approximate to DP-LSTM.

Figure 7 shows the overall forecasting effect of DP-LSTM. It can be seen that the model can achieve better fitting for data fluctuations.

The forecasts generated using the LightGBM, SARIMA, and DP-LSTM are shown in Figure 8. Reviewing the evaluation indexes in Table 5 and Figure 8, it can be observed that the prediction effect of the DP-LSTM is better than traditional algorithm. DP-LSTM needs to train the structure and weight parameters. The training time of DP-LSTM is about 3 hours, which is longer than that of LightGBM, but shorter than that of SARIMA. The training speed of DP-LSTM needs to be further optimized.

5.7. Residual Sequence Analysis. Residual sequence analysis (error analysis) plays a crucial role in time series prediction and analysis. If a time series is a white noise, there is a zero correlation among this series’ values. Prediction models do not work well on such series. Contrarily, while forecasting a time series, ideally, the series comprising a model’s forecasting errors should be white noise. Suppose the forecast errors constitute a series that is not white noise. In that case, the predictive model can be further optimized. We get residual sequence by subtracting the forecasted value of DP-LSTM from the test set’s real value. We use the white noise test function of Python for testing. The residual is not a white noise sequence because the P value is lower than 0.05 when the lag is between one and forty. Although the accuracy of the model training is high, there is extractable information that shall be further optimized to improve the forecasting accuracy. Figure 9 shows the distribution of the residual sequence. Although there is related information in the residual sequence, the error changes within 0.1. Therefore, the error

must be considered when the model is applied for decision-making.

6. Conclusion

The 5G communication technologies and their AI-embedded infrastructures enable the design and deployment of sophisticated network-based services. However, the hosts deploying these services can experience dependability and QoS issues. This paper constructed a deep learning-based model DP-LSTM using LSTM and hybrid optimization to address such problems in the “12306 ticketing system.” This system serves hundreds of millions of railway passengers daily and can experience network flow fluctuations due to demand variation. The proposed model forecasts the network flow data peak for the next hour based on the recent day’s access data. The performance of the LSTM network structure is optimized through a fusion of DE and PSO optimization algorithms. We used MAPE, MAE, and RMSE to evaluate the proposed model’s performance using real data experimentally. A comparison with the mainstream time series forecasting algorithms demonstrated the superiority of the proposed model. However, error analysis/residual sequence analysis showed that the proposed model could be further optimized. The proposed system can help for resource planning of the “12306 ticketing system,” thereby improving its dependability and QoS. Such solutions can also reduce the overall costs, particularly in cloud-based environments driven by pay-per-use model.

Data Availability

Data can be obtained by contacting the first author (fan.chun.mei@163.com).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported in part by the Chinese Academy of Railway Sciences under Grant Number 2019YJ122 and in part by the National Key R&D Program of China under Grant Number 2019YFF0301400.

References

- [1] R. Li, Z. Zhao, X. Zhou et al., “Intelligent 5G: when cellular networks meet artificial intelligence,” *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 2017.
- [2] X. Zhou, X. Xu, W. Liang et al., “Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.
- [3] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, “Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.
- [4] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, “Blockchain empowered arbitrable data auditing scheme for network storage as a service,” *IEEE Transactions on Services Computing*, vol. 13, pp. 1–1, 2019.
- [5] C. N. Tadros, M. R. M. Rizk, and B. M. Mokhtar, “Software defined network-based management for enhanced 5G network services,” *IEEE Access*, vol. 8, pp. 53997–54008, 2020.
- [6] M. Shariat, Ö. Bulakci, A. de Domenico et al., “A flexible network architecture for 5G systems,” *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5264012, 19 pages, 2019.
- [7] S. Din, A. Paul, and A. Rehman, “5G-enabled hierarchical architecture for software-defined intelligent transportation system,” *Computer Networks*, vol. 150, pp. 81–89, 2019.
- [8] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational LSTM enhanced anomaly detection for industrial big data,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.
- [9] Y. Bai, P. Hao, and Y. Zhang, “A case for web service bandwidth reduction on mobile devices with edge-hosted personal services,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 657–665, Honolulu, HI, USA, 2018-April.
- [10] S. Wang, H. Elahi, Y. Hu, Y. Zhang, and J. Wang, “A botnets control strategy based on variable forgetting rate of control commands,” *Concurrency and Computation: Practice and Experience*, pp. 1–21, 2020.
- [11] K. Wang, H. Li, Y. Feng, and G. Tian, “Big data analytics for system stability evaluation strategy in the energy internet,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1969–1978, 2017.
- [12] K. S. Kim, D. K. Kim, C. B. Chae et al., “Ultrareliable and low-latency communication techniques for tactile internet services,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 376–393, 2019.
- [13] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2020.
- [14] Z. Cai, Z. He, X. Guan, and Y. Li, “Collective data-sanitization for preventing sensitive information inference attacks in social networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 1–1, 2016.
- [15] J. Chen and J. Chen, “Stability analysis and parameters optimization of islanded microgrid with both ideal and dynamic constant power loads,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3263–3274, 2018.
- [16] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, “A blockchain-based multi-cloud storage data auditing scheme to locate faults,” *IEEE Transactions on Cloud Computing*, pp. 1–1, 2021.
- [17] F. Liu, Z. Sun, P. Zhang, Q. Peng, and Q. Qiao, “Analyzing capacity utilization and travel patterns of Chinese high-speed trains: an exploratory data mining approach,” *Journal of Advanced Transportation*, vol. 2018, Article ID 3985302, 9 pages, 2018.
- [18] D. D. Clark, C. Partridge, J. Christopher Ramming, and J. T. Wroclawski, “A knowledge plane for the internet,” *Computer Communication Review*, vol. 33, no. 4, pp. 3–10, 2003.
- [19] A. Mestres, A. Rodriguez-Natal, J. Carner et al., “Knowledge-defined networking,” *Computer Communication Review*, vol. 47, no. 3, pp. 2–10, 2017.

- [20] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.
- [21] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2020.
- [22] J. Li and J. Wang, "Short term traffic flow prediction based on deep learning," in *CICTP 2019*, pp. 2457–2469, American Society of Civil Engineers, Reston, VA, 2019.
- [23] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, no. 2019, pp. 2091–2100, 2020.
- [24] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.
- [25] B. Krollner, B. Vanstone, and G. Finnie, "Financial time series forecasting with machine learning techniques: a survey," in *Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN 2010): Computational Intelligence and Machine Learning*, pp. 25–30, 2010.
- [26] G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pp. 1–8, Kovilpatti, India, 2016, IEEE.
- [27] 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, pp. 1–9, 2015.
- [28] K. Lin, Q. Lin, C. Zhou, and J. Yao, "Time series prediction based on linear regression and SVR," in *Third International Conference on Natural Computation (ICNC 2007)*, pp. 688–691, Haikou, China, 2007.
- [29] Y. Song, *Stock Trend Prediction: Based on Machine Learning Methods*, UCLA, 2018, <https://escholarship.org/uc/item/0cp1x8th>.
- [30] S. Cerna, C. Guyeux, H. H. Arcolezzi, R. Couturier, and G. Royer, "A Comparison of LSTM and XGBoost for Predicting Firemen Interventions," in *Trends and Innovations in Information Systems and Technologies*, WorldCIST 2020, Á. Rocha, H. Adeli, L. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds., Springer, Cham, 2020.
- [31] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, "Deep autoregressive networks," *31st International Conference on Machine Learning*, vol. 4, pp. 2991–3000, 2014.
- [32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [33] M. Ragupathy and X. S. Ma, "Long short term memory based total traffic prediction for container load balancing," *Technical Disclosure Commons*, p. 4, 2018.
- [34] M. Li, Y. Wang, Z. Wang, and H. Zheng, "A deep learning method based on an attention mechanism for wireless network traffic prediction," *Ad Hoc Networks*, vol. 107, p. 102258, 2020.
- [35] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.
- [36] M. Meiss, F. Menczer, and A. Vespignani, "On the lack of typical behavior in the global web traffic network," in *Proceedings of the 14th international conference on World Wide Web - WWW'05*, pp. 510–518, New York, New York, USA, 2005, ACM Press.
- [37] B. Song, C. Fan, Y. Wu, and J. Sun, "Data prediction for public events in professional domains based on improved RNN-LSTM," *Journal of Physics: Conference Series*, vol. 976, article 012007, 2018.
- [38] T. Dlamini and S. Vilakati, "LSTM-based traffic load balancing and resource allocation for an edge system," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8825396, 15 pages, 2020.
- [39] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019, IEEE.
- [40] S. Liu, G. Liao, and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2787–2790, Wuhan, China, 2018, IEEE.
- [41] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 5, 2020.
- [42] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [43] H. Fu, "A novel hybrid differential evolution and particle swarm optimization algorithm for binary CSPs," in *2012 International Conference on Computer Science and Electronics Engineering*, pp. 545–549, Hangzhou, China, 2012.
- [44] Y. Dodge, *The Concise Encyclopedia of Statistics*, Springer New York, New York, NY, 2008.
- [45] A. E. R. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Applied Soft Computing Journal*, vol. 73, pp. 969–991, 2018.
- [46] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers and Industrial Engineering*, vol. 143, no. March, article 106435, 2020.
- [47] K. Chen, "APSO-LSTM: an improved LSTM neural network model based on APSO algorithm," *Journal of Physics: Conference Series*, vol. 1651, article 012151, 2020.
- [48] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.
- [49] J. Kennedy, "Particle swarm optimization," *Studies in Computational Intelligence*, vol. 780, pp. 760–766, 2019.
- [50] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [51] J. Wang, S. Li, Z. An, X. Jiang, W. Qian, and S. Ji, "Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines," *Neurocomputing*, vol. 329, pp. 53–65, 2019.
- [52] M. Liu, W. Wu, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

- [53] X. Huang, G. C. Fox, S. Serebryakov, A. Mohan, P. Morkisz, and D. Dutta, "Benchmarking deep learning for time series: challenges and directions," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5679–5682, Los Angeles, CA, USA, 2019.
- [54] X. Zhou, W. Liang, K. I.-K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 246–257, 2019.
- [55] Y. Wang, S. Sun, X. Chen et al., "International Journal of Electrical Power and Energy Systems Short-term load forecasting of industrial customers based on SVM and XGBoost," *International Journal of Electrical Power & Energy Systems*, vol. 129, no. February, article 106830, 2021.
- [56] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.
- [57] C. Faloutsos, J. Gasthaus, T. Januschowski, and Y. Wang, "Forecasting big time series," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 2102–2105, 2018.
- [58] P. F. Pai, K. P. Lin, C. S. Lin, and P. T. Chang, "Time series forecasting by a seasonal support vector regression model," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4261–4265, 2010.
- [59] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "Blockchain-enabled accountability mechanism against information leakage in vertical industry services," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.