This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

A Multi-Task Learning-based Network Traffic Prediction Approach for SDN-enabled Industrial Internet of Things

Shupeng Wang, Laisen Nie, Guojun Li, Yixuan Wu, and Zhaolong Ning

***Abstract*—With the rapid advance of Industrial Internet of Things (IIoT), to provide flexible access for various infrastructures and applications, Software Defined Networks (SDN) has been involved in constructing current IIoT networks. To improve the quality of services of industrial applications, network traffic prediction has become an important research direction, which is beneficial for network management and security. Unfortunately, the traffic flows of the SDN-enabled IIoT network contain a large number of irregular fluctuations, which makes network traffic prediction difficult. In this paper, we propose an algorithm based on multi-task learning to predict network traffic according to the spatial and temporal features of network traffic. Our proposed approach can effectively obtain network traffic predictors according to the evaluations by implementing it on real networks.**

***Index Terms*—Network traffic prediction, Multi-task learning, Industrial Internet of Things, Software Defined Networks**

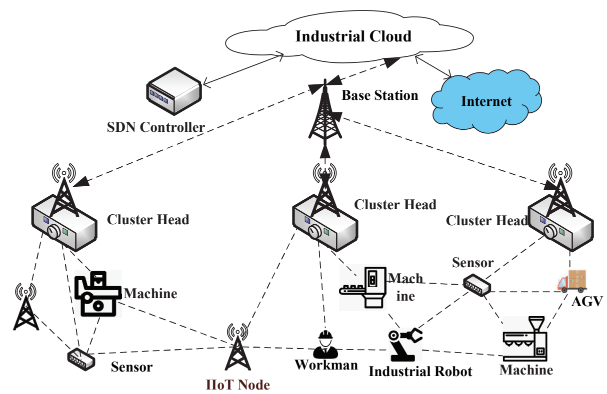


Fig. 1. An illustration of the SDN-enabled IIoT network.

I. Introduction

In recent years, Internet of Things (IoT) has become an indispensable and important infrastructure to provide intelligent life, such as intelligent transportation system and smart home [1]. The increase of network users promotes the rapid development of IoT. As a special paradigm of IoT, Industrial Internet of Things (IIoT) is a novel term that provides ubiquitous connections for sensors, automated guided vehicles, machines, and industrial robots [2]. The applications of IIoT can improve industrial productivity and efficiency obviously [2], [3]. Software-Defined Network (SDN) improves network controllability by the separation of the data and control planes. Without introducing new link layer protocols, the SDN can be compatible with existing industrial communication protocols, and it is flexible to modify and reconstruct the network based on various quality of service requirements of IIoT users. Hence, the SDN can improve the efficiency of IIoT and reduce network complexity significantly [4]–[6]. The structure diagram of the SDN-enabled IIoT network is shown in Fig. 1.

As the foundation for the development of contemporary society, it is very important to manage the network to make it efficient and secure. Network traffic is an important parameter for network management and security. Thereby, traffic reconstruction and prediction are crucial research directions in SDN-enabled IIoT networks. By extracting statistical features of network traffic, network operators can implement network management and security functions appropriately. The accurate predictor of network traffic is significant for network managers to plan and allocate network resources reasonably, and it also can ensure the normal operation of the network to improve the efficiency of network utilization [7].

In order to solve the challenges of traffic prediction, researchers have launched a lot of work. For instance, Li et al. proposed a dictionary learning-based alternating direction method to solve the problem of traffic learning and prediction in cellular networks [8]. The proposed method consists of a coarse prediction module, a sparse constraint and refine module, and an alternating direction module. Besides, for traffic prediction problems, Song *et al.* proposed an online rolling traffic prediction method [9]. The proposed method can learn the low-dimensional embedding data, and also can impute missing embedding data. Although many traffic prediction techniques have been developed, existing traffic prediction techniques still have some challenges in SDN-enabled IIoT networks. First, current traffic prediction methods do not effectively consider spatio-temporal features. Besides, the fitting effect of existing methods for traffic flow needs to be improved.

The accurate prediction of network traffic is significant to optimize the network structure and bandwidth distribution in

Shupeng Wang is with Institute of Information Engineering, Chinese Academy of Sciences (CAS), Beijing, China, e-mail: wangshupeng@iie.ac.cn.
Laisen Nie (Co-corresponding author) and Yixuan Wu are with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, China, e-mail: {nielaisen, wuyixuan1}@nwpu.edu.cn.
Guojun Li (Co-corresponding author) is with Lab of BLOS Trusted Information Transmission, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China, e-mail: ligj@cqupt.edu.cn.
Zhaolong Ning is with the School of Software, Dalian University of Technology, Dalian 116024, China, e-mail: z.ning@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

SDN-enabled IIoT networks. Besides, network traffic prediction has important significance to the efficient management, security, and resource allocation of the SDN-enabled IIoT network. Aiming at the problems of network tomography and the spatio-temporal correlation of network traffic in SDN-enabled IIoT networks, we integrate Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) network, and use CNN and LSTM network to extract the spatio-temporal features of network traffic, respectively. We can obtain the spatio-temporal features of data by convolution operation. Therefore, we use CNN to deal with the problem of spatial-temporal connection of network traffic and get the spatial-temporal features of network traffic. For the obvious temporal features of network traffic, we use LSTM network to obtain this information. At the same time, for the sake of the accuracy of traffic prediction, according to the linear relationship among link load, network traffic, and routing matrix, we put forward a traffic prediction algorithm based on Multi-Task Learning (MTL). The main contributions of this paper are as follows:

- Network traffic expresses various statistical features, e.g., spatial, temporal, and spatio-temporal dependencies. Aiming at improving prediction errors caused by irregular network traffic fluctuations, we take into account the spatio-temporal features of network traffic as a prior information for prediction. To capture the spatio-temporal features of network traffic effectively, we design a deep architecture based on CNN and LSTM. The kernels used in the designed CNN can extract the irregular fluctuations of network traffic.
- Based on that, the network tomography model, which shows the linear relationship between network traffic and link load, is used as a redundant noise for the designed deep architecture. We take into account network tomography model and network traffic prediction jointly, and use multi-task learning to improve the accuracy of network traffic prediction.
- We implement our prediction approach on a real IIoT network data set. Existing network traffic prediction methods usually focus on the problem of the IIoT and traditional Internet service provider backbone networks. The traffic flows in a backbone network show a periodic trend that is easy to be tracked. Unfortunately, the traffic trends of users and network edge are not periodic but irregular. We consider the realistic scenario consisting of users and network edge in this paper, and the designed architecture can obtain a precise network traffic predictor as well as the periodic temporal features, according to the evaluation on the real IIoT network data set.

The following sections of this paper are arranged as follows. In Section II, we introduce the recent research work on network traffic prediction. Section III gives the preliminary of traffic prediction. In Section IV, we describe our traffic prediction method for the SDN-enabled IIoT network which combines CNN, LSTM, and multi-task learning. In Section V, the Abilene and IIoT network data set are used to evaluate the proposed method. Finally, in Section VI, we summarize our work in this paper.

II. Related Work

A. *SDN-enabled Traffic Prediction Methods*

Traffic prediction system is the basic requirement of network applications in SDN-enabled IIoT networks. Many applications in data center networks, such as anomaly detection, network planning, load balancing, traffic engineering, network security, and resource scheduling, require an accurate traffic prediction system to monitor and record network traffic. A number of different approaches have been proposed to overcome the challenges of network traffic prediction. Lazaris *et al*. proposed an LSTM module for SDN-enabled traffic prediction system [10]. They used historical measured traffic data to train their LSTM model. The LSTM module is a scalable framework, and the method can predict traffic flows with various rate time series. Besides, Al-Jamali *et al*. put forward a network traffic prediction algorithm based on partial recurrent spike neural network [11]. The architecture of recurrent spike neural network is composed of three network layers, namely input layer, hidden layer with multiple self-feedback neurons, and output layer. The model predicted the next round of network traffic by learning the historical traffic.

B. *Machine Learning-based Methods*

Traffic prediction allows network operators to achieve satisfied service. For the data prediction method, Liu *et al*. proposed a fast adaptive gradient radial basis function network [12] to realize traffic prediction. This model can be used to predict non-linear and non-stationary time series traffic data. The model simulated the data through the orthogonal least square method and replaced the nodes with the worst performance in operation. Combining Markov chains with tensors to realize network traffic prediction is one effective method. Liu *et al*. proposed a novel multi-variate multi-order Markov transform to realize traffic prediction [13]. First of all, they proposed tensor concatenation and unified product. Then, through the combination of unified product and Markov model, a new model was obtained to achieve high-performance traffic prediction. Moreover, Li *et al*. proposed a smoothing-aided support vector machine for traffic prediction [14]. The method used the smoothing method to effectively alleviate the disturbance caused by the drastic fluctuations of network traffic flows.

At present, it has been demonstrated that the fitting effectiveness of the non-linear prediction method based on machine learning algorithm is far better than the previous linear prediction method, and the prediction method based on neural network has higher accuracy. Therefore, deep learning techniques have been widely used in IIoT networks [15]. Chen *et al*. designed a hybrid traffic prediction scheme based on an LSTM network and sparse auto-encoder [16]. They trained the LSTM network and the sparse auto-encoder network respectively to get the corresponding trained model. Morales *et al*. proposed a virtual network topology reconfiguration method for traffic prediction [17]. In this model, since artificial neural network can adapt to network traffic in a non-supervised

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

manner, they used artificial neural network to achieve stable and adaptive traffic prediction.

Due to the strong spatial correlation of traffic sequences, spatial information has always been included. Meanwhile, it is of great significance to apply Deep Learning (DL) to the field of network traffic prediction. Nie *et al*. proposed a Monte Carlo Q-Learning (MCQL) method to realize prediction method [18]. The method modeled traffic data as a Markov decision process. Then, they reduced the dimensions of the Traffic Matrix (TM) by using dictionary learning method. Finally, they used MCQL to make traffic prediction. Guo *et al*. proposed a 3D CNN method to realize traffic prediction [19]. The traffic data were modeled using 3D CNN and recalibration block respectively. Then, they predicted traffic by weighting the modeled data. Moreover, Filali *et al*. investigated a preemptive SDN load balancing with machine learning for traffic prediction [20]. In this model, they constructed two prediction algorithms based on auto regressive integrated moving average and LSTM approaches to forecast data. Meanwhile, Hardegen *et al*. realized the traffic prediction by using a machine learning component leveraging deep neural networks, and through context-related tagging to achieve high precision of traffic prediction [21].

III. Problem Definition

A. *End-to-End Network Traffic*

A series of network entities (e.g., sensors and routers) make up the SDN-enabled IIoT network, as illustrated in Fig. 2. A TM shows the traffic between any Origin-Destination (OD) node pairs in the network. For different nodes, the TM can be divided based on Point of Presences (PoPs), links, and routers.

We define the TM as matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \begin{bmatrix} X(1,1) & X(1,2) & \cdots & X(1,T) \\ X(2,1) & X(2,2) & \cdots & X(2,T) \\ \vdots & \vdots & \cdots & \vdots \\ X(N^2,1) & X(N^2,2) & \cdots & X(N^2,T) \end{bmatrix}, \quad (1)$$

where $X(n,t)$ represents the value of the nth OD flow at time t, $n = 1,2,...,N^2$, $t = 1,2,...,T$. After that, the problem of traffic prediction can be denoted as:

$$X(n,t+1) = f\left(X(n,t), X(n,t-1), \ldots, X(n,t-p+1)\right), \quad (2)$$

where p represents the length of prior network traffic for prediction. In other words, the network traffic prediction method predicts the next time data $X(n,t+1)$ through the known traffic $\left(X(n,t), X(n,t-1), \ldots, X(n,t-p+1)\right)$.

The above definition regards the traffic as a time series, and the traffic prediction in this case only considers the temporal features of network traffic. Considering that the TM has spatial features such as the heavy-tailed and power-law distributions, the end-to-end network traffic prediction problem can be defined as follows:

$$\boldsymbol{X}(:,t+1) = g\left(\boldsymbol{X}(:,t), \boldsymbol{X}(:,t-1), \ldots, \boldsymbol{X}(:,t-p+1)\right), \quad (3)$$

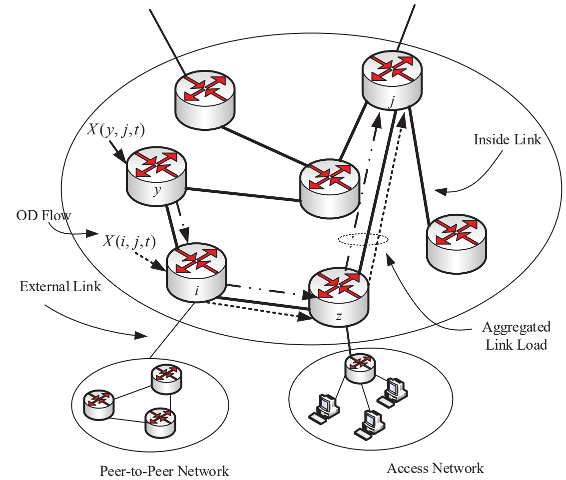


Fig. 2. An illustration of network tomography technique.

where $\boldsymbol{X}(:,t+1) = \left(X(1,t+1), ..., X\left(N^2,t+1\right)\right)^T$ represents a snapshot of the TM. We use DL method to fit the function $g(\cdot)$, so as to realize network traffic prediction.

B. *Network Tomography*

Network tomography is an end-to-end network traffic measurement technique [22]. It defines the linear relationship between the TM, link load, and routing matrix, which can be expressed as:

$$\boldsymbol{Y}(:,t) = \boldsymbol{B}\boldsymbol{X}(:,t), \quad (4)$$

where $\boldsymbol{Y}(:,t)$ and $\boldsymbol{B}$ represent the link load and routing matrix, respectively. The network tomography model is widely used in network traffic estimation, because link load is available by the simple network management protocol.

IV. Our Methodology

A. *Network Traffic Prediction Based on DL*

The DL architecture designed for network traffic prediction based on CNN and LSTM is shown in Fig. 3. The CNN can use local data to reflect the overall situation, and it can obtain spatio-temporal features of TM. Besides, it has a fast and effective feature extraction capability. Therefore, we use CNN to get the spatio-temporal features of network traffic. For the obvious temporal features of network traffic, the LSTM network is used in this paper. At the same time, we use multi-task learning at the end of our approach to improve the fitting effectiveness. The additional task can be viewed as a redundant noise of the designed deep architecture, which has been proved that it can increase training efficiency.

The deep architecture for network traffic prediction contains 6 hidden layers as shown in Fig. 3, i.e., a convolutional layer, a pooling layer, an LSTM layer, two fully connected layers, and a dropout layer. The first hidden layer is a convolutional layer, which contains $Z^{(l)}$ convolution kernels with size $Q_1 \times Q_1$. The convolutional layer is followed by a pooling layer of size $Q_2 \times Q_2$ with a stride of Q_2. The LSTM and fully connected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

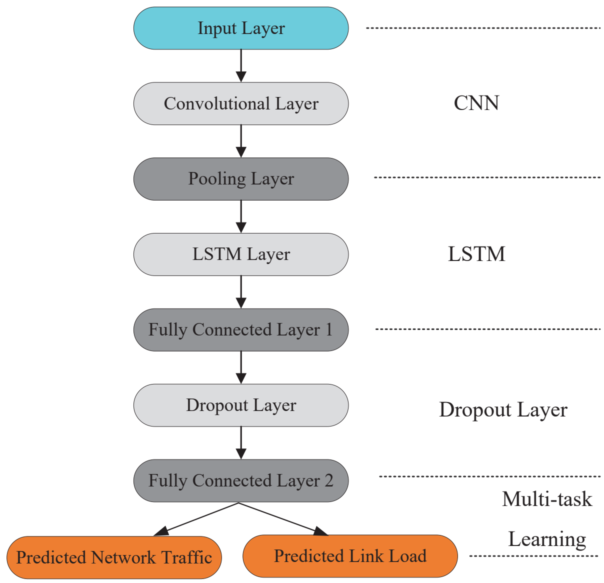


Fig. 3. Deep architecture for network traffic prediction.

layers are set to Q_3 and Q_4 neurons, respectively. Furthermore, the dropout layer is generally set as 50% dropout, and the amount of output units of the last fully connected layer is $N^2 + R$ for traffic prediction based on MTL, where R is the number of links.

We define the input of the designed deep architecture as $\boldsymbol{X}$, and $\boldsymbol{H}_i$ is the feature map of the ith layer. Then $\boldsymbol{H}_0 = \boldsymbol{X}$, and $\boldsymbol{H}_i$ can be expressed as:

$$\boldsymbol{H}_1 = f\left(\boldsymbol{H}_0 \otimes \boldsymbol{W}_1 + \boldsymbol{b}_1\right), \tag{5}$$

where $\boldsymbol{W}_1$ denotes the weight vector of the first layer, and $\boldsymbol{b}_1$ denotes the bias. Notation $\otimes$ denotes a convolution operation. The convolutional layer is followed by the pooling layer of size $Q_2 \times Q_2$ with a stride of Q_2, which pools the local parts of each neuron, so as to achieve the purpose of extracting the features of feature maps again. The neurons in the pooling layer are locally connected to the convolutional layer, thus it can reduce the number of neurons in the pooling layer and computational complexity. Besides, a pooling layer includes maximum pooling, mean pooling and so on. Maximum pooling is to maximize the local value, which can well extract local features. Mean pooling averages local values to reflect the overall features of data.

LSTM is used to solve the issue of gradient vanishing and exploding gradient existing in Recurrent Neural Network (RNN). Compared with RNN, LSTM has improved output gate, input gate, and forget gate in the module. There are three main phases in an LSTM. The first one to run is the forget gate, which is mainly to selectively forget the input from the previous node and remember important information. The process is shown as follows:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f\left[\boldsymbol{s}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_f\right), \tag{6}$$

where $\boldsymbol{f}_t$ denotes the output of forget gate. $\boldsymbol{W}_f$ denotes the weight of forget gate. $\boldsymbol{b}_f$ is its bias, and σ is the sigmoid activation function. Besides, $\boldsymbol{s}_{t-1}$ is the output of the previous state, and $\boldsymbol{x}_t$ is the input of the current state. After that, the input gate is processed. The input gate processing is shown as follows:

$$\begin{cases} \boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i\left[\boldsymbol{s}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_i\right) \\ \tilde{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_c\left[\boldsymbol{s}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_c\right) \end{cases}, \tag{7}$$

where $\boldsymbol{i}_t$ and $\tilde{\boldsymbol{c}}_t$ are the output of the input gate and the cell input activation vector, respectively. Notation $\tanh(\cdot)$ represents the hyperbolic tangent activation function. $\boldsymbol{W}_i$ and $\boldsymbol{W}_c$ denote the weights of the input gate and the cell state vector, respectively. $\boldsymbol{b}_i$ and $\boldsymbol{b}_c$ are their biases. Then, we will update the cell state $\boldsymbol{C}_t$, which records the state information of data, and it plays the role of information transmission. This process is shown as follows:

$$\boldsymbol{C}_t = \boldsymbol{f}_t \otimes \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \otimes \tilde{\boldsymbol{c}}_t. \tag{8}$$

After that, the output gate and output of LSTM are as follows:

$$\begin{cases} \boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o\left[\boldsymbol{s}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_o\right) \\ \boldsymbol{a}_t = \boldsymbol{o}_t \otimes \tanh(\boldsymbol{C}_t) \end{cases}, \tag{9}$$

where $\boldsymbol{a}_t$ is the output of the current state. $\boldsymbol{o}_t$ shows the output of output gate. $\boldsymbol{W}_o$ is the weight of outout gate, and $\boldsymbol{b}_o$ is its bias.

B. Network Traffic Prediction Based on Multi-Task Learning

MTL is a paradigm of machine learning. It improves the fitting effectiveness of the DL architecture by sharing the information with each other. In DL, the sharing of hard or soft parameters in the hidden layer is the main realization form of MTL. All tasks in hard parameter sharing use the same hidden layer, but several output layers for specific tasks are set aside in the output layer. The advantage of hard parameter sharing is that it can greatly reduce the possibility of over-fitting. Our approach adopts the method of hard parameter sharing for network traffic prediction in SDN-enabled IIoT networks.

The SDN-enabled IIoT network traffic has various features, e.g., mutagenicity, self-similarity, long-range dependency, and chaos. These features make the SDN-enabled IIoT network fundamentally different from the traditional Internet. The traditional network traffic prediction approach is based on investigating the features of network traffic and using mathematical model to measure its statistical features, but it is not quite suitable for the current SDN-enabled IIoT network. That is because the traffic flows of SDN-enabled IIoT networks obey non-linear features. Motivated by that, the DL method can be used to predict the nonlinear network traffic. However, a DL architecture based on single task learning cannot provide a precise predictor of network traffic with various non-linear features. To deal with this issue, we arise an additional task regarded as a noise. Network tomography, as a network traffic inference technique used to obtain end-to-end network traffic, describes the linear relationship among traffic flows, routing information, and link loads. In our approach, we take advantage of the network tomography model as the additional task to construct traffic prediction based on MTL.

The large-scale network measurement cannot be implemented, due to the measurement cost of the traditional IP

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

network. Hence, the traditional network traffic prediction approaches need a matrix interpolation algorithm to obtain the prior of network traffic. Under SDN-enabled IIoT network, it can provide a network-level traffic measurement. As a result, our approach can predict network traffic without the completion of prior. Meanwhile, without the matrix interpolation procedure, it is useful to carry out real-time network traffic prediction. Thereby, for M training samples $\left(X_{t,K}^{(m)}, X_{t+1}^{(m)}\right)$, we denote $\boldsymbol{X}_{t,K}^{(m)} = (\boldsymbol{X}(:,t), ..., \boldsymbol{X}(:,t-(K-1)))$ as K snapshots of the historical TM for prediction. Then, the training sample is defined as $\left(\boldsymbol{X}_{t,K}^{(m)}, \left(\boldsymbol{X}_{t+1}^{(m)}, \boldsymbol{Y}_{t}^{(m)}\right)\right)$. After that, all data sets are normalized, which is shown as follows:

$$\tilde{X}(n,t) = \frac{X(n,t) - \bar{X}(n,t)}{\sigma}, \tag{10}$$

where X, $\bar{X}$, and $\tilde{X}$ represent the original value, the mean value, and the normalized value, respectively. σ represents the standard deviation.

Then, considering the predicted TM and corresponding link loads, we use the Iterative Proportional Fitting Procedure (IPFP) to regulate predictors. We denote a snapshot of predicted TM as $\hat{\boldsymbol{X}}(t)$, and the corresponding link load is $\boldsymbol{Y}(t)$. For SDN-enabled IIoT networks, the constraint condition of the TM is as follows:

$$\begin{cases} \boldsymbol{Y}(t) = \boldsymbol{B}\hat{\boldsymbol{X}}(t) \\ \hat{X}(n,t) \geq 0, \left(n = 1,2,...,N^2\right) \end{cases}. \tag{11}$$

In order to make the final predictors meet above constraints, the IPFP algorithm can be used to adjust network traffic predictors ulteriorly. We define P and $S(X)$ as the sets of joint probability density distribution. If $P(X) \in P$ and $I(P(X)\| S) = \min I(P\| S)$, $P(X)$ is represented as the I-Projection of $S(X)$ on P. The I-Projection on the set of joint probability distribution with constraint $R(Y)$ in $S(X)$ can be calculated as follows:

$$P(X) = \begin{cases} S(x) \cdot \frac{R(Y)}{S(Y)} & otherwise \\ 0 & if \quad S(Y) = 0 \end{cases}, \tag{12}$$

where $R(Y)$ is a low-dimensional probability distribution defined in subspace X, and $S(Y)$ is the marginal effect probability distribution of $S(Y)$ in subspace Y. The IPFP algorithm is an iterative process. We use Eq. (12) to search a joint probability distribution satisfying the current constraints. Then, the iteration goes on until the algorithm converges. The details of the IPFP algorithm for network traffic recalibration are shown in Algorithm 1. The overall traffic prediction algorithm is shown in Algorithm 2.

Algorithm 1 Traffic recalibration based on IPFP

Input: predicted traffic matrix $\hat{\boldsymbol{X}}$; routing matrix $\boldsymbol{B}$; link matrix $\boldsymbol{Y}$.
Output: recalibrated traffic matrix $\boldsymbol{X'}$.
1: **for** $k = 1, 2 \cdots K$ **do**
2: **for** $j = 1, 2 \cdots N^2$ **do**
3: $\beta = \sum_i \frac{B(j,i)\hat{X}(i,t)}{Y(j,t)}$
4: $X'(i,t) = \hat{X}(i,t)/\beta$ for all i when $B(j,i) \geq 0$
5: **end for**
6: **end for**

Algorithm 2 Traffic Prediction Based on MTL

Input: training set $\left(\boldsymbol{X}_{t,K}^{(m)}, \left(\boldsymbol{X}_{t+1}^{(m)}, \boldsymbol{Y}_{t}^{(m)}\right)\right)$; number of training samples M; prior data length K; iteration times L.
Output: traffic matrix $\boldsymbol{X}_{t+1,N^2}$.
1: Normalize the training set $\left(\boldsymbol{X}_{t,K}^{(m)}, \left(\boldsymbol{X}_{t+1}^{(m)}, \boldsymbol{Y}_{t+1}^{(m)}\right)\right)$
2: Let $l = 1$
3: Input $\left(\boldsymbol{X}_{t,K}^{(m)}, \left(\boldsymbol{X}_{t+1}^{(m)}, \boldsymbol{Y}_{t}^{(m)}\right)\right)$ for forward propagation
4: Feedback based on loss function, and update weights and bias
5: If $l \neq L$, let $l = l+1$, go to Step 4 and start training again
6: If $l = L$, saving the trained DL architecture
7: Input the test data into the trained deep architecture to obtain the prediction matrix $\left(\tilde{\boldsymbol{X}}'_{t+1,N^2}, \tilde{\boldsymbol{Y}}'_{t,R}\right)$
8: $\boldsymbol{X'}_{t+1,N^2}$ is obtained after renormalize predicted data $\tilde{\boldsymbol{X}}'_{t+1,N^2}$
9: IPFP is used to optimize $\boldsymbol{X'}_{t+1,N^2}$ to get the final predicted value $\boldsymbol{X}_{t+1,N^2}$

V. Numerical Results

A. Simulation Data Set

The simulation adopts the network traffic data from the Abilene backbone network and an IIoT network testbed. In this paper, we use the Abilene network traffic to simulate the backbone network of an IIoT network. The Abilene network consists of 12 nodes and 54 links. In a network with N nodes, the number of OD flows is N^2, and then the Abilene backbone network has 144 OD flows. The data set via Abilene consists of 2016 time slots for evaluation. Then, the TM, routing matrix, and link load are 144×2016, 54×144, and 54×2016, respectively. The public IIoT data set is collected from a testbed consisting of switches, sensors, personal computers, and servers [23]. We establish the TM of this IIoT network by recording the timestamp of each packet in a PCAP format. In this data set, we record the TM with 9800 time slots for evaluation. The IIoT data set cannot provide the link load and routing matrix, hence we construct the routing matrix by the Dijkstra algorithm, and the link load is calculated by the linear relationship between link load and TM.

The proposed prediction method is implemented based on Matlab. Besides, the machine configuration is Intel Core processor (3.2 GHz) with 8 GB RAM. In the deep architecture, as shown in Table I, the convolutional layer contains 6 convolution kernels with the size of 5×5, and the size of the pooling layer is 2×2 with a stride of 2. The network traffic of IIoT networks shows a large number of irregular fluctuations, thus a small kernel that can extract its spatio-temporal features is better. Meanwhile, LSTM contains 10 neurons, and the first fully connected layer contains 5 neurons. Besides, the numbers of neurons in fully connected layer 2 are 198 and 649 for the Abilene and IIoT networks, respectively. Adaptive

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

TABLE I
NETWORK HYPERPARAMETER SETTING

Layer Category	Parameter
Convolution Layer	Convolution Kernel Size: 5×5 Number of Convolution Kernels: 6
Pooling Layer	Size: 2×2
LSTM Layer	Number of Neural Units: 10
Fully Connected Layer 1	Number of Neural Units: 5 Activation Function: tanh
Dropout Layer	50% dropout
Fully Connected Layer 2	Number of Neural Units for Abilene Network: 198 Number of Neural Units for IIoT Network Network: 649 Activation Function: tanh

moment estimation is used as training for the designed deep architecture. The maximum numbers of training times, initial learning rate, and batch number are set as 1, 0.005, and 2, respectively. We compare the proposed method with Principal Component Analysis (PCA) and Sparsity Regularized Matrix Factorization (SRMF) methods, which are the network traffic prediction methods by extracting the spatio-temporal features of TM. PCA is a method based on principal component and matrix theory analysis. It first projects the TM to the principal component space, in which the TM is low rank. Then, it predicts the end-to-end traffic according to the network tomography model. The SRMF method, as an interpolation algorithm, is proposed to recover the missing network data. The problem of traffic prediction can be known as recovering a TM with missing elements by whole columns. SRMF is expressed as follows:

$$\begin{aligned} \min \left\| \xi\left(\boldsymbol{L}\boldsymbol{K}^T\right) - \boldsymbol{C} \right\|_F^2 + \lambda\left(\|\boldsymbol{L}\|_F^2 + \|\boldsymbol{K}\|_F^2\right) \\ + \left\|\boldsymbol{J}\left(\boldsymbol{L}\boldsymbol{K}^T\right)\right\|_F^2 + \left\|\left(\boldsymbol{L}\boldsymbol{K}^T\right)\boldsymbol{T}^T\right\|_F^2, \end{aligned} \tag{13}$$

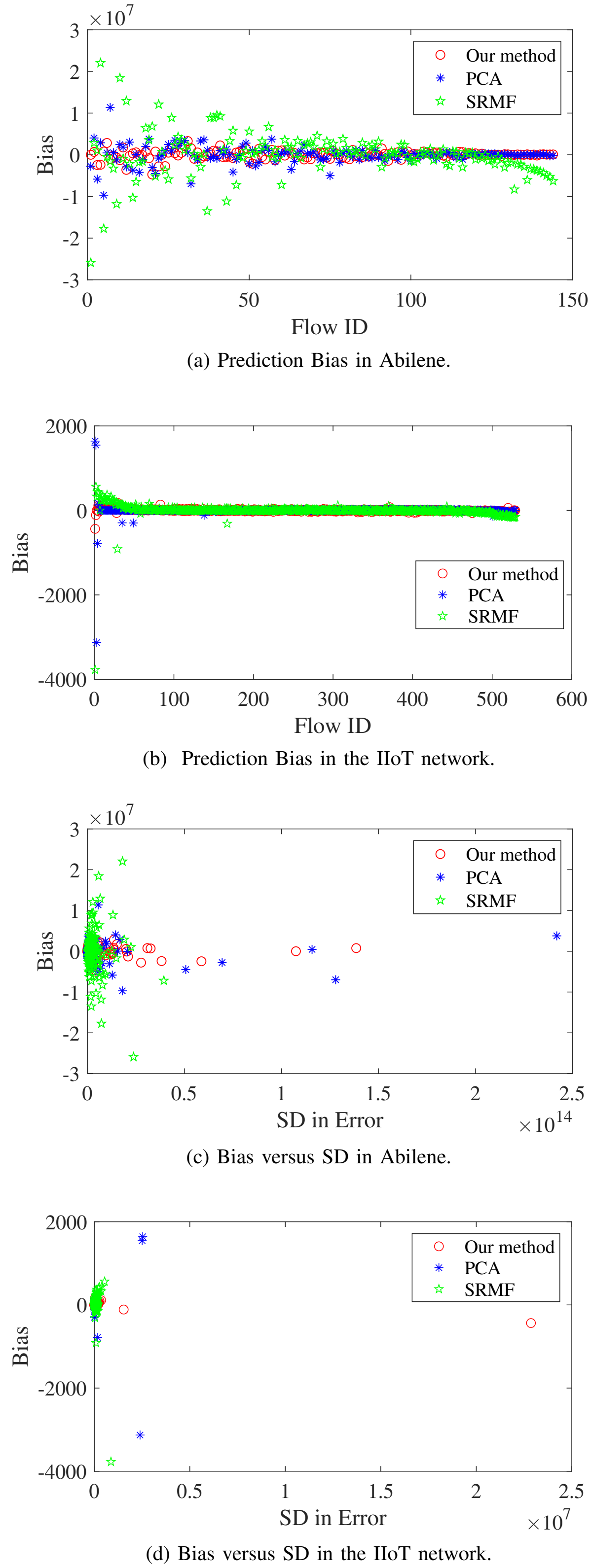
where $\xi(\cdot)$ represents the linear operator, and it denotes data loss in the TM. $\boldsymbol{C}$ represents the observed value of the TM, and λ is the regularization parameter. We can obtain $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{K}^T = \boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^T$, $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{A}^{\frac{1}{2}}$, and $\boldsymbol{K} = \boldsymbol{V}\boldsymbol{A}^{\frac{1}{2}}$ by way of singular value decomposition. Notation $\|\cdot\|_F^2$ represents the Frobenius norm. $\boldsymbol{J}$ and $\boldsymbol{T}$ are spatial and temporal constraint matrices, respectively, which can express the spatial and temporal structures of the TM. $\boldsymbol{V}$, $\boldsymbol{A}$, and $\boldsymbol{U}$ are the singular value decompositions of $\boldsymbol{X}$.

B. Simulation Result Analysis

In this section, we will evaluate the prediction results by using the bias of network data, the sampling Standard Deviation (SD), the Spatial Relative Error (SRE), the Temporal Relative Error (TRE), and the corresponding Cumulative Distribution Functions(CDF), respectively. The bias is defined as follows:

$$error_{bias}(n) = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{X}(n,t) - X(n,t)\right), \tag{14}$$

where $\hat{X}(n,t)$ and $X(n,t)$ are predicted network traffic and its true traffic, respectively.



(a) Prediction Bias in Abilene.

(b) Prediction Bias in the IIoT network.

(c) Bias versus SD in Abilene.

(d) Bias versus SD in the IIoT network.

Fig. 4. Biases and SD in Abilene and IIoT networks.

Fig. 4(a) shows the prediction biases of three methods for the Abilene network data set. The x-axis represents the IDs of OD flows, which is arranged from large to small according to the traffic mean value, and the y-axis is the bias. It can be seen from Fig. 4(a) that the larger the mean value of OD flow is, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics

greater the errors of three methods become. With the decrease of the mean values of traffic flows, the prediction biases of three methods decrease. In general, the MTL-based method has a better prediction bias, and the bias fluctuation is small and stable. PCA and SRMF have large biases in OD flows with larger traffic mean values. For SRMF, when the traffic mean values are small, there will be more obvious negative estimation phenomenon than the other two algorithms.

For the IIoT network data set, as shown in Fig. 4(b), SRMF and PCA have larger biases for predicting OD flows with larger mean values. When the mean value of OD flow is small, SRMF shows a mild negative estimation. Furthermore, the bias of the MTL-based method is relatively stable, and the bias value is close to 0 except for large OD flows.

Figs. 4(c) and 4(d) show the prediction bias and sampling standard deviation of three algorithms for the data set via Abilene and IIoT networks, respectively. The x-axis and y-axis are the sampling standard deviation and prediction bias, respectively. The sampling standard deviation is defined as follows:

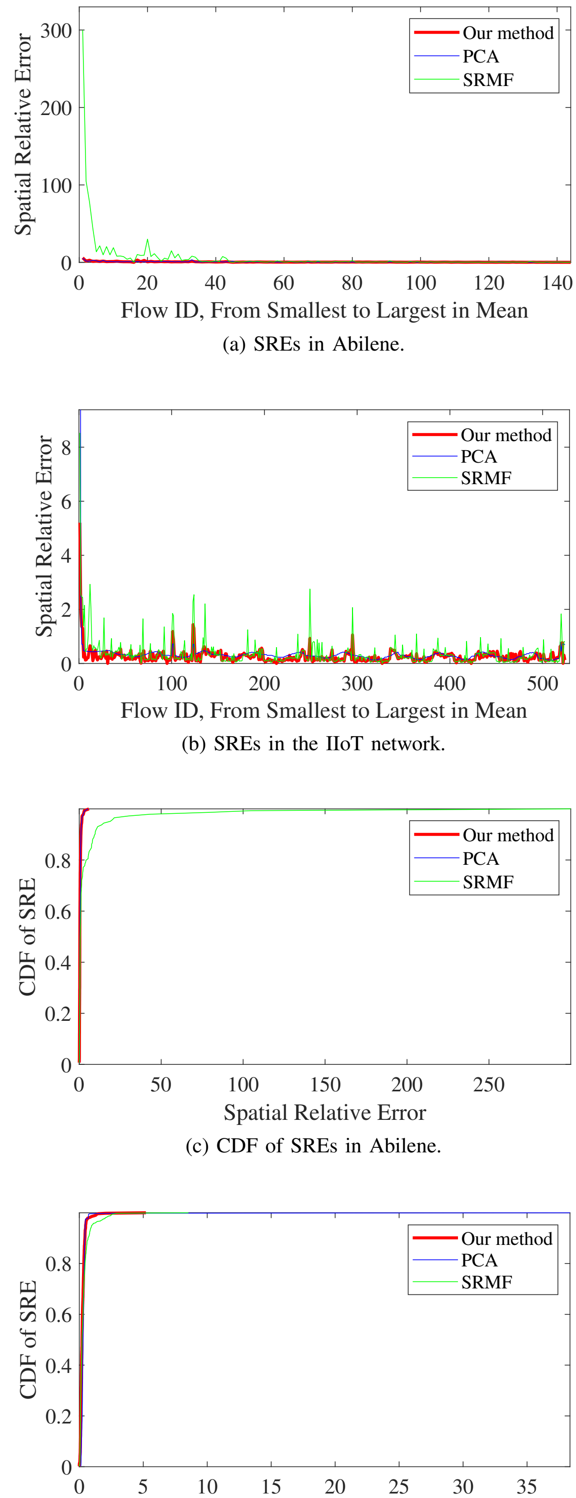
$$error_{SD} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(\hat{X}(n,t) - X(n,t) - error_{bias}(n)\right)^2}. \tag{15}$$

In Fig. 4(c), the prediction bias of the our method in the Abilene network is more stable and smaller, and the sampling standard deviation is also smaller than SRMF and PCA. For SRMF and PCA, when the sampling standard deviation is small, the algorithms have high prediction bias. As shown in Fig. 4(d), for the IIoT network, the sampling standard deviation of our method is larger than the other methods. For SRMF, when the sampling standard deviation is small, it has larger prediction bias. However, on the whole, the sampling standard deviation and prediction bias of the proposed method are smaller. From the evaluations of bias versus SD in Abilene and IIoT networks, our method is suitable for predicting the traffic flows with long range dependency, due to its small prediction bias and relatively large sampling standard deviation. PCA takes advantage of the principal components to approximate a TM for prediction, and SRMF extracts the relationship of adjacent network traffic elements (i.e., spatio-temporal features) to predict a TM. As a result, they prefer to predict a TM with short range dependency.

Figs. 5(a) and 5(b) describe the SREs of two networks, and the definition of SRE is as follows:

$$error_{SRE}(n) = \frac{\left\|\hat{X}(n,t) - X(n,t)\right\|_2}{\left\|X(n,t)\right\|_2}, \tag{16}$$

where $error_{SRE}(n)$ is spatial relative error, and $n = 1, 2, ..., N^2$, $t = 1, 2, ..., T$. Notation $\|\cdot\|_2$ denotes the ℓ_2 norm. In Figs. 5(a) and 5(b), the x-axis is the IDs of OD flows, which are arranged from small to large according to the mean values of OD flows. The y-axis is the SRE. From Figs. 5(a) and 5(b), SRMF has a larger SRE than the other two methods. The SREs of PCA and our method are relatively low. However, our method show some fluctuations in SRE. Figs. 5(c) and 5(d) describe the CDFs of SREs for two networks. It can be seen from the results of the CDF in the two figures that the SREs of PCA and our method are small, while the SRE of SRMF is larger. The mean values of SREs of our method, PCA, and SRMF in the Abilene network are 0.54, 0.62, and 0.61, respectively. Meanwhile, they are 0.26, 0.38, and 0.36 for the



(a) SREs in Abilene.

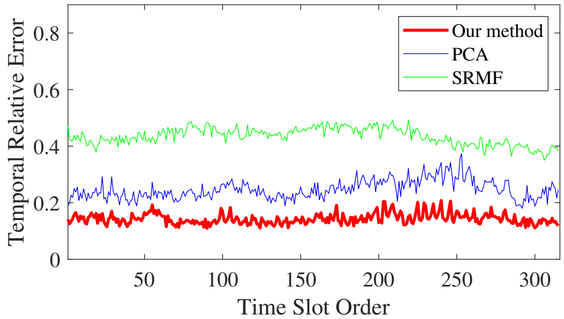
(b) SREs in the IIoT network.

(c) CDF of SREs in Abilene.

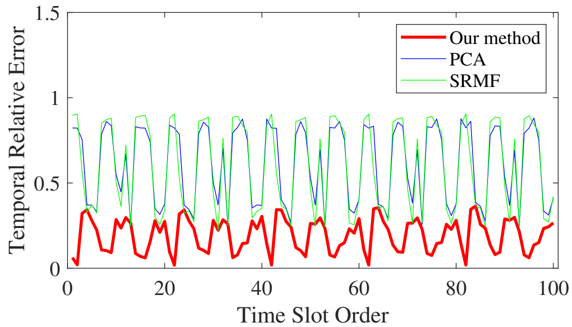
(d) CDF of SREs in the IIoT network.

Fig. 5. SREs and TREs in Abilene and IIoT networks.

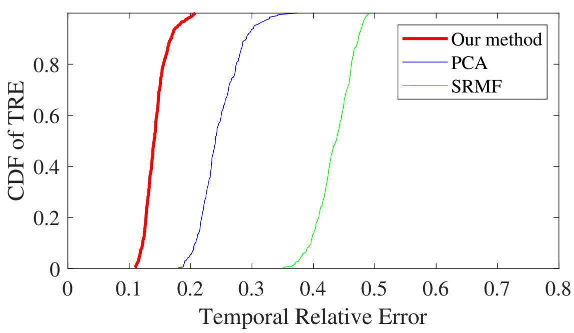
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3141743, IEEE Transactions on Industrial Informatics



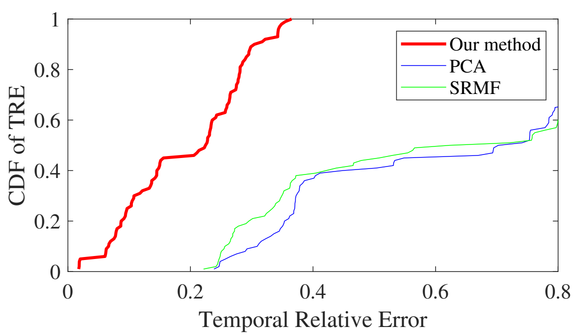
(a) TREs in Abilene.



(b) TREs in the IIoT network.



(c) CDF of TREs in Abilene.



(d) CDF of TREs in the IIoT network.

Fig. 6. CDFs of SREs and TREs in Abilene and IIoT networks.

IIoT network traffic prediction, respectively.

Figs. 6(a) and 6(b) display the TRE of two networks, and the TRE can be denoted as:

$$error_{TRE}(t) = \frac{\left\|\hat{X}(n,t) - X(n,t)\right\|_2}{\left\|X(n,t)\right\|_2}. \tag{17}$$

As shown by Fig. 6(a), for the Abilene network, the mean TRE of our method is smaller than the other two methods. However, the proposed algorithm has high jitter performance in TRE. In Fig. 6(b), for the IIoT network, PCA has a larger prediction error. Our method has less TRE than the other two methods. The traffic flows of IIoT networks have much more fluctuations, thus the principal component of TM changes frequently. As a result, the PCA method shows the largest TRE compared with the other methods. Moreover, the mean values of SREs of three methods in the Abilene network are 0.14, 0.24, and 0.44, respectively. Besides, they are 0.19, 0.61, and 0.60 in the IIoT Network, respectively. Figs. 6(c) and 6(d) describe the CDFs of TREs in Abilene and IIoT networks, respectively. It can be seen from the results of the CDF in the Fig. 6(c) that the TRE of SRMF is larger than the other two methods, while the proposed method has the smallest TRE. Meanwhile, we can obtain the same conclusion according to Fig. 6(d).

VI. Conclusion and Future Works

In this paper, we design a multi-task learning-based approach to predict the SDN-enabled IIoT network traffic, according to the spatio-temporal features of traffic flows. Aiming at the spatio-temporal features of TM in SDN-enabled IIoT networks, we design a DL architecture for traffic prediction, which has the advantages of high real-time and accuracy. We compare the proposed method with two existing network traffic prediction algorithms. The results show that the overall deviation of the proposed approach is less than the comparison algorithms, and it also has satisfied prediction effectiveness.

With the development of the SDN-enabled IIoT network, it is becoming a large-scale heterogeneous network. The memory and computing resources of IIoT nodes are limited, such as the nodes employed on vehicles. Hence, the training algorithm using a small number of samplings deserves to be considered, because limited nodes cannot provide the sufficient number of prior of network traffic for the training algorithm.

VII. Acknowledgements

This work was supported by the National Key R&D Program of China 2019YFC1511300, by the National Natural Science Foundation of China under Grants 61931019, 61971084, and 62171378, and by the Dalian Young Science and Technology Star 2020RQ002.

References

[1] C.-H. Wang, J.-J. Kuo, D.-N. Yang, and W.-T. Chen, “Collaborative Social Internet of Things in Mobile Edge Networks,” *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 473–11 491, 2020.

[2] G. Han, L. Liu, J. Jiang, L. Shu, and G. Hancke, “Analysis of Energy-Efficient Connected Target Coverage Algorithms for Industrial Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 135–143, 2017.

[3] W. Liang, W. Huang, J. Long, K. Zhang, K.-C. Li, and D. Zhang, “Deep Reinforcement Learning for Resource Protection and Real-Time Detection in IoT Environment,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[4] K. Bao, J. D. Matyjas, F. Hu, and S. Kumar, "Intelligent Software-Defined Mesh Networks With Link-Failure Adaptive Traffic Balancing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 266–276, 2018.

[5] Y. Cao, R. Wang, M. Chen, and A. Barnawi, "AI Agent in Software-Defined Network: Agent-Based Network Service Prediction and Wireless Resource Scheduling Optimization," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5816–5826, 2020.

[6] Y. Bi, G. Han, C. Lin, Y. Peng, H. Pu, and Y. Jia, "Intelligent Quality of Service Aware Traffic Forwarding for Software-Defined Networking/Open Shortest Path First Hybrid Industrial Internet," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1395–1405, 2020.

[7] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[8] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, "The Learning and Prediction of Application-Level Traffic Data in Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3899–3912, 2017.

[9] X. Song, Y. Guo, N. Li, and L. Zhang, "Online Traffic Flow Prediction for Edge Computing-Enhanced Autonomous and Connected Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2101–2111, 2021.

[10] A. Lazaris and V. K. Prasanna, "An LSTM Framework for Software-Defined Measurement," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 855–869, 2021.

[11] N. A. S. Al-Jamali and H. S. Al-Raweshidy, "Intelligent Traffic Management and Load Balance Based on Spike ISDN-IoT," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1640–1651, 2021.

[12] T. Liu, S. Chen, S. Liang, S. Gan, and C. J. Harris, "Fast Adaptive Gradient RBF Networks For Online Learning of Nonstationary Time Series," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2015–2030, 2020.

[13] H. Liu, L. T. Yang, J. Chen, M. Ye, J. Ding, and L. Kuang, "Multivariate Multi-Order Markov Multi-Modal Prediction With Its Applications in Network Traffic Management," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 828–841, 2019.

[14] Y. Li, J. Wang, X. Sun, Z. Li, M. Liu, and G. Gui, "Smoothing-Aided Support Vector Machine Based Nonstationary Video Traffic Prediction Towards B5G Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7493–7502, 2020.

[15] B. A. Al-Maytami, P. Fan, A. Hussain, T. Baker, and P. Liatsis, "A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing," *IEEE Access*, vol. 7, pp. 160 916–160 926, 2019.

[16] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic Flow Prediction Based on Deep Learning in Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3776–3789, 2021.

[17] F. Morales, M. Ruiz, L. Gifre, L. M. Contreras, V. Lopez, and L. Velasco, "Virtual network topology adaptability based on data analytics for traffic prediction," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A35–A45, 2017.

[18] L. Nie, Z. Ning, M. S. Obaidat, B. Sadoun, H. Wang, S. Li, L. Guo, and G. Wang, "A Reinforcement Learning-Based Network Traffic Prediction Mechanism in Intelligent Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2169–2180, 2021.

[19] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep Spatial–Temporal 3D Convolutional Neural Networks for Traffic Data Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3913–3926, 2019.

[20] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Preemptive SDN Load Balancing With Machine Learning for Delay Sensitive Applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 947–15 963, 2020.

[21] C. Hardegen, B. Pfülb, S. Rieger, and A. Gepperth, "Predicting Network Flow Characteristics Using Deep Learning and Real-World Network Traffic," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2662–2676, 2020.

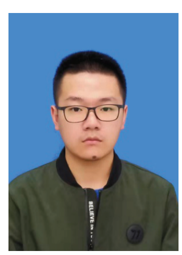
[22] A. Anirudh Sabnis, R. K. Sitaraman, and D. Towsley, "OCCAM: An Optimization Based Approach to Network Inference," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 2, p. 36–38, Jan. 2019.

[23] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

Shupeng Wang received the M.S. and Ph.D. degrees both in computer architecture from Harbin Institute of Technology, Harbin, China, in 2004 and 2007, respectively. He is currently a Senior Engineer with the Institute of Information Engineering, Chinese Academy of Sciences (CAS), Beijing, China. His research interests include big data management and analytics, network storage, etc.

Laisen Nie received his Ph.D. degree in communication and information system from Northeastern University, Shenyang, China, in 2016. Now He is an Associate Professor in School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include network measurement, network security, and cognitive networks.

Guojun Li received the M.S. and Ph.D. degrees both in communications and information system from Chongqing University of Posts and Telecommunications and Chongqing University in 2009 and 2012, respectively. He is now the director of the Lab of BLOS Trusted Information Transmission in Chongqing University of Posts and Telecommunications. His research efforts have been associated with beyond line of sight wireless communications and networks.



Yixuan Wu was born in Zhejiang, China, in 1994. He received the B. Eng. degree in optoelectronic information science and engineering from Ningbo University, Ningbo, China, in 2017. He is a graduate student in School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include network measurement and anomaly detection.

Zhaolong Ning (M'14-SM'18) received the M.S. and Ph. D. degrees both in communications and information system from Northeastern University, China in 2011 and 2014, respectively. He was a Research Assistant at Kyushu University from 2013 to 2014, Japan. He is an associate professor with the School of Software, Dalian University of Technology, Dalian, China. His research interests include Internet of things, mobile edge computing, deep learning, and resource management. He has published over 120 scientific papers in international journals and conferences.