A GRU-based traffic situation prediction method in multi-domain software defined network

Wenwen Sun[1,2] and Shaopeng Guan[1]

[1] School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, Shandong, China
[2] School of Electronic and Information Engineering, Huaibei Institute of Technology, Huaibei, Anhui, China

ABSTRACT

With the continuous development and improvement of Software-Defined Networking (SDN), large-scale networks are divided into multiple domains. Each domain, which is managed by a controller, forms multi-domain SDN architecture. In multi-domain SDN, the dynamics and complexity are more significant, bringing great challenges to network management. Comprehensively and accurately predicting traffic situation in multi-domain SDN can better maintain network stability. In this article, we propose a traffic situation prediction method based on the gated recurrent unit (GRU) network in multi-domain SDN. We first analyzed the relevant factors that affect data traffic and control traffic and transformed them into a time series of actual situation values. Then, to enhance the prediction performance of GRU, we used the salp swarm algorithm to optimize the hyperparameters of GRU automatically. Finally, we adopted hyperparameter optimized GRU to achieve traffic situation prediction in multi-domain SDN. The experimental results indicate that the proposed method outperforms other traditional machine learning algorithms in terms of prediction accuracy.

INTRODUCTION

Software-Defined Networking (SDN) is a new network architecture, which decouples the data plane and the control plane, and controls the network in a software-defined way. Therefore, it is more flexible and intelligent than a traditional network (*Kreutz et al., 2014*). The controller is the core component of the control plane and controls the entire network. The switches in the data plane are responsible for data forwarding (*Xu, Wang & Xu, 2020*). At the beginning of SDN design, a single controller was used to manage the entire network. Due to the small network scale at that time, a single controller could effectively manage the entire network (*Hu et al., 2018*). With the network scale expansion and business demand growth, the inherent defects of a single controller have become increasingly prominent, such as limited processing capacity of the controller and single point of failure (*Wang et al., 2017*). Therefore, large-scale networks need to be reasonably

Corresponding author
Shaopeng Guan,
konexgsp@gmail.com

divided into multiple domains. Each domain, which is managed by a controller, forms a multi-domain SDN network architecture.

The multi-domain architecture improves the scalability and flexibility of the network. However, as a complex system, the state of the network is always dynamic and is affected by many factors, such as network attacks, disasters and emergencies. In addition, the network topology is complex, and frequent information exchanges and emerging new applications cause a surge in network traffic. In the multi-domain architecture, the dynamics and complexity are more significant. Therefore, the management of multi-domain SDN network faces huge challenges.

Network traffic situation prediction, based on the network traffic data within a period of time (*i.e.,* the situation values obtained by situation assessment), uses expert knowledge and theoretical methods such as data mining to predict the future network status. It can enable network managers to grasp the next development trend, provide an important basis for network resource allocation in time, and avoid network congestion to a certain extent (*Roman, Lopez & Gritzalis, 2008*; *Arendt et al., 2016*). Therefore, traffic situation prediction has become an important part of network management (*Xie et al., 2018*).

In traditional networks, the existing traffic situation prediction methods can be divided into two categories: linear methods and nonlinear methods. Among them, the linear methods mainly include auto regression (AR) (*Lu et al., 2016*), moving average (MA) (*Feng & Shu, 2005*), autoregressive moving average (ARMA) (*Xu & Zeng, 2014*), *etc*. As the network architecture becomes more complex, the characteristics of the time series data in the network exceed the comprehensible range of the Poisson distribution and Markov distribution model. Furthermore, the linear model is only suitable for time series prediction with stable characteristics. For time-series with non-stationary and non-linear characteristics, the linear model cannot fully reflect the complexity change, which makes the prediction accuracy low. The non-linear methods mainly include support vector machine (SVM) (*Feng & Liu, 2011*), radial basis function neural network (RBFNN) (*Jiang et al., 2017*; *Wei, 2017*), general regression neural network (GRNN) (*Zhuo, Zhang & Gong, 2012*), Elman network (*Guoli, 2019*) and Back Propagation Neural Network (BPNN) (*Haviluddin & Alfred, 2014*), *etc*. Compared with the linear methods, the nonlinear methods improve prediction accuracy. However, there are some limitations. Although SVM requires a small number of samples, its prediction accuracy is greatly affected by the key parameters which are difficult to determine. RBFNN can approximate arbitrary nonlinear functions, but it has disadvantages such as being easily trapped in local optima and difficult to determine the network structure. Most of the neurons in GRNN cannot be activated, which has higher requirements for training samples. Elman network and BPNN have disadvantages such as over-fitting and poor stability. In addition, the above-mentioned traditional network traffic situation prediction methods still have the following shortcomings: network traffic is a typical time series and has the characteristics of non-linearity, non-stationarity and time-dependence. These prediction methods are limited by the internal structure of the model and cannot well capture the long-term dependence in the time series.

At present, there are few traffic situation prediction methods in SDN (*Hua et al., 2018*). In *Liu et al. (2021)*, a traffic situation prediction method based on RBFNN was proposed, which uses the SDN controller to collect network flow statistics, and uses RBFNN to analyze the nonlinear relationship of the network flow statistics to realize the traffic situation prediction. In *Wang et al. (2021)*, a traffic situation prediction method based on ARMA was proposed. This method obtains network traffic information through the forwarding rules of the data, and combines ARMA to realize the traffic situation prediction. However, the above-mentioned method has the shortcomings of not fully considering the nonlinearity, non-stationarity, and long-term dependence of the network traffic. Therefore, the existing traffic situation prediction methods are no longer suitable for multi-domain SDN.

The gated recurrent unit (GRU) network is a typical deep learning model. Since the introduction of GRU, it has been widely used in the fields such as speech (*Yuan, Tian & Lu, 2018*), image (*Mou, Ghamisi & Zhu, 2017*), and natural language processing (*Morchid, 2018*). It has a great performance in solving complex and non-linear forecasting problems (*Huang, Li & Huang, 2020*), such as urban traffic flow prediction (*Sun, Boukerche & Tao, 2020*), energy consumption prediction (*Kim & Cho, 2019*), and urban rainfall prediction (*Haidar & Verma, 2018*). We propose GRU for traffic situation prediction in multi-domain SDN. The multi-domain SDN is dynamic and depends on many factors. If all factors are used for situation prediction, irrelevant and redundant information will greatly increase the computational complexity. In addition, the hyperparameters of the traditional GRU network are manually set, so that it consumes significant time and becomes difficult to select the most suitable hyperparameters. Based on these, we analyze the relevant factors affecting data traffic and control traffic; and use the salp swarm algorithm (SSA) to automatically optimize the hyperparameters of the GRU. Therefore, our contributions in this work can be summarized as follows:

1. We analyze the relevant factors that affect data traffic and control traffic, and transform them into a time series of actual situation values.
2. Manually setting hyperparameters of CNN and GRU has the disadvantage of consuming time and it is difficult to select the most suitable hyperparameters. The SSA is used to automatically optimize the hyperparameters of the GRU.
3. The traffic situation values in multi-domain SDN have more significant characteristics such as non-linearity, non-stationarity, and long-term dependence. Hyperparameters optimized GRU is used to predict the traffic situation values.

The rest of the article is organized as follows. The "Traffic situation factors" section analyzes the traffic situation factor. The "GRU hyperparameter optimization" section introduces the GRU hyperparameter optimization method. The "Multi-domain SDN traffic situation prediction" section explains multi-domain SDN traffic situation prediction method. The "Experiment and analysis" section simulates and analyzes the experimental results. Finally, the conclusions are given in the "Conclusions".

TRAFFIC SITUATION FACTORS

Traffic is an important parameter to measure network operating load and status. To accurately predict the network status in the complex and changeable multi-domain SDN, various factors affecting traffic in it must fully be considered. In multi-domain SDN, there are two types of network traffic, including data traffic and control traffic (*Shu et al., 2016*). We analyze the factors that affect data traffic and control traffic, as well as extract them as situation factors. When the network equipment fails, broadcast packets are sent, which paralyzes network communication. Therefore, the values of the traffic situation factors, such as the average number of switch bytes and the number of lost packets, are calculated under the assumption that the network equipment does not fail.

Data traffic factors

Data traffic refers to the traffic transmitted between switches. It exists in the switches and their communication links. Switch and link performance play a vital role in the transmission of data traffic (*Tsai et al., 2018*). Therefore, we extract factors that affect switch performance such as the average number of switch bytes and the ratio of the number of switch bytes to the number of switch packets (*Sood, Yu & Xiang, 2016*), as well as extract factors that affect link performance such as the number of the lost packets, the maximum transmission delay, and the average bandwidth utilization (*Hu et al., 2020*), as data traffic factors.

(1) The average number of switch bytes (ANB): the average number of bytes of all switches per unit time. It is defined as follows:

$$ANB = \frac{1}{TN} \sum_{i=1}^{N} bytes\_num_i \qquad (1)$$

In Eq. (1), T is the sampling period, $bytes\_num_i$ is the number of bytes of the ith switch, and N is the number of switches.

(2) The ratio of the number of switch bytes to the number of switch packets (RBP): the ratio of the number of bytes to the number of packets of all switches. It is defined as follows:

$$RBP = \frac{\sum_{i=1}^{N} bytes\_num_i}{\sum_{i=1}^{N} packets\_num_i} \qquad (2)$$

where $packets\_num_i$ is the number of packets of the ith switch.

(3) The number of lost packets (NLP): the number of lost and errored packets in the network per unit time. It is calculated as follows:

$$NLP = \frac{packet\_num_{loss}}{T} \qquad (3)$$

where $packet\_num_{loss}$ is the number of lost and errored packets, communicating between

switches. The number of lost packets can characterize the network's service quality. If the number of lost and errored packets is smaller, the service quality of the network will be better. When 1,000 packets are sent, the number of lost packets is less than 10, preferably 0.

(4) The maximum transmission delay (MTD): the maximum transmission delay between the sending port and the receiving port of all links in the sampling period. It is calculated as follows:

$$MTD = \max\{portdelay(i)\} \tag{4}$$

where $portdelay(i)$ represents the transmission delay from the sending port to the receiving port of the ith link. The maximum transmission delay can better characterize the transmission quality of the network. If the delay is very large at that time, it will mean that congestion has occurred on some links.

(5) The average bandwidth utilization (ABU): The average value of the link load on all links to the link bandwidth. It is calculated as follows:

$$ABU = \frac{1}{L} \sum_{i=1}^{L} \frac{bytes\_transmit_i}{bandwidth_i} \tag{5}$$

where $bandwidth_i$ represents the bandwidth of the ith link, $bytes\_transmit_i$ represents the number of bytes transmitted per unit time on the ith link, and L represents the total number of links.

Control traffic factors

Control traffic refers to the traffic transmitted between the controllers and the switches. Mainly through *Packet_in* and *Flow_mod* packets, the controllers communicate with the switches. Therefore, we extract the average numbers of *Packet_in* and *Flow_mod* packets as control traffic factors.

(1) The average number of *Packet_in* (ANP): The average number of *Packet_in* packets interacted by all switches and controllers per unit time. It is calculated as follows:

$$ANP = \frac{1}{TN} \sum_{i=1}^{N} Packet\_in\_num(i) \tag{6}$$

where $Packet\_in\_num(i)$ represents the number of *Packet_in* packets transmitted by the ith switch. When a large number of *Packet_in* packets appear, excessive occupation of resources of controller and switch causes network failures.

(2) The average number of *Flow_mod* (ANF): the average number of *Flow_mod* packets interacted by all switches and controllers per unit time. It is calculated as follows:

$$ANF = \frac{1}{TN} \sum_{i=1}^{N} Flow\_mod\_num(i) \tag{7}$$

where $Flow\_mod\_num(i)$ represents the number of *Flow_mod* packets transmitted by the ith switch. Similar to the above equation, when a large number of *Flow_mod*

Table 1 Multi-domain SDN network situation level.

Traffic situation value	Description
0~0.25	The network is in a state of low utilization, with less business flow
0.25~0.50	The network is in a high utilization state and it is in good condition
0.50~0.75	The network is busy and maybe congested
0.75~1.0	The network is at a higher risk and may face paralysis

packets appear, excessive occupation of controller and switch resources causes network failures.

This article refers to the quantitative calculation method of literature (*Hong, Fessler & Balzano, 2018*). The specific process is as follows: first, carry out the min-max normalization coding for the above situation factor values (*Cerda & Varoquaux, 2020*); then, according to the importance of each factor, assign different weights to the factors; finally, obtain the traffic situation values through weighted calculation of the normalized values of situation factors. The traffic situation values (TSV) are calculated as follows:

$$TSV = \omega_{ANB} * ANB_{normal} + \omega_{RBP} * RBP_{normal} + \omega_{NLP} * NLP_{normal}$$
$$+ \omega_{MTD} * MTD_{normal} + \omega_{ABU} * ABU_{normal} + \omega_{ANP} * ANP_{normal} + \omega_{ANF} * ANF_{normal} \tag{8}$$

where ANB_{normal}, RBP_{normal}, NLP_{normal}, MTD_{normal}, ABU_{normal}, ANP_{normal} and ANF_{normal} are the normalized values of situation factors; ω_{ANB}, ω_{RBP}, ω_{NLP}, ω_{MTD}, ω_{ABU}, ω_{ANP} and ω_{ANF} are the weights of the corresponding situation factors. Among them, the principal component analysis (PCA) is adopted to determine the weights (*Tao et al., 2019*). This method can learn the laws from historical data more scientifically and objectively, and reduce the interference of people's subjective opinions (*Jolliffe & Cadima, 2016*). Therefore, the obtained traffic situation values can more accurately represent the current network state.

Under different network operating states, the network traffic situation values correspond to different division results. Generally, it can be divided into four levels of network status according to expert's experience (*Bhatia et al., 2020*), as shown in Table 1.

GRU HYPERPARAMETER OPTIMIZATION

The gated recurrent unit (GRU) network is good at processing time series. It uses multiple processing layers to approximate arbitrary complex functions and captures long-term dependence of time series by using multiple gates. Multi-domain SDN network traffic situation values have the characteristics such as non-linearity, long-term dependence, and non-stationarity, which is very suitable for using the GRU network to predict traffic situation values. The hyperparameters have a greater impact on the prediction accuracy of the GRU network. The hyperparameters of the traditional GRU network are manually optimized, which has the disadvantage of time-consuming and being difficult to select the most suitable hyperparameters. To improve the prediction effect, we use the SSA to automatically optimize the GRU hyperparameters.
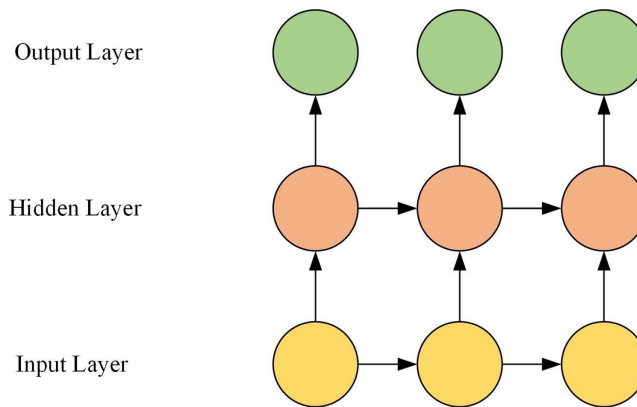
Figure 1 GRU network structure.

Table 2 Key GRU hyperparameters.

Hyperparameters	Type	Selection interval
The number of hidden layers	Discrete	Integer range [1,5]
The number of GRU neurons in each hidden layer	Discrete	Integer range [20,70]
Window size	Discrete	Integer range [2,10]

Key GRU hyperparameters

The GRU is composed of the input layer, the hidden layer, and the output layer. The hidden layer is composed of multiple GRU neurons. The GRU network structure is shown in Fig. 1.

The prediction effect of GRU mainly depends on the hyperparameters. The key hyperparameters of GRU are shown in Table 2.

1. The number of hidden layers and neurons in each hidden layer are important hyperparameters of GRU. If the number of hidden layers and GRU neurons in each hidden layer are larger, the network's expression ability and fitting ability will be stronger, and the prediction effect will be better. On the contrary, overfitting will be likely to occur.

2. The window size is to predict the value of the next moment based on how much historical data is present. If the window size is too small (large), it will not reflect true traffic situation, and the prediction effect will be poor.

GRU hyperparameters optimized by SSA

The Salp Swarm Algorithm (SSA) is a heuristic algorithm that simulates the foraging of salps in the biological world (*Mirjalili et al., 2017*). Compared with other heuristic algorithms such as the Particle Swarm Optimization (PSO) and the Firefly Algorithm (FA), it has various advantages such as high optimization accuracy, good robustness, and fast convergence speed. SSA divides the population into leaders and followers. The leader leads
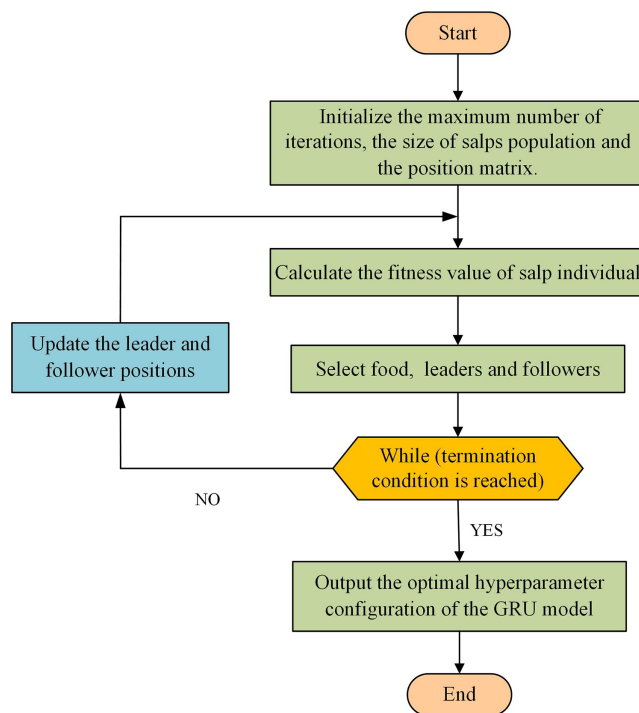
Figure 2 Flowchart of GRU hyperparameters optimized by SSA.

the followers, which forms the salp chain to perform population optimization. As the leader leads the followers, the entire population quickly and accurately converges to the optimal solution of the problem. To improve the prediction accuracy, we adopt SSA to automatically optimize the hyperparameters of the GRU network. The specific idea is as follows: the position of each salp individual represents a kind of hyperparameter configuration of the GRU network. Through the iterative optimization of salp individuals, the individual position with the best fitness value is the optimal hyperparameter configuration of the GRU network.

The specific steps are as follows, and the flowchart is shown in Fig. 2.

Step 1 Initialize the maximum number of iterations L, the size of the salp population N, and the position matrix $X_{N \times D}$.

Step 2 Calculate the fitness value of salp individuals in the population.

Step 3 Select the position of the food. The salp individuals in the population are sorted according to their fitness values, and the salp position with the best fitness value is regarded as the food position.

Step 4 Select the position of the leaders and followers. After selecting the food position, there are $N - 1$ salp individuals remaining in the population. The salps with the top half of the fitness value are regarded as the leaders and the rest as the followers.

Step 5 Update the leader and follower positions according to the literature (*Mirjalili et al., 2017*).
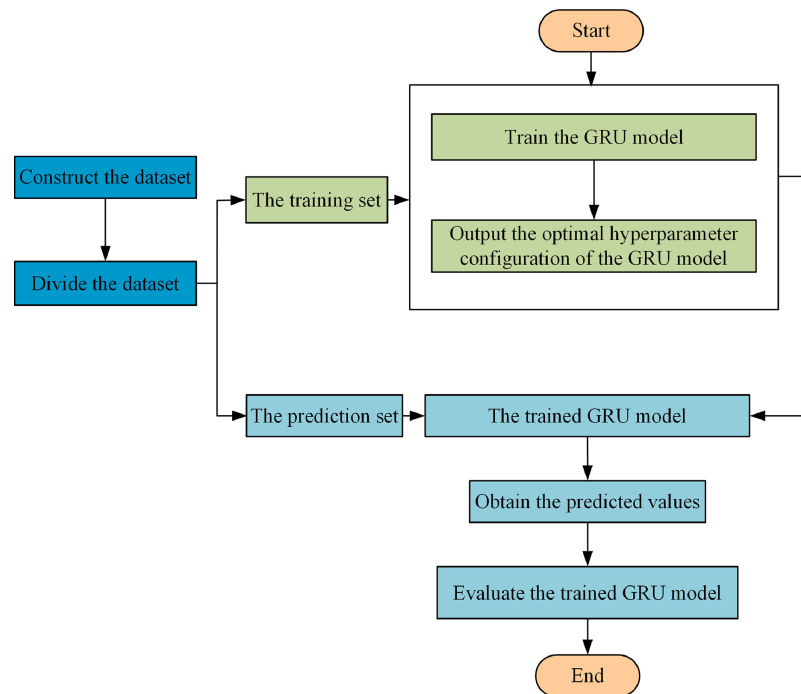
Figure 3 Flowchart of multi-domain SDN traffic situation prediction.

Table 3 Construction results of the dataset.

Sample number	Input data	Output data
1	$x_1, x_2, \cdots, x_\tau$	$x_{\tau+1}$
2	$x_2, x_3, \cdots, x_{\tau+1}$	$x_{\tau+2}$
...	...	...
$n - \tau$	$x_{n-\tau}, x_{n-\tau+1}, \cdots x_{n-1}$	x_n

Step 6 Determine whether the maximum number of iterations L is reached. If it is reached, the individual position with the best fitness value is taken as the optimal hyperparameter configuration of the GRU model. Otherwise, go to **Step 2**.

MULTI-DOMAIN SDN TRAFFIC SITUATION PREDICTION

In this paper, GRU optimized by SSA is applied to the traffic situation prediction in multi-domain SDN. The historical traffic situation values are used as the input of the model, to predict the traffic situation value at the next moment. The specific steps of the traffic situation prediction in multi-domain SDN are as follows, and the specific flowchart is shown in Fig. 3.

Step 1 Construct the dataset. According to the window size τ, the time series $X = (x_1, x_2, \cdots, x_n)$ of actual multi-domain SDN situation values is used to construct the dataset. The number of samples in the dataset is $n - \tau$. The construction results of the dataset are shown in Table 3.
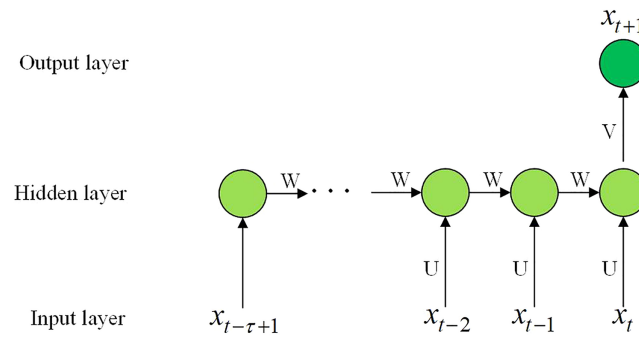
Figure 4 The GRU prediction model.

Step 2 Divide the dataset. The dataset in **Step 1** is divided into the training set and prediction set. The training set is used to train the GRU model, and the prediction set is used to verify the effectiveness of the trained model.

Step 3 Train the GRU model. The GRU prediction model is shown in Fig. 4.

1. Initialize the GRU hyperparameters: the number of hidden layers, the number of GRU neurons in each hidden layer, and the step size.
2. Initialize the salp population: the maximum number of iterations L, the size of the salp population N, and the position matrix $X_{N \times D}$.
3. According to the training set and fitness function, the fitness value of the salp individual is calculated. The positions of salp individuals are updated iteratively, and the salp individual position with the best fitness value is the optimal hyperparameter configuration of the GRU model. The root mean square error of the predicted values and the actual values is taken as the fitness function. It is calculated as follows (*Buche, Schraudolph & Koumoutsakos, 2005*):

$$fitness(i) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (y' - y)^2} \tag{9}$$

where m is the number of samples in the training set, y' is the output value of GRU, and y is the real value. When the fitness value of the individual salp position is the smallest, it is taken as the GRU model with the optimal hyperparameter.

Step 4 The predicted values are obtained by inputting the data in the prediction set into the trained GRU network. The predicted value is compared with the actual value, and the prediction accuracy of the model is measured by the evaluation index.

EXPERIMENT AND ANALYSIS

We used Mininet to construct the SDN network, where the controller uses the open-source Python-based controller Ryu. The deep learning module is developed based on the Keras framework, and the operating system environment is Ubuntu 16.04. A built multi-domain SDN platform, that can effectively simulate the real network environment, is
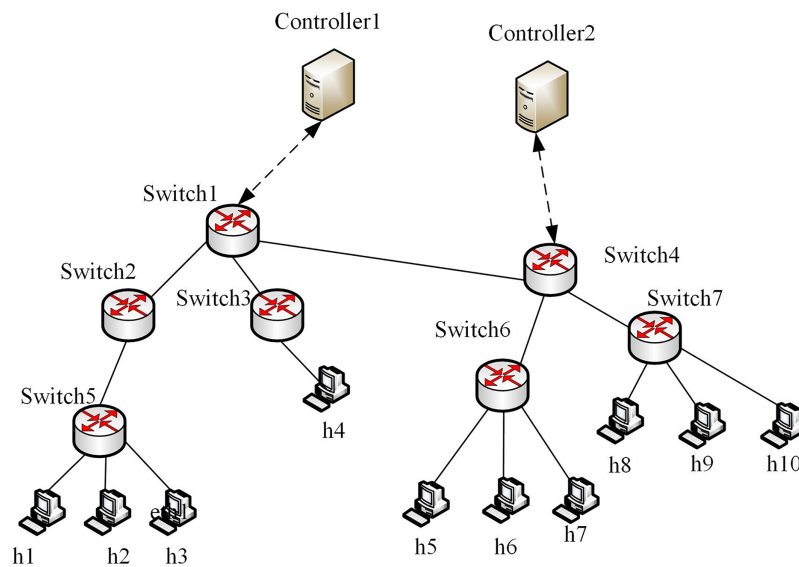
Figure 5 Multi-domain SDN network topology.

composed of the SDN_1 domain and SDN_2 domain. There are two controllers, seven switches, and 10 hosts. The experimental topology is shown in Fig. 5.

To demonstrate the effectiveness of the proposed method, we selected public traffic datasets MAWI2019 (*Blaise et al., 2020*) and Moore2003 (*Moore & Zuev, 2005*) as our experiment datasets. The two datasets contain raw traffic data, and the publication years, data terminals and IP addresses differ greatly. Thus, they can be used to effectively evaluate the generality of the proposed method. The traffic in MAWI and Moore datasets were injected into multi-domain SDN in Fig. 5, respectively, to simulate the traffic in the datasets. By extracting the situation factors in the "Traffic situation factors" section, the corresponding values are obtained. For the MAWI and Moore datasets, the changes of the situation factor values of multi-domain SDN over time are shown in Fig. 6.

It can be seen from Fig. 6 that the different situation factors related to traffic in the multi-domain SDN have quite different value ranges and changing trends. According to the literature (*Hong, Fessler & Balzano, 2018*), the traffic situation values on MAWI and Moore datasets are calculated. The traffic situation values on MAWI and Moore datasets are shown in Figs. 7 and 8.

It can be seen from Fig. 7 that traffic situation values are relatively high and there is more business traffic when the traffic sequence is between 100–290 on MAWI dataset. The network situation is in a stable state when the traffic sequence is between 0–100 or 290–350.

It can be seen from Fig. 8 that the traffic situation values are relatively high and there is more business traffic when the traffic sequence is between 100–200 or 270–350 on Moore dataset. The network situation is in a stable state when the traffic sequence is between 0–100 or 200–270.

To verify the effectiveness of the proposed method, we used the traffic situation values on MAWI and Moore to construct the datasets according to Step 1 of section "Multi-
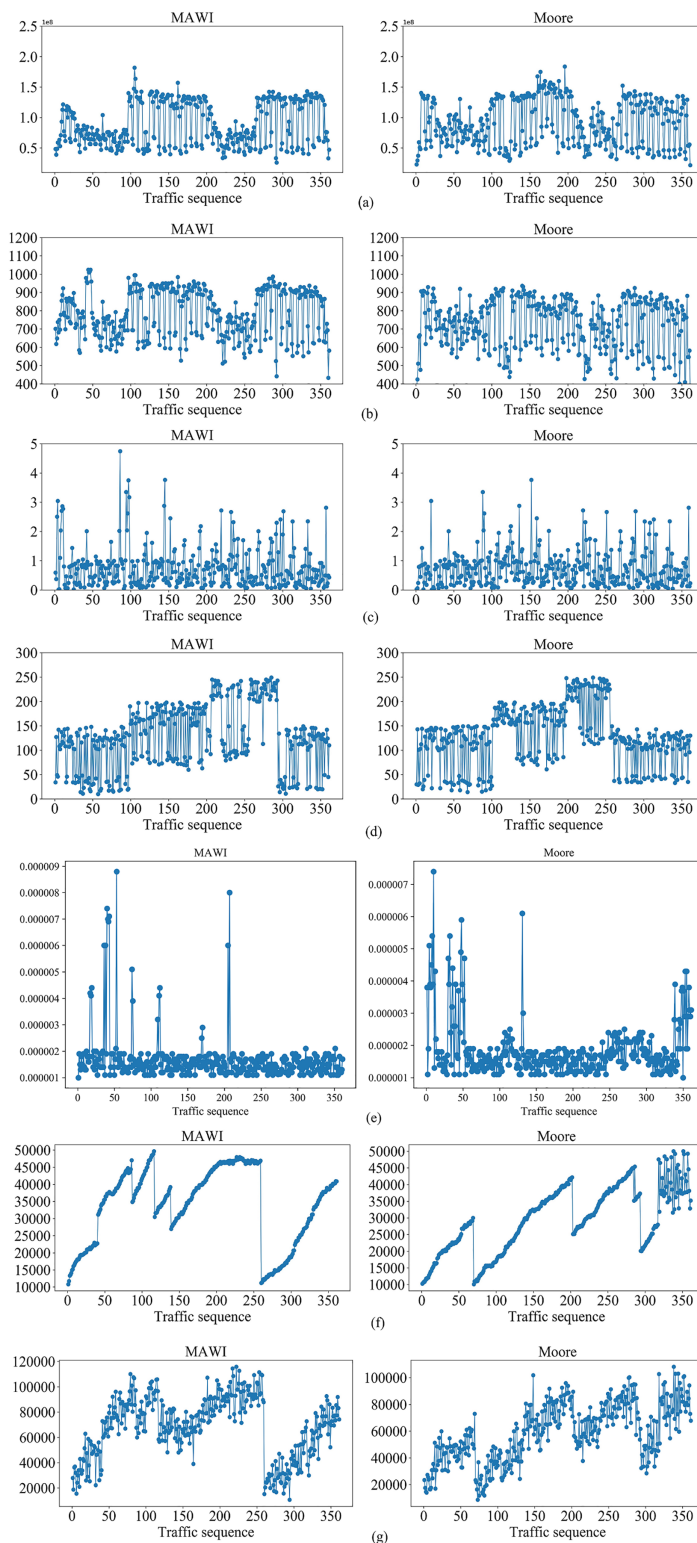
Figure 6 Changes of the situation factor values over time on MAWI and Moore datasets, respectively. (A) The average number of switch bytes (bps). (B) The ratio of the number of switch bytes to the number of switch packets. (C) The number of lost packets (s). (D) The maximum transmission delay (pps). (E) The average bandwidth utilization. (F) The average number of Packet_in (pps). (G) The average number of Flow_mod (pps).
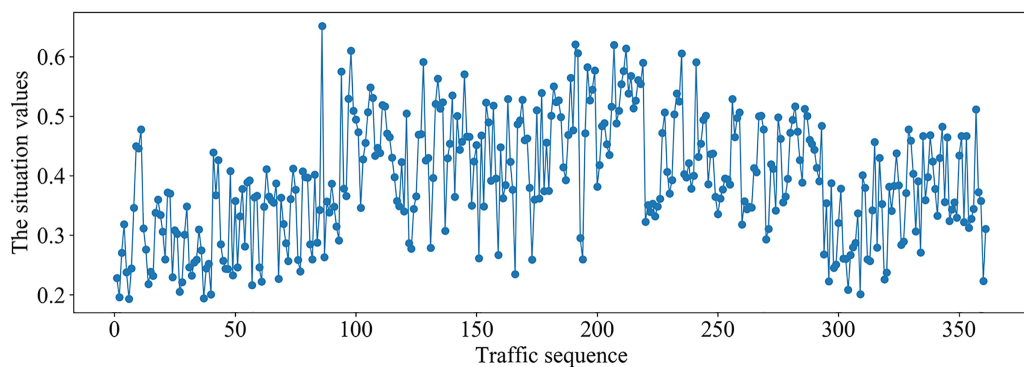
Figure 7 Traffic situation values on MAWI dataset.
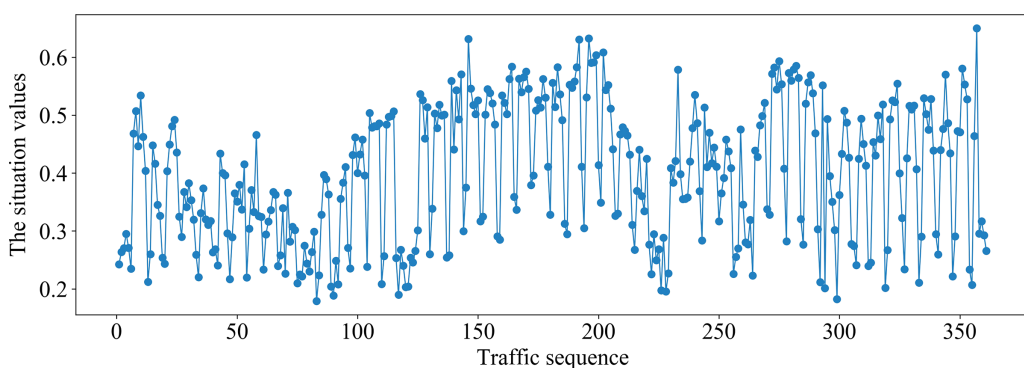
Figure 8 Traffic situation values on the Moore dataset.

domain SDN traffic situation prediction", respectively. The dataset constructed from the traffic situation values on MAWI is named as dataset_1. The dataset constructed from the traffic situation values on the Moore is named as dataset_2. To enable the model to be better trained and improve the model prediction accuracy, both dataset_1 and dataset_2 are divided into a training set and prediction set according to a ratio of 8:2. The training set is adopted to train the model, and the prediction set is used to detect the prediction effect of the trained model.

We utilized PSO, FA and SSA, respectively, to optimize the GRU hyperparameters on dataset_1 and dataset_2. Among them, the size of the population is set to 20, and the maximum number of iterations is set to 40. The changing curves of fitness values of the three algorithms in the iterative process are shown in Figs. 9 and 10, respectively.

It can be seen from Figs. 9 and 10 that, compared with PSO and FA, the SSA has better fitness values and converges faster during the iteration process.

We adopted LR, ARIMA, SVM, RF, DT, BP, RNN, and our scheme (*i.e.*, SSA-GRU), respectively, to forecast the traffic situation values on dataset_1 and dataset_2. The optimal GRU hyperparameters selected by SSA on dataset_1 and dataset_2 are shown in Tables 4 and 5, respectively. The prediction results of each algorithm on dataset_1 and dataset_2 are shown in Figs. 11 and 12, respectively.
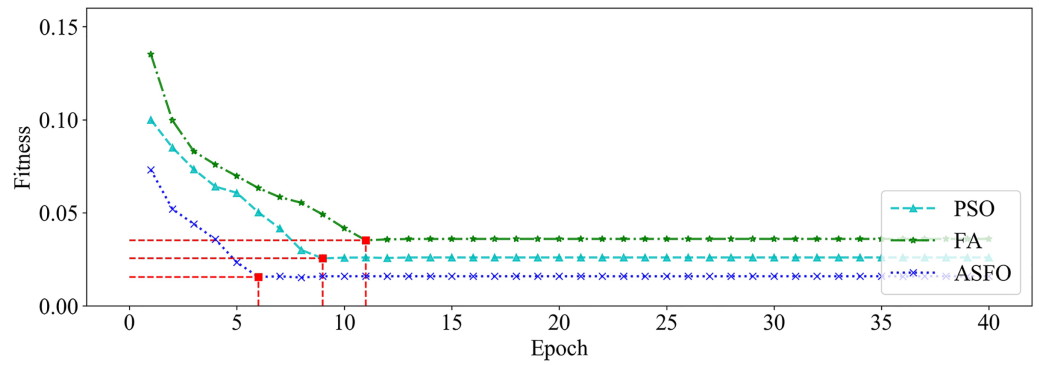
Figure 9 The changing curve of fitness value on dataset_1.
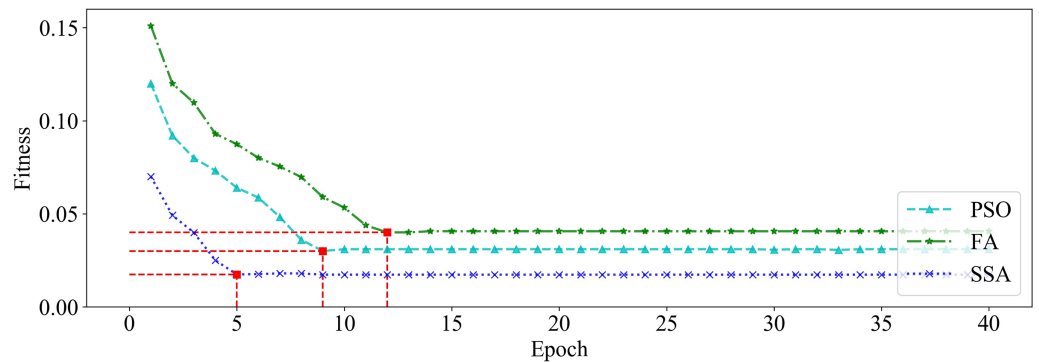
Figure 10 The changing curve of fitness value on dataset_2.

Table 4 Optimal GRU hyperparameters on the dataset_1.

Hyperparameter	Selection
Number of hidden layers	3
Number of GRU neurons in the first hidden layer	38
Number of GRU neurons in the second hidden layer	35
Number of GRU neurons in the third hidden layer	27
Window size	6

In addition, the root mean square error (RMSE), average absolute error (MAE), average relative error (MRE) (*Wang et al., 2018*) and mean absolute percentage error (MAPE) were used to evaluate the prediction performance of different algorithms. The formulas are defined as follows:

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y'_i - y_i)^2} \tag{10}$$
Table 5 Optimal GRU hyperparameters on the dataset_2.	
Hyperparameter	**Selection**
Number of hidden layers	3
Number of GRU neurons in the first hidden layer	40
Number of GRU neurons in the second hidden layer	33
Number of GRU neurons in the third hidden layer	28
Window size	6
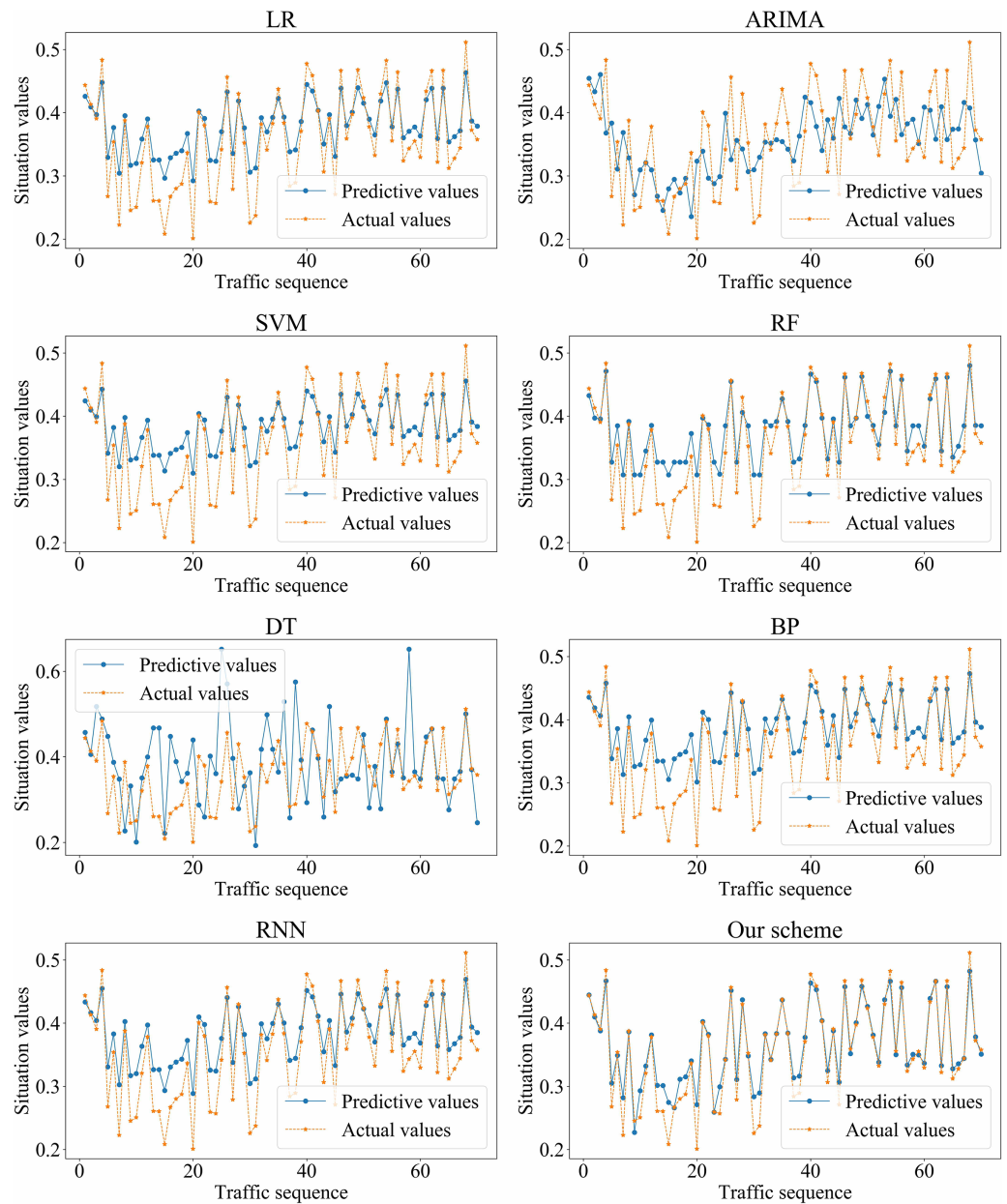
Figure 11 Prediction results of different algorithms on dataset_1.
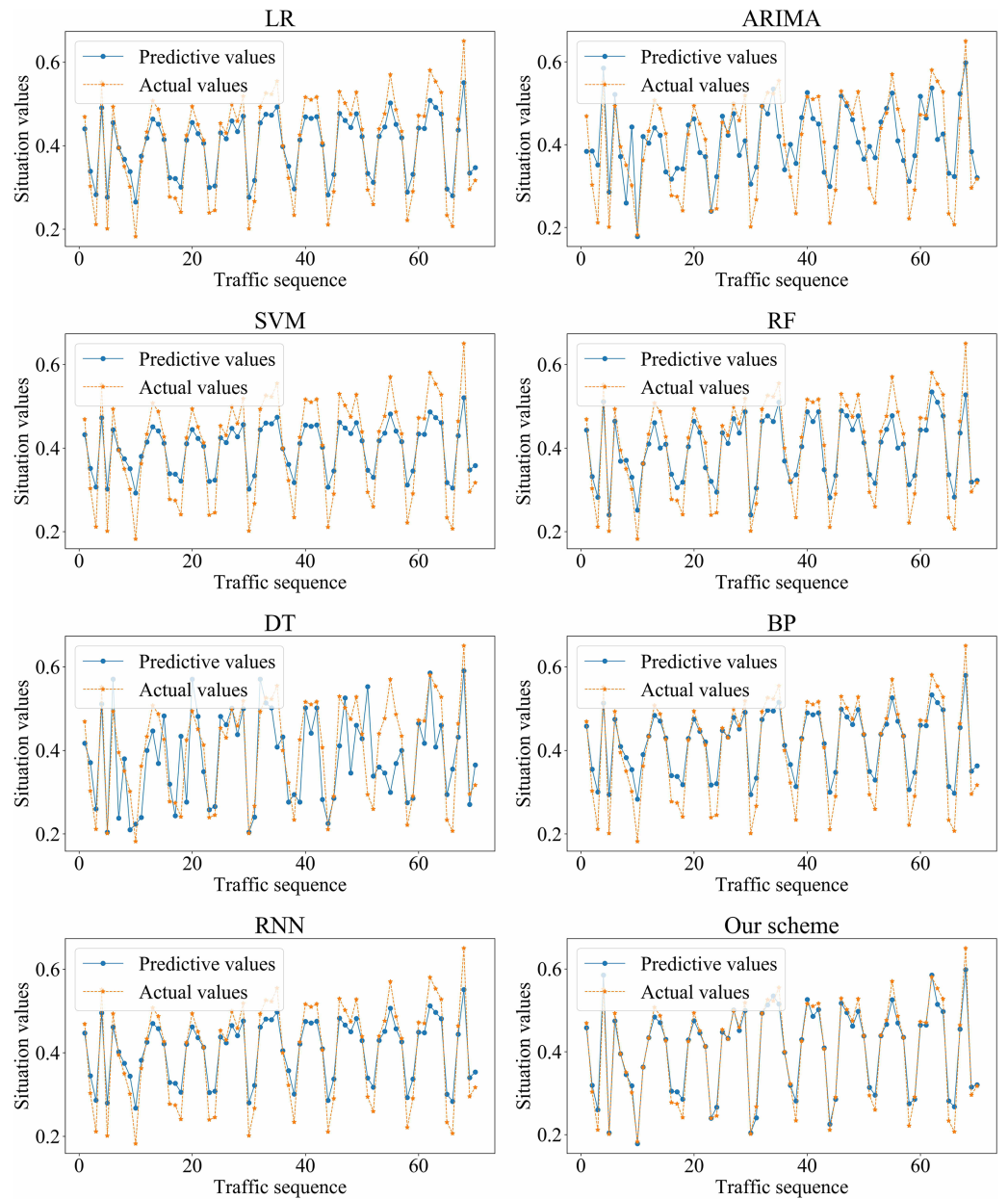
Figure 12 Prediction results of different algorithms on dataset_2.

$$MAE = \frac{1}{M} \sum_{i=1}^{M} \left| y'_i - y_i \right| \tag{11}$$

$$MRE = \frac{1}{M} \sum_{i=1}^{M} \frac{\left| y'_i - y_i \right|}{y_i} \tag{12}$$

$$MAPE = \frac{1}{M} \sum_{i=1}^{M} \frac{\left| y'_i - y_i \right|}{y_i} \times 100\% \tag{13}$$

Table 6 Prediction performance of different algorithms on dataset_1.

Algorithms	RMSE	MAE	MRE	MAPE/%
LR	0.0413	0.0341	0.1151	11.51%
ARIMA	0.0786	0.0665	0.2010	20.10%
SVM	0.0493	0.0408	0.1380	13.80%
RF	0.03885	0.0299	0.1048	10.48%
DT	0.1244	0.0992	0.3091	30.91%
BP	0.04676	0.0386	0.1322	13.22%
RNN	0.0422	0.0354	0.12	12%
SSA-GRU	0.0234	0.0155	0.0563	5.63%

Table 7 Prediction performance of different algorithms on dataset_2.

Algorithms	RMSE	MAE	MRE	MAPE/%
LR	0.0465	0.0413	0.1275	12.75%
ARIMA	0.0736	0.0624	0.1863	18.63%
SVM	0.0613	0.0542	0.1627	16.27%
RF	0.0498	0.0430	0.1271	12.71%
DT	0.087	0.0653	0.1742	17.42%
BP	0.0471	0.0372	0.1275	12.75%
RNN	0.0461	0.0397	0.1229	12.29%
SSA-GRU	0.0226	0.0162	0.0487	4.87%

where M is the number of samples in the dataset, y' is the output value of GRU, and y is the real value.

The prediction performance of different algorithms on dataset_1 and dataset_2 is shown in Tables 6 and 7, respectively.

Based on the above experiment results, the proposed method has better prediction accuracy and prediction effect. The prediction error is smaller. The predictive and actual values fit better.

CONCLUSIONS

To predict the multi-domain SDN traffic status comprehensively and accurately, we proposed a traffic situation prediction method based on the GRU network in multi-domain SDN. First, we analyzed the relevant factors that affect data traffic and control traffic, and convert them into a time series of actual situation values. Then, to overcome the shortcomings of manually optimizing the hyperparameters of GRU such as time-consuming and difficulty in selecting the most suitable hyperparameters, we used SSA to automatically optimize the hyperparameters of GRU, which improves the prediction effect. Finally, we used hyperparameter optimized GRU to predict the situation values. The results of multiple experiments on the traffic datasets MAWI and Moore show that the proposed method has higher prediction accuracy than traditional machine learning

algorithms. Limited by the experimental conditions, only two domains without network failures have been tested. In future work, further research will be conducted on three and more domains with network failures. To better verify the prediction effect of our proposed method, we will deploy the proposed method in a real environment and explore more application scenarios.

Funding
This work was supported by the Scientific Research Projects of Universities in Anhui Province (KJ2021A1243). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures
The following grant information was disclosed by the authors:
Scientific Research Projects of Universities in Anhui Province: KJ2021A1243.

Competing Interests
The authors declare that they have no competing interests.

Author Contributions
- Wenwen Sun conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Shaopeng Guan analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability
The Moore dataset is available at figshare: kuan, james (2022): Moore dataset. figshare. Dataset. https://doi.org/10.6084/m9.figshare.18467507.v1.
The NISM dataset is available at figshare: kuan, james (2022): NISM dataset.zip. figshare. Dataset. https://doi.org/10.6084/m9.figshare.18467504.v1.

Supplemental Information
Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.1011#supplemental-information.

REFERENCES

Arendt D, Best D, Burtner R, Paul CL. 2016. CyberPetri at CDX 2016: real-time network situation awareness. In: *2016 IEEE Symposium on Visualization for Cyber Security (VizSec)*. Piscataway: IEEE, 1–4.

Bhatia J, Dave R, Bhayani H, Tanwar S, Nayyar A. 2020. SDN-based real-time urban traffic analysis in VANET environment. *Computer Communications* **149(3)**:162–175 DOI 10.1016/j.comcom.2019.10.011.

Blaise A, Bouet M, Conan V, Secci S. 2020. Detection of zero-day attacks: an unsupervised port-based approach. *Computer Networks* **180(1)**:107391 DOI 10.1016/j.comnet.2020.107391.

Buche D, Schraudolph NN, Koumoutsakos P. 2005. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **35(2)**:183–194 DOI 10.1109/TSMCC.2004.841917.

Cerda P, Varoquaux G. 2020. Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering* **34(3)**:1164–1176 DOI 10.1109/TKDE.2020.2992529.

Feng H, Liu Y. 2011. Network traffic prediction based on wavelet analysis and AR-LSSVM. *Computer Engineering and Applications* **47(20)**:102–104 DOI 10.7763/IJFCC.2013.V2.130.

Feng H, Shu Y. 2005. Study on network traffic prediction techniques. In: *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing.* Piscataway: IEEE, 1041–1044.

Guoli W. 2019. Traffic prediction and attack detection approach based on PSO optimized Elman neural network. In: *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA).* Piscataway: IEEE, 504–508.

Haidar A, Verma B. 2018. A novel approach for optimizing climate features and network parameters in rainfall forecasting. *Soft Computing* **22(24)**:8119–8130 DOI 10.1007/s00500-017-2756-7.

Haviluddin H, Alfred R. 2014. Daily network traffic prediction based on backpropagation neural network. *Australian Journal of Basic and Applied Sciences* **8**:164–169 DOI 10.1097/BRS.0b013e31829ef4d7.

Hong D, Fessler JA, Balzano L. 2018. Optimally weighted PCA for high-dimensional heteroscedastic data. *Available at https://arxiv.org/abs/1810.12862.*

Hu T, Guo Z, Yi P, Baker T, Lan J. 2018. Multi-controller based software-defined networking: a survey. *IEEE Access* **6**:15980–15996 DOI 10.1109/ACCESS.2018.2814738.

Hu T, Yi P, Lan J, Hu Y, Sun P. 2020. FTLink: efficient and flexible link fault tolerance scheme for data plane in Software-Defined Networking. *Future Generation Computer Systems* **111(1)**:381–400 DOI 10.1016/j.future.2019.11.015.

Hua J, Zhao L, Zhang S, Liu Y, Ge X, Zhong S. 2018. Topology-preserving traffic engineering for hierarchical multi-domain SDN. *Computer Networks* **140(5)**:62–77 DOI 10.1016/j.comnet.2018.04.011.

Huang W, Li Y, Huang Y. 2020. Deep hybrid neural network and improved differential neuroevolution for chaotic time series prediction. *IEEE Access* **8**:159552–159565 DOI 10.1109/ACCESS.2020.3020801.

Jiang Y, Li C, Yu L, Cao B. 2017. On network security situation prediction based on RBF neural network. In: *2017 36th Chinese Control Conference (CCC).* Piscataway: IEEE, 4060–4063.

Jolliffe IT, Cadima J. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374(2065)**:20150202 DOI 10.1098/rsta.2015.0202.

Kim TY, Cho SB. 2019. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **182(5)**:72–81 DOI 10.1016/j.energy.2019.05.230.

Kreutz D, Ramos FM, Verissimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. 2014. Software-defined networking: a comprehensive survey. *Proceedings of the IEEE* **103(1)**:14–76 DOI 10.1109/JPROC.2014.2371999.

Liu C, Zhang G, Li B, Ma R, Jiang D, Zhao Y. 2021. A SDN-based intelligent prediction approach to power traffic identification and monitoring for smart network access. *Wireless Networks* **27(5)**:3665–3676 DOI 10.1007/s11276-019-02235-9.

Lu Z, Zhou C, Wu J, Jiang H, Cui S. 2016. Integrating granger causality and vector auto-regression for traffic prediction of large-scale WLANs. *KSII Transactions on Internet & Information Systems* **10(1)**:136–151 DOI 10.3837/tiis.2016.01.008.

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. 2017. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* **114**:163–191 DOI 10.1016/j.advengsoft.2017.07.002.

Moore AW, Zuev D. 2005. Internet traffic classification using Bayesian analysis techniques. In: *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems.* Association for Computing Machinery, 50–60.

Morchid M. 2018. Parsimonious memory unit for recurrent neural networks with application to natural language processing. *Neurocomputing* **314(2)**:48–64 DOI 10.1016/j.neucom.2018.05.081.

Mou L, Ghamisi P, Zhu XX. 2017. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* **55(7)**:3639–3655 DOI 10.1109/TGRS.2016.2636241.

Roman R, Lopez J, Gritzalis S. 2008. Situation awareness mechanisms for wireless sensor networks. *IEEE Communications Magazine* **46(4)**:102–107 DOI 10.1109/MCOM.2008.4481348.

Shu Z, Wan J, Lin J, Wang S, Li D, Rho S, Yang C. 2016. Traffic engineering in software-defined networking: measurement and management. *IEEE Access* **4**:3246–3256 DOI 10.1109/ACCESS.2016.2582748.

Sood K, Yu S, Xiang Y. 2016. Performance analysis of software-defined network switch using M/Geo/1 model. *IEEE Communications Letters* **20(12)**:2522–2525 DOI 10.1109/LCOMM.2016.2608894.

Sun P, Boukerche A, Tao Y. 2020. SSGRU: a novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Computer Communications* **160(22)**:502–511 DOI 10.1016/j.comcom.2020.06.028.

Tao Y, Shi H, Song B, Tan S. 2019. A novel dynamic weight principal component analysis method and hierarchical monitoring strategy for process fault detection and diagnosis. *IEEE Transactions on Industrial Electronics* **67(9)**:7994–8004 DOI 10.1109/TIE.2019.2942560.

Tsai PW, Tsai CW, Hsu CW, Yang CS. 2018. Network monitoring in software-defined networking: a review. *IEEE Systems Journal* **12(4)**:3958–3969 DOI 10.1109/JSYST.2018.2798060.

Wang Y, Jiang D, Huo L, Zhao Y. 2021. A new traffic prediction algorithm to software defined networking. *Mobile Networks and Applications* **26(2)**:716–725 DOI 10.1007/s11036-019-01423-3.

Wang R, Liang X, Zhu X, Xie Y. 2018. A feasibility of respiration prediction based on deep Bi-LSTM for real-time tumor tracking. *IEEE Access* **6**:51262–51268 DOI 10.1109/ACCESS.2018.2869780.

Wang G, Zhao Y, Huang J, Wang W. 2017. The controller placement problem in software defined networking: a survey. *IEEE Network* **31(5)**:21–27 DOI 10.1109/MNET.2017.1600182.

Wei D. 2017. Network traffic prediction based on RBF neural network optimized by improved gravitation search algorithm. *Neural Computing and Applications* **28(8)**:2303–2312 DOI 10.1007/s00521-016-2193-z.

Xie J, Yu FR, Huang T, Xie R, Liu J, Wang C, Liu Y. 2018. A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges. *IEEE Communications Surveys & Tutorials* **21(1)**:393–430 DOI 10.1109/COMST.2018.2866942.

Xu J, Wang L, Xu Z. 2020. An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Computer Networks* **169**:107092 DOI 10.1016/j.comnet.2019.107092.

Xu S, Zeng B. 2014. Network traffic prediction model based on auto-regressive moving average. *Journal of Networks* **9(3)**:653–659 DOI 10.4304/jnw.9.3.653-659.

Yuan Y, Tian C, Lu X. 2018. Auxiliary loss multimodal GRU model in audio-visual speech recognition. *IEEE Access* **6**:5573–5583 DOI 10.1109/ACCESS.2018.2796118.

Zhuo Y, Zhang Q, Gong Z. 2012. GRNN model of network situation forecast. *Journal of PLA University of Science and Technology (Natural Science Edition)* **13(4)**:147–151 DOI 10.3969/j.issn.1009-3443.2012.02.006.