IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 20, 2019, accepted December 5, 2019, date of publication December 13, 2019, date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959303

Long-Term Traffic Scheduling Based on Stacked Bidirectional Recurrent Neural Networks in Inter-Datacenter Optical Networks

AO YU[1], HUI YANG[1], TING XU[1], BAOGUO YU[2], QIUYAN YAO[1], YAJIE LI[1], TAO PENG[3], HUIFENG GUO[3], JUN LI[4], AND JIE ZHANG[1]

[1]State Key Laboratory of Information Photonics and Optical Communication, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]State Key Laboratory of Satellite Navigation System and Equipment Technology, Shijiazhuang 0500002, China
[3]ZTE Corporation, Shenzhen 518057, China
[4]504th Research Institutes, China Aerospace Science and Technology Corporation, Xi'an 710000, China

Corresponding author: Hui Yang (yang.hui.y@126.com)

This work was supported in part by the NSFC Project under Grant 61871056 and Grant 91638203, in part by the Young Elite Scientists Sponsorship Program (YESS) by China Association for Science and Technology (CAST) under Grant 2018QNRC001, in part by the Fundamental Research Funds for the Central Universities under Grant 2018XKJC06 and Grant 2019PTB-009, in part by the Open Fund of State Key Laboratory of Institute of Information Photonics and Optical Communications (IPOC) of Beijing University of Posts and Telecommunications (BUPT) under Grant IPOC2018A001 and Grant IPOC2019ZT01, in part by the Zhongxing Telecommunication Equipment (ZTE) Research Fund and State Key Laboratory of Satellite Navigation System and Equipment Technology under Grant EX166840043 and Grant CEPNT-2017KF-04, and in part by the Key Laboratory Fund under Grant 6142411182112.

ABSTRACT With the rapid evolution of high-speed mobile communications, cloud computing, and other high-bitrate datacenter-supported services, efficient and flexible traffic scheduling has become one of the fundamental tasks of inter-datacenter optical networks (IDCONs). Traffic scheduling algorithms based on long-term traffic prediction, which have intelligent and global resource allocation ability, have been proved to perform well in IDCONs. However, the low accuracy of existing long-term traffic prediction methods, which is caused by the accumulated errors produced in the recursive multi-step prediction process, directly restricts the efficiency of traffic scheduling. In this paper, we consider the problem of highly efficient traffic scheduling in IDCONs by leveraging one step long-term traffic prediction to reduce the prediction errors. We first design a multiple time interval feature-learning network (MTIFLN) to handle the challenging task of one step long-term traffic prediction. By integrating five bidirectional RNNs (B-RNNs) to one single framework, the MTIFLN has a strong ability to extract the long-term traffic features at different time intervals. Moreover, the stacked architecture of MTIFLN helps to reduce the prediction errors through multi-resampling process. A traffic prediction-based resource allocation (TP-RA) algorithm is proposed together with a global factor to evaluate the efficiency of traffic prediction and achieve effective traffic scheduling based on both traffic prediction results and network resource utilization. Simulation results indicate that with our proposal, the MTIFLN can accurately predict the traffic for more than 24 hours in one step, and the TP-RA algorithm enables IDCONs to make more efficient use of network resources.

INDEX TERMS Long-term traffic prediction, traffic scheduling, inter-datacenter optical networks.

I. INTRODUCTION

In recent years, the rapid growth of high bit-rate datacenter (DC) applications, such as virtual reality, live video streaming, and cloud computing, are driving the demand for high-efficient traffic scheduling in inter-datacenter optical networks (IDCONs) [1], [2]. To assign intelligent and efficient resources for the IDCONs traffic, traffic prediction is emerging as a viable solution to cope with the highly efficient traffic scheduling requirements of IDCONs [3]. According to the traffic duration time, traffic prediction can be classified into two types in terms of short-term traffic prediction and long-term traffic prediction, which is related to traffic ranging from a few seconds to several days [4]. If datacenter operators predict traffic in a short-term way (5 minutes or less), the local traffic scheduling plan can be optimized to enable

The associate editor coordinating the review of this manuscript and approving it for publication was Angelos Antonopoulos.

traffic to adapt and update their routes. This is especially useful because the short-term traffic accounts for majority (more than 80%) of IDCONs traffic. Practically, if datacenter operators predict the long-term traffic (a few hours to several days), traffic congestion can be avoided in a global manner, further improving the efficiency of traffic scheduling. For example, social network service operators, such as Twitter and Facebook, can predict information cascades hours before hot events occur, and delay low priority large applications in terms of data backup and video upstreaming to reserve resources for the future traffic flood in advance. Thus, given the increasing demand for traffic scheduling, there is a need for applying long-term traffic prediction for better resource utilization in IDCONs.

Artificial intelligence (AI), including machines learning (ML) and deep learning (DL) [5], [6], is emerging as a technology to realize "smart" optical networking, where AI can help in network control and management functions like flexible traffic scheduling with low cost. Among many AI-based algorithms, ML is the most preferred method for traffic prediction due to its strong ability to provide fine-grained strategies. For instance, Qian *et al.* [7] apply support vector machine (SVM), one of the most representative ML algorithms, for short-term (5 minutes ahead) traffic prediction and the results show that the SVM outperforms many other ML-based prediction methods. Wu and Huang [8] also adopt the SVM for short-term traffic prediction (5 minutes ahead) and propose an empirical mode decomposition (EMD) method to remove the data noise in the raw training samples. Artificial neural network (ANN) is another widely used ML algorithm for traffic prediction because of its capability of handling multi-dimensional traffic data [9]. To enhance the capability of non-linear traffic model learning, new tools such as wavelet transformation are embedded into ANN. Ning and Yunping [10] propose a new prediction model combined the ANN and wavelet neural network (WNN), which employs nonlinear wavelet basis functions, to solve nonlinear fitting problems of short-term prediction (5 minutes ahead).

Recently, DL has drawn growing attention from several researchers, which exploits much deeper and more complex architectures to extract inherent traffic features [11]. In general, DL-based approaches are proved to have much better performance than the ML methods [12]. Troia *et al.* [13] integrate the recurrent neural network (RNN) with gated recurrent units (GRU) to handle the long-term traffic prediction problem (1 hour ahead) and use the results of prediction results to optimize the traffic scheduling of optical backbone networks. Despite its achievements, the long-term traffic prediction is realized through several recursive short-term prediction process (5 minutes each), which would inevitably lead to accumulated error. Mo *et al.* [14] develop a long short-term memory (LSTM) network, which is a variant of RNN to extract the temporal traffic features with long dependence, to accurately predict traffic 30 minutes in one step. They further consider the prediction results as a guidance for traffic scheduling in optical networks. Moreover, some researches combine the LSTM and RNN to obtain more accurate traffic prediction results. Azzouni and Pujolle [15] design a LSTM-RNN framework for predicting short-term traffic (15 minutes ahead), which outperforms the single LSTM.

Up to the present, the above works mainly focus on the problem of short-term traffic prediction or the long-term prediction (1 hour or longer) that based on the recursive multi-step short-term traffic prediction. For multi-step prediction methods, the prediction for the prior time step is used as an input for making a prediction on the following time step. In the case of predicting the traffic for the next 10 minutes, a one-step 5 minutes prediction model is typically developed. This model would be used to predict the first 5 minutes, then this prediction would be used as an input to predict next 5 minutes. Although this recursive strategy can also obtain long-term results, it allows prediction errors to accumulate such that performance can quickly degrade as the prediction time horizon increases. To support the highly efficient traffic scheduling requirements of IDCONs, the problem of one step long-term traffic prediction needs to be well addressed. However, to our knowledge, there is no study on the IDCON one step long-term traffic prediction.

In this paper, we consider the problem of highly efficient traffic scheduling in IDCONs by leveraging one step long-term traffic prediction. To solve the challenging task of one step long-term traffic prediction, we propose the multiple time-intervals feature learning network (MTIFLN) that integrates multiple bidirectional RNNs (B-RNNs) into one framework. MTIFLN can extract the long-term traffic features from both forward/backward directions after multiple resampling process. Then, a traffic prediction-based resource allocation (TP-RA) algorithm is introduced based on the traffic prediction results and network resource utilization. Different from previous works which serve traffic just exploiting the resource utilization, the proposed scheme further considers the long-term prediction. The simulation results demonstrate that the MTIFLN model can effectively improve the accuracy of long-term traffic prediction, and the TP-RA algorithm can utilize both application resources and transport resources efficiently.

The rest of this paper is organized as follows. Section II introduces the long-term traffic prediction problem and the traffic model. In Section III, we describe the structure of the proposed MTIFLN. In Section IV, we delineate the TP-RA algorithm into the traffic scheduling for INDCONs. The performance evaluation and analysis are shown in Section V, and conclusions are drawn in Section VI.

II. LONG-TERM TRAFFIC PREDICTION PROBLEM AND TRAFFIC MODEL

In this section, we first summarize two reasons why the accuracy of current long-term traffic prediction methods is low, and we describe the traffic model of IDCONs. We also introduce the resampling process of training samples as well as the reason why resampling is suitable for long-term traffic feature extraction. This section highlights the unique goal of long-term traffic prediction in IDCONs.

A. LONG-TERM TRAFFIC PREDICTION PROBLEM

Over the past few years, long-term traffic prediction has become an important and challenging topic for improving the efficiency of traffic scheduling in IDCONs. Besides the accumulated errors, there are two reasons for the poor accuracy of one step long-term traffic prediction.

Long-term traffic features are not fully considered. A majority (over 80%) of the IDCONs traffic is short-term traffic that mostly exists in several minutes. However, more than 80% of the entire traffic volume is carried by long-term traffic, which requires high bandwidth (e.g., data migration and live streaming). In general, the long-term traffic features exist over several hours and the traditional short-term training methods cannot fully extract the long-lived features. To effectively learn the long-term features of IDCONs traffic, the learning model should have the ability to capture the long-term features.

Backward traffic features are not fully utilized. The traffic data fed into a prediction model is chronologically arranged and the traffic features are extracted in a forward direction. However, in the training process, the useful features might be filtered out or not efficiently extracted. Therefore, it is essential to consider the features in backward direction to supplement the missing information. Moreover, the periodicity of traffic is another reason for including backward dependency into our work. Analyzing the periodicity of the traffic data from backward directions is also helpful for extract long-term features.

B. INTER-DATACENTER TRAFFIC MODEL

We consider the datacenters are geographically distributed and interconnected through backbone optical networks. The topology of the IDCON is denoted as $G\ (V,\ E)$, where V and E indicate the sets of nodes and links, respectively. Given a time series $(t_1, t_2, \ldots, t_n)$, we denote the traffic at time t_i as $T_i(s_i,\ d_i,\ k_i,\ t_{pi},\ \eta_i)$, where the s_i and d_i are the source and destination ports, k_i represents the bandwidth requirements, t_{pi} is the arrival time, and η_i is the duration time of traffic. We define the traffic is between a given source and destination, rather than classifying it by applications, and all the datacenter in this network can serve this traffic. Here, the total traffic information can be represented as a M-dimensional vector $T = (T_1, T_2, \ldots, T_M)$ at each sample time, where M indicates the number of the traffic. Thus, we have a M-dimensional vector as the input of B-RNN for each timestamp t_i. Our goal is to predict the $(T_{M+1}, T_{M+2}, \ldots, T_{M+m})$, where $m = 1, 2, 3, \ldots$. In addition, we divide the historical traffic into weekday and weekend traffic, and train them separately, because the features of weekdays and weekends traffic are different. The proposed traffic model has the potential to be algorithmically simpler, faster, scalable, and more efficient than other traffic models. However, it is not scalable for the following traffic scheduling, due to the need for excessive message passing between hosts, routers and management systems in IDCONs. In this work, we expect to achieve one step traffic prediction for at least 24 hours. The associated symbol definitions are given in Table 1, where each part of it describes the symbols of traffic model, traffic prediction and traffic scheduling, respectively.

TABLE 1. Table of key notations.

Symbol	Description
Traffic Model	
$G\ (V, E)$	Network model.
T	Set of original traffic.
T', T'', T'''	Set of resampled traffic with different intervals.
t'	Actual arrival time of traffic.
t_p	Predicted arrival time of traffic.
η	Duration time of traffic.
s	Source ports of traffic.
d	Destination ports of traffic.
k	Bandwidth requirements of traffic.
M	Number of traffic.
v, p, q	Number of the resampled traffic with different intervals.
Traffic Prediction	
x	Input of MTIFLN.
y	Output of MTIFLN.
h, h'	Output of hidden layer from different directions.
H	Output of B-RNN 1.
H', H'', H'''	Output of B-RNN 2, 3, 4.
W^*	Weight matrix of different neural network layers.
b	Bias of neural networks.
b_{in}	Bias of input gate of LSTM.
b_f	Bias of forget gate of LSTM.
b_{out}	Bias of output gate of LSTM.
b_c	Bias of memory cell of LSTM.
Traffic Scheduling	
R_c	CPU usage.
R_u	RAM utilization.
B_l	Bandwidths of each candidate path.
H_p	Hop of each candidate path.
α	Global evaluation factor.
μ	Value center of predicted traffic at different time slot.
R	Total amount of application resources.
R_p	Required application resources of predicted traffic.
S	Total amount of transport resources.
S_p	Required transport resources of predicted traffic.
β	Adjustable proportion between prediction and resource parameters with different user requirements.
τ	Adjustable proportion between R_c and R_u.

C. RESAMPLING WITH TIME INTERVALS

To obtain the long-term traffic features, we perform three down-resampling process in the raw traffic data with three different time intervals including 30 minutes, 1 hour and 2 hours, which consider the training efficiency and prediction accuracy. The reason is that too short resampling intervals cannot fully extract the long-term traffic features, while too long resampling intervals will reduce the sample size and lead to overfitting. During the resampling process, we extract three vectors $T' = (T'_1, T'_2, \ldots, T'_v)$, $T'' = (T''_1, T''_2, \ldots, T''_p)$, and $T''' = (T'''_1, T'''_2, \ldots, T'''_q)$, where v, p, and q are the number of the resampled traffic with time intervals of 30 minutes, 1 hour and 2 hours, respectively. Each vector represents the traffic after the resampling process. With these resampled vectors, the input of B-RNNs is transformed to four new sequence of T, T', T'', and T'''. For the missing value of the training data after resampling, linear interpolation is adopted to fill this gap and make sure that all the training vectors have the same number of samples. The linear interpolation for time series data has been shown to be effective and has no obvious influence on the prediction accuracy [16]. In this way, the long-term features will be duplicated many times, and the short-term features that also contribute to the long-term traffic prediction will be preserved.

One of the main objectives of this work is the extraction of the long-term features. However, the original arrival features of traffic might be ignored since the order of traffic is disrupted by resampling. To overcome this defect, we use the B-RNN model to extract features of these out-of-order data. The unique bidirectional architecture allows B-RNNs to extract global contextual features from forward and backward directions.

Then, the reason why resampling is suitable for long-term traffic feature extraction is analyzed. In the process of feature extraction, the prediction error is usually decomposed into bias term and variance term [17]. The bias term measures how poorly the prediction model approximates the actual traffic data, and it is large when the prediction model does not have enough capacity to extract the features. The variance term measures the generalization ability of prediction model, and it is large when the model has too much capacity for modeling the sampling error in the historical traffic data.

The fitting degree of the prediction model is different for resampling training set with different time intervals. Thus, we can get the corresponding variance when the models fit to different resampling training sets. The process of resampling and integrating training set can perturb the training data and increase the diversity of traffic features. As all the training sets are resampled from the same data set, variance will decrease after the resampling and integration progresses. If we adopt a stacked architecture and use predictors with high feature extraction capacity, the prediction results would have low bias while incurring the low variance. This new architecture is introduced in the next section.

III. MODEL ARCHITECTURES

As explained above, there is a need for stacked architecture to allow extracting long-term features from historical traffic data. The envisioned mechanism should be effective when the entire context of the traffic is needed, hence requiring a hierarchical bidirectional prediction solution. In addition, the prediction model should memorize traffic patterns in a relatively long time, such as serval hours, to enhance the long-term feature extraction ability. To obtain more accurate long-term traffic prediction results, we propose the multiple time-intervals feature learning network (MTIFLN).

As shown in Fig.1, the proposed MTIFLN includes four channels for feature extraction with four different time intervals. For each channel, four vectors corresponding to resampled traffic data with four-time intervals are first imported to the B-RNN 1, 2, 3, and 4. These B-RNNs are employed for extracting temporal features from historical traffic data with different time intervals. Outputs generated by B-RNN 1, 2, 3, and 4 are integrated and delivered to the B-RNN 5 to learn globe temporal features of historical traffic data and predict the future traffic. The descriptions of each model in the following sub-sections all focus on the long-term traffic prediction.

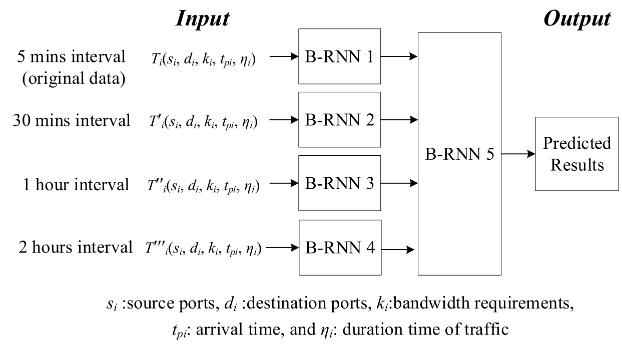


FIGURE 1. The structure of the proposed MTIFLN.

A. LONG SHORT-TERM MEMORY

To extract the features with long-range dependency of long-term traffic from historical traffic data, we adopt the LSTM architecture as a part of the proposed structure in this work. Several previous studies have shown that LSTM works well on prediction tasks. Although many variants of LSTM have been introduced in recent years, large-scale analysis of LSTM variants shows that these variants cannot significantly improve the performance of the standard LSTM [18]. Therefore, the origin LSTM architecture is applied in this work and described below.

The difference between LSTM architecture and the traditional RNN architecture is the composition of the hidden layer. The hidden layers of RNN are replaced by the LSTM cells, which also contain input $x(t)$ and output $h(t)$. Each LSTM cell has an input gate $in(t)$, a forget gate $f(t)$, an output gate $out(t)$ and a memory cell $c(t)$. This unique gate control architecture, especially the forget gate, helps LSTM to be an effective model for long-term feature extraction [19]. The corresponding output $h(t)$ of LSTM can be calculated according to Eq. (1)-(6).

$$in(t) = sigm(W_{x,in}x(t) + W_{h,in}h(t-1) + b_{in}) \quad (1)$$
$$f(t) = sigm(W_{x,f}x(t) + W_{h,f}h(t-1) + b_f) \quad (2)$$
$$out(t) = sigm(W_{x,out}x(t) + W_{h,out}h(t-1) + b_{out}) \quad (3)$$
$$\hat{c}(t) = \tanh(W_{x,c}x(t) + W_{h,c}h(t-1) + b_c) \quad (4)$$
$$c(t) = in(t)\Theta\hat{c}(t) + f(t)\Theta\hat{c}(t-1) \quad (5)$$
$$h(t) = out(t)\Theta \tanh(c(t)) \quad (6)$$

where W_* is the weight matrix of different neural network layers, and b is the bias. It should be noted that W_* is different in different layers. The operation Θ presents the dot product. The hidden status of LSTM is $(c(t), h(t))$. The long-term sequence data is saved in $c(t)$, the output gate is used to update the sequence data, and the forget gate is used to filter out useless information.

B. THE B-RNN MODEL

The B-RNN model can process traffic data in both forward and backward directions through two separate LSTM cells, and then connect the outputs of two hidden layers to the same output layer.

In this work, we train the B-RNN model by combining the forward and backward LSTM layers with back propagation through time (BPTT) algorithm. The BPTT training algorithm training process is divided into three steps. First, we calculate the output $h(t)$ of each hidden forward and backward layers. Then, the error value from both resampling process and neural network architecture will be calculated. After obtaining the output of hidden layers and error value, we can update the gradient of weight.

Specifically, given the traffic vector data $(x_1, x_2, \ldots, x_M)$, the hidden state $h(t)$ of forward LSTM layer can be obtained. Similarly, if we input the traffic vector to the backward LSTM layer, another hidden state $h'(t)$ can be obtained. The $h(t)$ and $h'(t)$ are regarded as different expressions of the hidden sequence data from different directions. Therefore, the output of the hidden layer of each B-RNN model is

$$H(t) = \frac{1}{k} \sum_{i=1}^{k} [h_i(t) + h_i'(t)] \tag{7}$$

which can be imported into the full-connected output layer for final traffic prediction.

To obtain satisfactory prediction results, the model initialization process is necessary. We initialize the neural network weight W_* of LSTM and the logistic regression layer by random values extracted from the zero-mean Gaussian distribution with standard deviation 0.01. Except for the forget gate, all the biases are initialized to 0. In the beginning of training, we set biases of the forget gate b_f to a higher value of 5, which is to make sure that important information is not lost, and the long-term sequence data can be better trained. Moreover, the hidden status of LSTM layers is also initialized to 0. The optimization step is also essential for B-RNN. In this paper, we use an extended expression of stochastic gradient descent algorithm called Adam [20], which can update the weight iteratively based on the training data. These strategies can significantly reduce the requirements of computing resources and decrease the convergence time, which is useful for improving training efficiency.

C. THE MTIFLN FRAMEWORK

The proposed MTIFLN includes 5 B-RNN modules. B-RNN 1, B-RNN 2, B-RNN 3 and B-RNN 4 are applied to extract long-term or short-term features of resampled traffic data with different time intervals. Outputs of these B-RNN modules are imported into B-RNN 5 for global features learning and traffic predicting. In addition, B-RNN 1, B-RNN 2, B-RNN 3 and B-RNN 4 are designed not only to extract traffic features but also to uniform the sample sizes for B-RNN 5 because the inputs of four channels have different sample sizes. Thus, we can fuse and extract the multiple time-intervals features by using MTIFLN.

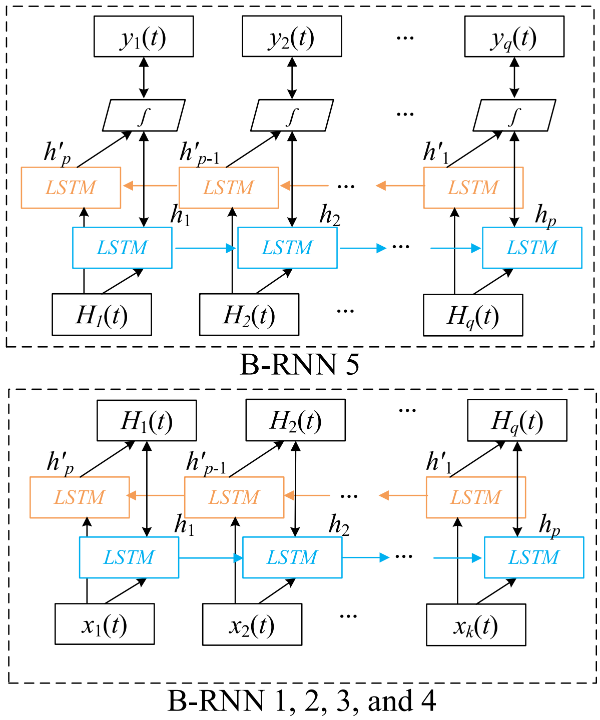


FIGURE 2. The structure of the B-RNN modules.

As shown in Fig. 2, B-RNN 1, B-RNN 2, B-RNN 3 and B-RNN 4 have two LSTM layers with both forward and backward directions, and B-RNN 5 also has two LSTM layers, followed by an output layer for prediction. We use the historical traffic data without resampling process as the input of B-RNN 1, and in this case, the original short-term features would be retained. To extract the long-term feature, we input the resampled historical traffic data to B-RNN 2, B-RNN 3 and B-RNN 4, with time intervals of 30 minutes, 1 hour and 2 hours, respectively. After extracting, both the long-term and short-term features can be memorized and then fed into the B-RNN 5 for further fusion and globe feature extraction. We integrate the outputs of different B-RNN models and generate a new training data set for B-RNN 5 model. Assume that the output of B-RNN 1, 2, 3, 4 are $H(t)$, $H'(t)$, $H''(t)$, $H'''(t)$, respectively, and the input of B-RNN 5 is the integration of the outputs of other B-RNNs ($H(t)$, $H'(t)$, $H''(t)$, $H'''(t)$). The $\int$ is the gate activation function, which normally is the sigmoid function, and the tanh is the hyperbolic tangent function. The final output of B-RNN 5 is an output sequence $(y_1(t), y_2(t), \ldots, y_q(t))$, which are the long-term prediction results. With this formulation, there is no time interval restriction in a relatively long period. Furthermore, we argue that our proposed MTIFLN can achieve better performance than a single B-RNN model by mixing features of different time intervals when handling the long-term traffic prediction problem.

In addition, we can estimate that the computational complexity of MTIFLN model is $O((K+1)mn)$, where K denotes the number of resampling, m means the number of hidden layer nodes, n indicates the number of input layer nodes. Considering that the computational complexity of the MTIFLN model is higher than traditional prediction algorithms, we adopt several data preprocessing methods to accelerate the training of the predictor including data reduction and data cleaning. During the data reduction phase, we choose the five elements from a large amount of information of original traffic data. In the data cleaning phase, we modify the samples with missing content and remove the noise data manually to generate complete and consistent samples for traffic prediction.

In our work, we use the MTIFLN to capture the long-term features of training set. The stacked architecture can extract long-term features from training set with different time intervals, and the bidirectional neural network structure is used to allow the RNN to take full account of the context information of traffic data. After obtaining a satisfactory prediction model, we can get the prediction results by entering new traffic data. Then the prediction results will be used as a guidance of the following traffic scheduling process to allocate resources in IDCONs.

IV. TRAFFIC PREDICTION-BASED RESOURCE ALLOCATION ALGORITHM

In this section, we design a traffic prediction-based resource allocation (TP-RA) to improve the efficiency of resource utilization. In the heart of the TP-RA, we calculate the traffic priority and reserve resources for future traffic based on prediction results and existing resources.

In the TP-RA algorithm, we consider three factors that affect the traffic scheduling process including the arrival time of future traffic, the application resources (i.e., CPU usage $R_c(t)$ and RAM utilization $R_u(t)$) and the transport resources (i.e., bandwidths B_l and hop H_p of each candidate path) [23]–[27].

To evaluate the traffic priority before traffic scheduling, we introduce a global evaluation factor α in Eq. (8) that contains the prediction, resource parameters.

$$\alpha = \frac{\left|E\left[t_{pj}\cdot t'\right]-\mu_{pj}(t)\,\mu_{pj}(t')\right|}{\sqrt{D\left[t_{pj}\right]}\cdot\sqrt{D[t']}}\beta + \left(\frac{\sum_{j=1}^{M}\int_{t_0}^{t_N}R_{pj}(t)\,dt}{\int_{t_0}^{t_N}R(t)\,dt}+\frac{\sum_{i=1}^{M}\int_{t_0}^{t_N}S_{pi}(t)\,dt}{\int_{t_0}^{t_N}S(t)\,dt}\right)\frac{(1-\beta)}{M} \tag{8a}$$

$$\left|E\left[t_{pj}\cdot t'\right]-\mu_{pj}(t)\,\mu_{pj}(t')\right|<1 \tag{8b}$$

The first term of Eq. (8a) represents the prediction parameter, which characterizes the prediction error between predicted results and actual traffic through defining correlation coefficients. In the case of long prediction period, we consider the prediction results of overall traffic instead of the single traffic. In the first part of Eq. (8a), t_{pj} represents the arrival time of the predicted traffic, t' represents the actual arrival time of traffic, μ_p is the value center of t_{pj}. Considering the minus sign before the term, the larger the prediction function is, the more relevant the predicted value is to the actual value. In other words, the prediction will be more accurate.

The second term of Eq. (8a) represents both the application and transport resource parameters, which uses integral to describe resource consumption in the global perspective. In the second part of Eq. (8a), R represents the total amount of application resources, R_{pj} is the resources including the application resources required for the jth arrival predicted traffic, S denotes the total amount of transport resources, S_{pi} represents the resources including the transport resources required for the ith arrival predicted traffic, and M is the total amount of traffic in IDCONs. The overall datacenter resource function with the parameters of each datacenter is expressed as dimensionless Eq. (9) and Eq. (10), where the parameters are normalized to meet the linear relationship between them. In Eq. (9), τ is the adjustable proportion between CPU usage and RAM utilization.

$$R(t) = \tau R_c(t) + (1-\tau)R_u(t) \tag{9}$$

$$S(t) = \sum_{l=1}^{H_p} B_l \tag{10}$$

In addition, $0<\beta<1$ is defined to balance the prediction and resource parameters with different user requirements. Equation (8b) ensures that for each time slot occupied by the traffic request, the global evaluation factor α can be normalized and positive. This global evaluation factor, which can illustrate the expected traffic arrival and the resources required to be reserved, considers both the accuracy of long-term traffic prediction and global resource utilization. Given the evaluation factor, we can calculate the traffic priority with minimum α value for optical signal and continuous spectrum path.

After the traffic priority calculation, allocate resources for each arrival predicted traffic t_{pj} according to the priority of traffic. In the heavy traffic scenario, we need to ensure that high-priority traffic (HPT) is given preferential process. In addition, we will reserve appropriate amount of resources for high-priority predicted traffic to ensure the transmission. Low-priority traffic (LPT), only requires a basic service provision and can be transmitted after the HPT. In this work, the HPT and LPT are calculated according to global evaluation factor α rather than fixed. We distinguish between HPT and LPT by setting a threshold, which is determined by network conditions. Therefore, there are two possible states for different priorities of the arrival traffic.

First, each predicted traffic could be allocated with sufficient resources in time when it reaches the datacenter network. In this case, no additional work is required. Servers will

```
Algorithm: TP-RA
Input: G(V, E), R(t), D(t)and T_i(s_i, d_i, k_i, t_pi, η_i)
Output: Optimal destination of datacenter and resource allocation
Initialize the evaluate factor α
Obtain predicted results T_i
for each T_i
  if no path is found and α_i= min {α_i} then
    block the request
  else
    if α_i> min {α_i} then
      if there exists LPT in the queue then
        if ΣR_j(t')>R(t)
          for each LTP_j in the queue
             block the LTP_j, and calculate R_j(t')
             R_j(t')=R_j(t')+R_{j+1}(t')
             for each coming traffic
                calculate D_i(t')
             end for
          end for until (R_j(t')>R(t))
             allocate resources for T_i when it arrivals
        end if
      end if
    end if
      block the request
  end if
    update R(t) and D(t)
end for
```

FIGURE 3. The pseudocode of proposed TP-RA algorithm.

allocate required network resources and transport resources upon traffic arrival.

Second, resources are not sufficient when the predicted traffic arrives. In this case, we need to check if there is LPT in the predicted traffic queue, and then discard or block the LPT until the process of HPT is complete. If there are still not sufficient consecutive resources when predicted traffic arrives, we check whether there is any LPT in existing traffic queue. If the evaluation factor α is more than the threshold, blocking LPT to preserve resources for HPT.

Finally, we update the traffic queue. The pseudocode of proposed TP-RA algorithm is shown in Fig. 3 and the time complexity of TP-RA algorithm is $O(n^2)$.

V. PERFORMANCE EVALUATIONS

In this section, we first evaluate the performance of the proposed MTIFLN, and then demonstrate the performance of TP-RA algorithm in IDCONs.

A. DATASET DESCRIPTION

The traffic data are collected every 5 minutes from 3 university datacenters including total 1590 servers, which are deployed in Beijing China in June 2019. Each trace in the traffic data includes: a) timestamp, b) priority of traffic, c) arrival time and processing time, d) source and destination ports, and e) IP addresses of source and destination. Although the MTIFLN model is powerful enough to discover and learn inherent features, we define more features to strengthen the ability of long-term extraction in terms of f) day of week, g) hour-to-hour autocorrelation, and h) day-to-day autocorrelation. As explained in Section II, since the traffic pattern of weekend and weekday is different, we use the day of week features to capture the different traffic scale of weekend and weekday. The hour-to-hour and day-to-day autocorrelations are designed to represent the long-term features of traffic in different time intervals.

Then, we add the results to the input characteristics of each corresponding B-RNN model. This simple solution can greatly reduce the prediction error. Then, we normalize all the above features of traffic to zero mean and unit variance. In our simulations, we have about 1000 hours of historical traffic data and want to predict the next 24 hours traffic. We set the prediction interval to a target time (instead of the fixed short-term 5 minutes). In other words, if we want to predict the traffic in next 24 hours, we need to set the prediction interval to 24 hours (one-step prediction).

One of the main objectives of this work is the extraction of the long-term traffic features of historical traffic data, learned using the MTIFLN model. Therefore, after the feature preprocessing, we resample the traffic dataset with different time intervals to obtain new training samples for MTIFLN by using MATLAB. During the resampling, three down-resampling process with time intervals of 30 minutes, 1 hour and 2 hours are deployed. In addition, the linear interpolation method is adopted to fill the gap of missing value and make sure that all the training vectors have the same number of samples. As shown in Fig. 4, the origin dataset of one-week traffic and the traffic information after three resampling process are illustrated. With the resampling interval increases, more and more short-term features of longer resample intervals are being deliberately ignored. In this way, the long-term features will be duplicated many times, and those short-term features from original data set that also contribute to the prediction of the long-term traffic prediction will be preserved. That means the B-RNNs can extract the long-term features without losing short-term features, because the different training data with different resample intervals will enable B-RNNs to learn different characteristics.

Finally, after obtaining all the traffic datasets (about 480,000 of each vector T, T', T'' , and T'''), we sort them in ascending order of their arrival time. Considering the training efficiency of the MTIFLN, we divide each vector into the same proportions of 70/20/10 training/validation/ testing datasets for further training process. That is a conventional data split method for deep learning, which splits the dataset into independent parts for training, validation, and testing, respectively.

B. SIMULATION SETUP

To verify the performance of the proposed MTIFLN and TP-RA algorithm in different network scales, we consider two network topologies, i.e., NSFNET (14 switching nodes, 21 bidirectional multicore fiber links, and 5 DCs) and USNET (24 switching nodes, 43 bidirectional multicore fiber links, and 8 DCs) in simulations for the IDCONs, as shown in Fig. 5.

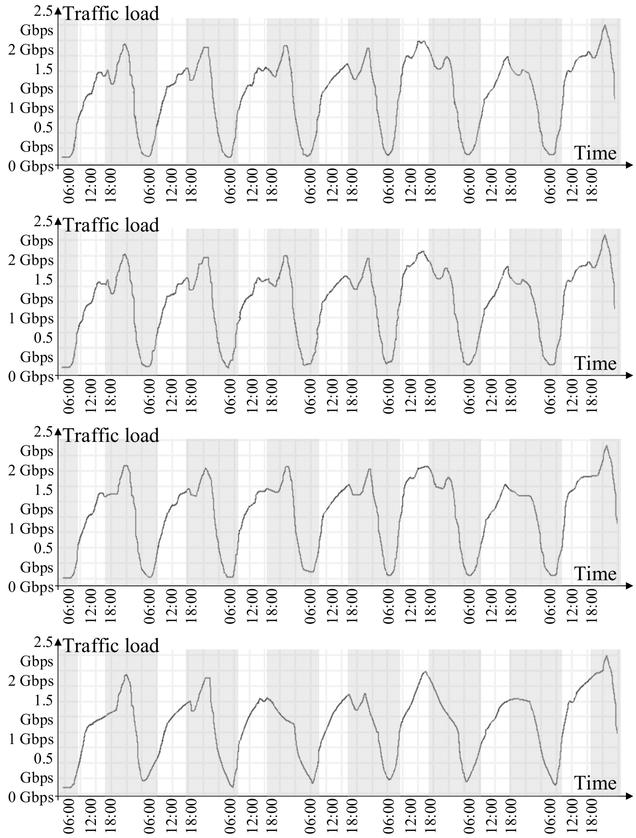


FIGURE 4. The illustration of examples of four training dataset in one week. The first diagram shows the raw traffic dataset collected from the network management system every 5 minutes. The rest (from top-to-bottom) shows the traffic datasets after resampling with time intervals of 30 minutes, 1 hour and 2 hours.

The simple NSFNET topology allows us to verify the effectiveness of the proposed algorithm more efficiently, while the USNET topology is more realistic and has more links that are complex and diverse. Both the two above topologies have the same link capacity of 1 Gbps in two directions, and the propagation delays are chosen to approximate realistic values between respective pairs of nodes. Furthermore, we choose the first three shortest links between each DCs from all the possible links of the IDCONs to reduce the computational cost of resource allocation. We use a multi-core server with 16 2.10GHz Intel Xeon(R) CPU E5-2620 v4 cores, 2 NVIDIA TITAN XP GPU cores and 64GB RAM to accelerate the training of MTIFLN. The server runs Ubuntu 16.04, and we compile our code in Theano [28] framework using python 2.7/3.5.

To construct the B-RNN model, we need to setup the neural network parameters, including the size of the input layer, the number of hidden layers, and the number of hidden units in each hidden layer. In our work, we use the proposed model to predict the traffic including 30 minutes traffic prediction, 2 hours traffic prediction, 24 hours traffic prediction, and 72 hours traffic prediction. To find the network architecture suitable for different target, for B-RNN 1, 2, 3, and 4, we choose the input units from 1 to 10, hidden layers from 1 to 5, and the number of hidden units from {32, 64, 128, 256,

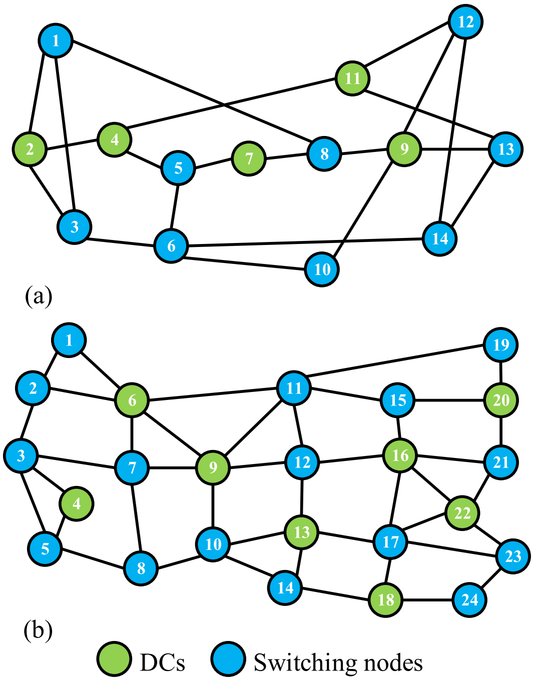


FIGURE 5. Network topologies used in simulation. (a) five DCs are inter-connected with NSFNET topology (14 nodes); (b) eight DCs are inter-connected with USNET topology (24 nodes).

512}. For B-RNN 5, the number of input units is equal to the total number of the hidden units of B-RNN 1, 2, 3, and 4. Moreover, the number of hidden layers for all B-RNNs is 2, because the B-RNNs only have two forward/backward hidden layers. The output layer units of B-RNN 5 is according to the prediction tasks. For instance, the output layer units of B-RNN 5 is 288 when we want to predict the following 24 hours traffic, because there are 288 five minutes sessions in one day. To accelerate the training of MTIFLN, the batch-size of all the B-RNNs is 32. After iterating, we obtained the most suitable architecture for different traffic prediction tasks as shown in Table 2.

To evaluate the validity of the prediction results from the perspective of accuracy and resources, we define the prediction accuracy as $P = 1 - \alpha$, where α is the global evaluation factor as shown in Eq. (8) and β is equal to 0.5. In addition, we use the Mean Absolute Error (MAE), the Mean Relative Error (MRE), and the Root Mean Square Error (RMSE) to evaluate the effectiveness of the proposed traffic prediction model, which are defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |g_i - \tilde{g}_i| \tag{11}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n} \frac{|g_i - \tilde{g}_i|}{g_i} \tag{12}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (|g_i - \tilde{g}_i|)^2} \tag{13}$$

where g_i is the real traffic, and $\tilde{g}_i$ is the predicted traffic.

TABLE 2. Structure of the MTIFLN.

Tasks	Parameters	B-RNNs 1	2	3	4	5
30 min	Input units	5	5	5	5	704
	Hidden layers	2	2	2	2	2
	Hidden units	256	256	128	64	256
	Output units					6
	Learning rate	0.001	0.001	0.001	0.006	0.008
2 h	Input units	4	4	4	4	352
	Hidden layers	2	2	2	2	2
	Hidden units	128	128	64	32	128
	Output units					24
	Learning rate	0.001	0.001	0.001	0.006	0.008
24 h	Input units	8	8	8	8	1408
	Hidden layers	2	2	2	2	2
	Hidden units	512	512	256	128	512
	Output units					288
	Learning rate	0.008	0.008	0.008	0.01	0.02
72 h	Input units	5	5	5	5	352
	Hidden layers	2	2	2	2	2
	Hidden units	128	128	64	32	128
	Output units					864
	Learning rate	0.01	0.01	0.01	0.015	0.03

During the training process, the Adam algorithm is used to optimize the MTIFLN. The training of B-RNN model takes about 16 hours. After training, the one-day traffic prediction time (on GPU) is about 0.7962 s.

C. PERFORMANCE OF TRAFFIC PREDICTION

We first compare the accuracy of the 72 hours long-term traffic prediction based on the state-of-the-art convolutional neural network (CNN) algorithm and the proposed MTIFLN. The CNN adopts the recursive multi-step prediction methods with 5 minutes time step. The CNN model shares the architecture consisting of 352 input units, 2 hidden layers, 128 units in each hidden layer, 864 output units that is similar to B-RNN 5 of MTIFLN when predicting the 72 hours traffic [29]. Specifically, an original training sample T is obtained by modifying the historical data, and we mix them with other resampled training samples in T', T'', and T''' to get the training dataset, because we want to reduce the impact of training data quantity on accuracy as much as possible.

Fig. 6 shows the results of the proposed MTIFLN model for the long-term traffic prediction of weekday and weekend, while the actual traffic load is also included for comparison. As shown in Fig. 6, the predicted traffic load of both CNN and MTIFLN has similar traffic patterns with the actual traffic load at the first 10 hours. However, the performance of the CNN model decreases when the traffic pattern changes, which is caused by many non-linear factors such as

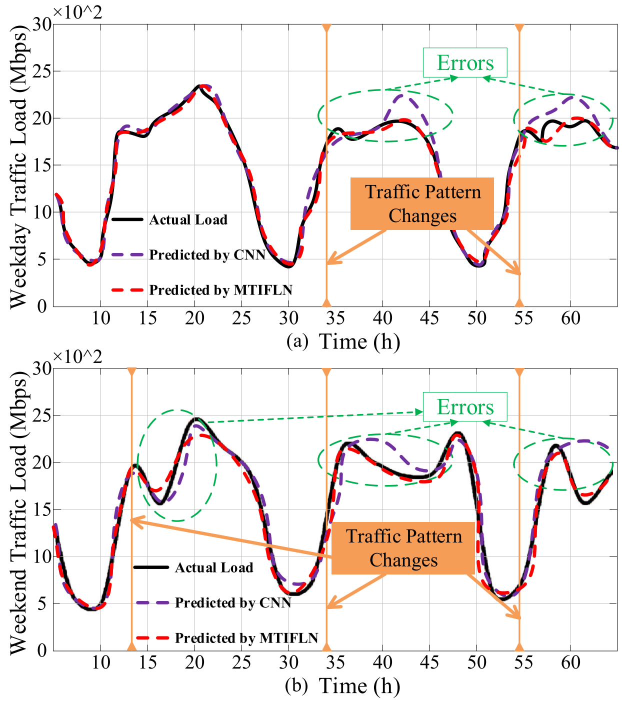


FIGURE 6. Comparison among the real traffic load, the MTIFLN based-prediction results and the CNN-based prediction results at (a) Weekdays, (b) Weekends.

the hot spot events, users' mobility pattern, etc. Meanwhile, the MTIFLN can maintain the accuracy of prediction after the change of traffic pattern. The reason for this phenomenon is that the hierarchical prediction solution can fully extract the long-term features from the resampled training datasets. Moreover, the forward/backward B-RNNs can make full use of the data of past and future time series, thus, we can estimate the traffic pattern changes more accurately. For the traffic after 55 hours, even the accuracy of MTIFLN is starting to decrease, because there are more unpredictable traffic pattern changes. At weekends, the prediction error of CNN becomes more intolerable, as the traffic model changes more frequent. That is because the high-priority traffic (HPT) with large bandwidth and high bustiness occur more frequently than weekdays, while the low-priority traffic (LPT) that is easy to predict is reduced.

To verify the accuracy of MTIFLN for different traffic prediction tasks, we compared the prediction performance of MTIFLN with the CNN, the single B-RNN, and the SVM for the traffic prediction tasks including 30 minutes, 2 hours, 24 hours, and 72 hours traffic prediction. All four algorithms predict the traffic in one-step instead of the recursive multi-step prediction. In these three compared methods, the CNN model has good performance for the prediction of time series data [30]. This model shares the same architecture of B-RNN 5 of MTIFLN. The single B-RNN is a representative of deep learning algorithms [20]. This model also shares the same architecture of B-RNN 5 of MTIFLN. The SVM method is a representative of machine learning algorithms. This model

TABLE 3. Performance of CNN, B-RNN, MTIFLN, and SVM.

Tasks	Model	Accuracy (%)	MAE	MRE (%)	RMSE
30 min	CNN	92.9	17.6	4.4	27.6
	B-RNN	90.4	24.8	6.2	37.2
	MTIFLN	97.7	17.1	4.3	26.4
	SVM	89.4	34.9	6.7	52.3
2 h	CNN	87.1	45.4	5.6	68.1
	B-RNN	81.3	53.8	13.5	80.7
	MTIFLN	93.6	44.8	4.7	67.2
	SVM	73.5	111.6	20.9	167.4
24 h	CNN	80.3	90.3	20.6	135.5
	B-RNN	72.1	109.6	27.4	164.4
	MTIFLN	90.9	53.5	11.2	80.2
	SVM	55.3	234.5	27.8	351.7
72 h	CNN	63.4	196.1	25.4	294.2
	B-RNN	65.3	271.3	30.5	406.9
	MTIFLN	76.6	93.9	11.2	140.9
	SVM	40.1	480.3	43.7	720.4

adopts the Radial Basis Function as the kernel function, and we train it through one-against-all decomposition [31]. In all cases, we use the same data sets for training, validation, and testing during the prediction.

In Table 3, we can see that the prediction accuracy of the MTIFLN is over 85% even for the 72 hours prediction task. In addition, it has low MAE, MRE, and RMSE values, which are much better than the other three compared algorithms. Note that the CNN model has high prediction accuracy for the first two prediction tasks (30 minutes and 2 hours traffic prediction), but the accuracy decreases rapidly compared with MTIFLN for the rest prediction tasks (24 hours and 72 hours traffic prediction). That is because the prediction has less variable factors at the first 2 hours, and the traffic rules can be obtained through a large number of original data training. However, there are many reasons for the change of traffic model in the long-term prediction, which makes the prediction more difficult for the CNN model. For the MTIFLN, this stacked bidirectional model can accurately extract long-term traffic features, which are essential for long-term prediction.

In Table 3, we also note that the MTIFLN is proved more accurate than the CNN, the B-RNN, and the SVM for the long-term prediction. The CNN has relatively high prediction performance, which is from 80% to 92% or so. While the average prediction accuracy of B-RNN and SVM drops much with the increase of time interval of the traffic data increasing. For the 30 minutes traffic prediction, the average accuracy of the B-RNN and the SVM is 90.4% and 89.4%, respectively. However, for the 72 hours traffic prediction task, the average accuracy of the B-RNN and the SVM has a large drop, which is 65.3% and 40.1%, respectively. It can be seen from the table that the proposed MTIFLN performs well in traffic prediction over 24 hours.

TABLE 4. Table of baseline models.

Model	Description
TP-RA	Proposed algorithm with the prediction results of MTIFLN.
CNN	Proposed algorithm with the prediction results of CNN.
B-RNN	Proposed algorithm with the prediction results of B-RNN.
SVM	Proposed algorithm with the prediction results of SVM.
CSO	Proposed algorithm without prediction.
FF	Schedule traffic in order of arrival time without prediction.
MTIFLN	Proposed one-step long-term traffic prediction method.
MSLT	Recursive multi-step long-term traffic prediction method.

D. PERFORMANCE OF TRAFFIC SCHEDULING

In this subsection, we leverage the scheme in Section V to launch the traffic scheduling in IDCONs based on the prediction results of MTIFLN, and demonstrate the vulnerabilities of TP-RA. We adopt two topologies (shown in Fig. 5) to evaluate the performance and scalability of TP-RA algorithm and compare it with the traditional traffic scheduling algorithms including the state-of-the-art cross stratum optimization (CSO) (proposed in [25]) and traditional First Fit (FF). Moreover, we also compare TP-RA with the resource allocation algorithm based on traffic prediction results. In Table 4, we summarize the baseline models used in this work.

For the CNN, single B-RNN, and SVM algorithms, we use the same resources allocation method based on their prediction results, respectively. In all cases, we use the same datasets T, T', T'', and T'''. The traffic requests to datacenter servers are setup with bandwidth randomly from 50Mbps to 500Mbps, and the resource usage of each server is randomly set from 0.1% to 1%.

Figure 7 compares the path blocking probability among TP-RA, CSO, FF, and TP-RA based on the prediction results of CNN, single B-RNN and SVM in two topologies. We can see that TP-RA algorithm achieves lower path blocking probability values as compared to the other algorithms in both topologies. That is because the TP-RA avoids the conflict of HPT in the heavy loaded router, where many services might be blocked or discarded due to the queue overflow. The predicted traffic is more likely to be processed successfully in the resource allocation phase. In addition, the path blocking probability of TP-RA algorithm increases when the traffic arrival rates rise (i.e., at 180, 210, 240 Erlang). This is because the DC servers in the low traffic load can provide more available resources than that in high traffic load scenario.

Fig. 8 shows the comparisons on resource occupation rate among six algorithms. The resource occupation rate reflects the percentage of occupied resources to all the resources in datacenter. We can see that the TP-RA could greatly enhance the resource occupation rate compared to the other algorithms in two topologies with different network scales. In the more complex topology USNET, the TP-RA performs even better and more consistent improvement in resource occupation

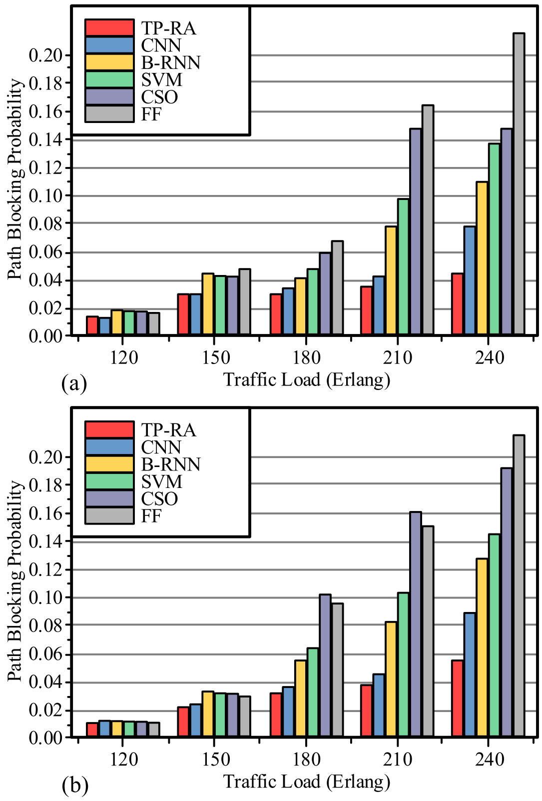


FIGURE 7. Comparisons of path blocking probability among different resource allocation strategies. (a) NSFNET; (b) USNET.

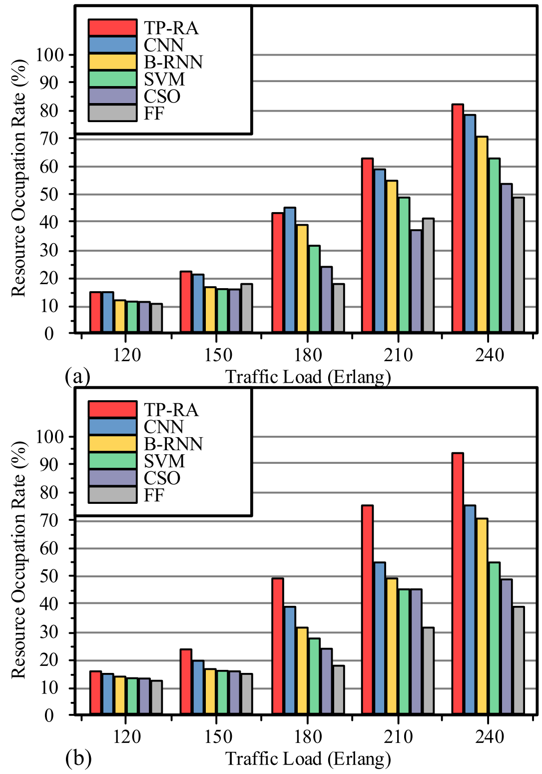


FIGURE 8. Comparisons of resource occupation rate among different resource allocation strategies. (a) NSFNET; (b) USNET.

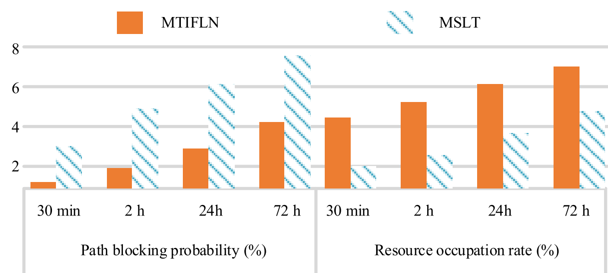


FIGURE 9. Comparisons of MTIFLN and recursive-based multi-step long-term (MSLT) traffic prediction on traffic scheduling in INDCONs.

rate. This is justified by the fact that TP-RA can preserve sufficient resources for predicted HPT in advance, reducing the blocking probability of HPT. As expected, the average blocking probability and resource occupation rate decreases with different IDCONs network scales.

Fig. 9 compares the performance of the proposed MTIFLN and recursive-based multi-step long-term (MSLT) traffic prediction on traffic scheduling in INDCONs [32]. To make the comparison more intuitive, MTIFLN adopts the same network structure as shown in Table 2, while the MSLT traffic prediction methods share the same structure of B-RNN 5 of MTIFLN. Both MTIFLN and MSLT use the more complex topology USNET as shown in Fig. 5, and both use the same training samples for training, validation, and testing. The MTIFLN directly predicts all the traffic of the prediction tasks (30 minutes, 2 hours, 24 hours, and 72 hours). The MSLT traffic prediction method set the time step to 5 minutes, and the prediction for the prior time step is used as an input for making a prediction on the following time step until all the traffic of the prediction tasks is predicted. As shown in Fig. 9, the MTIFLN outperforms the MSLT in four different prediction tasks, and the performance difference between them increases with the increase of prediction time. That is because the MSLT allows prediction errors to accumulate in each time step, which would lead to a rapid decline in prediction accuracy. Unlike the recursive-based MSLT, the proposed MTIFLN can predict the long-term traffic in one step. In fact, the stacked architecture of MTIFLN helps to decrease the prediction errors, because the prediction errors in the previous four B-RNNs would cancel each other out during the resampling process before input to B-RNN 5. Thus, MTIFLN increases the prediction accuracy to cope with the traffic scheduling requirements and improve the network performance.

VI. CONCLUSION

This paper studies two tightly coupled tasks including long-term traffic prediction and corresponding traffic scheduling in IDCONs. With the improvement of the accuracy of long-term prediction, the efficiency of traffic scheduling has been greatly optimized. To extract the long-term features that hidden in the historical data, we first perform the resampling process to break the limitation of the long-term series and propose the multiple time-intervals feature learning network (MTIFLN). This stacked architecture comprises five single B-RNNs integrated with LSTM cells to achieve high-precision long-term traffic prediction. Besides traffic prediction, this paper also introduces traffic prediction-based resource allocation algorithm in IDCONs, which can reserve resources for future traffic in advance based on a global evaluation factor. In the simulation phase, we evaluate the traffic prediction performance of MTIFLN for four different long-term traffic prediction tasks. Additionally, the efficiency of TP-RA algorithm and the benefits of long-term traffic prediction are also verified. After delaying the low-priority traffic, the resource utilization is dramatically improved and the path blocking probability is greatly decreased in two topologies with different network scales. As the future work, the relationship between inter-datacenter fabric and traffic prediction will be further explored.

REFERENCES

[1] H. Yang, J. Zhang, Y. Zhao, J. Han, Y. Lin, and Y. Lee, "SUDOI: Software defined networking for ubiquitous data center optical interconnection," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 86–95, Feb. 2016.

[2] A. Ghosh, S. Ha, E. Crabbe, and J. Rexford, "Scalable multi-class traffic management in data center backbone networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2673–2684, Dec. 2013.

[3] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, 1st Quart., 2016.

[4] S. K. Singh and A. Jukan, "Machine-learning-based prediction for resource (Re) allocation in optical data center networks," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D12–D28, 2018.

[5] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1383–1408, 2nd Quart., 2018.

[6] J. Mata, I. De Miguel, R. J. Duran, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," *Opt. Switching Netw.*, vol. 28, pp. 43–57, Apr. 2018.

[7] Y. Qian, J. Xia, K. Fu, and R. Zhang, "Network traffic forecasting by support vector machines based on empirical mode decomposition denoising," in *Proc. Int. Conf. Consum. Electron., Commun. Netw.*, 2012, Paper 3327-3330.

[8] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, 2009.

[9] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: A comparison between traditional and deep learning neural networks," *Int. J. Big Data Intell.*, vol. 3, no. 1, pp. 28–37, 2016.

[10] M. Ning and C. Yunping, "An ANN and wavelet transformation-based method for short term load forecast," in *Proc. Int. Conf. Energy Manage. Power*, 1998, Paper405–410.

[11] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4761–4773, Oct. 15, 2018.

[12] A. Yu, H. Yang, W. Bai, L. He, H. Xiao, and J. Zhang, "Leveraging deep learning to achieve efficient resource allocation with traffic evaluation in datacenter optical networks," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, 2018, Paper W4I. 2.

[13] S. Troia, R. Alvizu, Y. Zhou, G. Maier, and A. Pattavina, "Deep learning-based traffic prediction for network optimization," in *Proc. IEEE 20th Int. Conf. Trans. Opt. Netw. (ICTON)*, Jul. 2018, pp. 1–4, Paper 1-4.

[14] W. Mo, L. Gutterman, Y. Li, G. Zussman, and D. Kilper, "Deep neural network based dynamic resource reallocation of BBU pools in 5G C-RAN ROADM networks," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, 2018, Paper Th1B. 4.

[15] A. Azzouni and G. Pujolle, "A long short-term memory recurrent neural network framework for network traffic matrix prediction," 2017, *arXiv:1705.05690*. [Online]. Available: https://arxiv.org/abs/1705.05690

[16] T. Blu, P. Thévenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 710–719, May 2004.

[17] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 569–575, Mar. 2010.

[18] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[19] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, p. 2451, 2000.

[20] Z. Cui, K. Ruimin, and Y. Wang, "Deep stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2017, *arXiv:1801.02143*. [Online]. Available: https://arxiv.org/abs/1801.02143

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[22] H. Yang, Q. Yao, A. Yu, Y. Lee, and J. Zhang, "Resource assignment based on dynamic fuzzy clustering in elastic optical networks with multi-core fibers," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3457–3469, May 2019.

[23] H. Yang, W. Bai, A. Yu, Q. Yao, J. Zhang, Y. Lin, and Y. Lee, "Bandwidth compression protection against collapse in fog-based wireless and optical networks," *IEEE Access*, vol. 6, pp. 54760–54768, 2018.

[24] H. Yang, J. Zhang, Y. Zhao, Y. Ji, J. Wu, Y. Lin, J. Han, and Y. Lee, "Performance evaluation of multi-stratum resources integrated resilience for software defined inter-data center interconnect," *Opt. Express*, vol. 23, no. 10, pp. 13384–13398, Nov. 2015.

[25] H. Yang, J. Zhang, Y. Ji, and Y. Lee, "C-RoFN: Multi-stratum resources optimization for cloud-based radio over optical fiber networks," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 118–125, Aug. 2016.

[26] H. Yang, J. Zhang, Y. Zhao, Y. Ji, J. Han, Y. Lin, and Y. Lee, "CSO: Cross stratum optimization for optical as a service," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 130–139, Aug. 2015.

[27] H. Yang, K. Zhan, M. Kadoch, Y. Liang, and M. Cheriet, "BLCS: Brain-like based distributed control security in cyber physical systems," *IEEE Netw.*, vol. 99, no. 1, pp. 1–7, Dec. 2020.

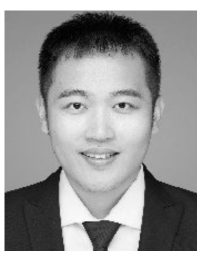
[28] H. Yang, J. Zhang, Y. Ji, Y. He, and Y. Lee, "Experimental demonstration of multi-dimensional resources integration for service provisioning in cloud radio over fiber network," *Sci. Rep.*, vol. 6, Jul. 2016, Art. no. 30678.

[29] F. Bastien, P. Lambin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: New features and speed improvements," 2012, *arXiv:1211.5590*. [Online]. Available: https://arxiv.org/abs/1211.5590

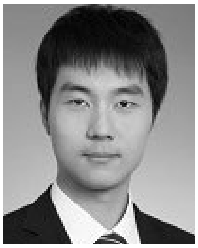
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] M. Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," in *Proc. ACM 10th Int. Conf. Mach. Learn. Comput.*, 2018, Paper 26-30.

[32] D. Zhang and S. Wang, "Neural network based multi-step predictive control for nonlinear systems," *Control Decis.*, vol. 20, no. 3, pp. 332–336, 2005.



AO YU received the B.S. degree from the China University of Petroleum (UPC), Shandong, China, in 2014. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His main research interests include deep learning, data center optical networks, traffic prediction, and fault location.



HUI YANG is currently the Vice Dean and an Associate Professor with the Beijing University of Posts and Telecommunications (BUPT). He has authored or coauthored 100 articles in prestigious journals and conferences. He is the first author of more than 50 of them. His research interests include SDN, fixed-mobile access networks, cross-stratum optimization, data center network, flexi-grid optical networks, blockchain, and so on. He is an Active Reviewer or a TPC Member for several journals and conferences. He received the Best Paper Award from NCCA'15 and Young Scientist Award in IEEE ICOCN'17. He has served as a Guest Editor of the IEEE JSAC, an Associate Technical Editor of the IEEE ComMag, and a General Chair of ISAI'16.

TING XU is currently pursuing the bachelor's degree with the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. Her main research interests include deep learning and traffic prediction.

BAOGUO YU is currently the Chief Scientist of the China Electronics Technology Group Company, Ltd., and the Deputy Chief Engineer of the State Key Laboratory of Satellite Navigation System and Equipment Technology.

QIUYAN YAO received the M.S. degree from the Hebei University of Engineering, Handan, China, in 2014. She is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. Her researches mainly focus on SDN, network architecture, access networks, flexi-grid optical networks, and so on.

YAJIE LI is currently an Associate Professor with the Beijing University of Posts and Telecommunications (BUPT). His research interests include deep learning, optical access networks, and so on.

TAO PENG is currently a Senior Engineer with Zhongxing Telecommunication Equipment Corporation (ZTE). His research interests include artificial intelligence, optical access networks, fault location, and so on.

HUIFENG GUO is currently a Senior Engineer with Zhongxing Telecommunication Equipment Corporation (ZTE). Her research interests include artificial intelligence, optical access networks, fault location, and so on.

JUN LI is currently a Senior Engineer with the 504th Research Institutes of China Aerospace Science and Technology Corporation. His research interests include artificial intelligence, optical access networks, fault location, and so on.

JIE ZHANG is currently a Professor and the Dean of the Institute of Information Photonics and Optical Communications, BUPT. He is supported by more than ten projects of the Chinese Government. He has published eight books and more than 100 articles. He has seven patents have also been granted. His researches focus on optical transport networks, packet transport networks, and so on. He served as a TPC Member for ACP 2009, PS 2009, ONDM 2010, and so on.