Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2353875, 12 pages
https://doi.org/10.1155/2021/2353875

Research Article

Network Traffic Prediction via Deep Graph-Sequence Spatiotemporal Modeling Based on Mobile Virtual Reality Technology

Kai Zhang,[1,2] Xiaohu Zhao,[1] Xiao Li,[1,2] XingYi You,[1,2] and Yonghong Zhu[3]

[1]*National and Local Joint Engineering Laboratory of Internet Application Technology on Mine, China University of Mining and Technology, Xuzhou, Jiangsu 221008, China*
[2]*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221008, China*
[3]*School of Information Engineering, Xuzhou University of Technology, Xuzhou, Jiangsu 221008, China*

Correspondence should be addressed to Xiaohu Zhao; xhzhao76@gmail.com

Received 8 June 2021; Revised 25 June 2021; Accepted 12 July 2021; Published 1 August 2021

Academic Editor: Balakrishnan Nagaraj

Accurate and real-time network traffic flow forecast holds an important role for network management. Especially at present, virtual reality (VR), artificial intelligence (AI), vehicle-to-everything (V2X), and other technologies are closely combined through the mobile network, which greatly increases the human-computer interaction activities. At the same time, it requires high-throughput, low delay, and high reliable service guarantee. In order to achieve ondemand real-time high-quality network service, we must accurately grasp the dynamic changes of network traffic. However, due to the increase of client mobility and application behavior diversity, the complexity and dynamics of network traffic in the temporal domain and the spatial domain increase sharply. To accurate capture the spatiotemporal features, we propose the spatial-temporal graph convolution gated recurrent unit (GC-GRU) model, which integrates the graph convolutional network (GCN) and the gated recurrent unit (GRU) together. In this model, the GCN structure could handle the spatial features of traffic flow with network topology, and the GRU is used to further process spatiotemporal features. Experiments show that the GC-GRU model has better prediction performance than other baseline models and can obtain spatial-temporal correlation in traffic lows better.

1. Introduction

Accurate network traffic prediction is the basis of network performance optimization and network integrated management [1]. The prediction results can be used for traffic engineering, anomaly detection, and energy consumption management [2, 3]. Especially in the last decade, the complexity and the diversity of the network and the communication scenario increase dramatically, which promote researchers proposed many technologies such as ultra-dense deployment of cellular cells, device-to-device (D2D) network technology mobile virtual reality (MVR), and mobile edge computing to improve network capacity and service quality [4–6]. The successful application of these technologies is tightly related to the accurate cognition of network traffic features and future trends. In the context of these scenarios, network resource optimization is accomplished by two key elements, traffic model and optimization algorithm. If we can more accurately and timely understand the dynamic trend of future traffic, high reliability and low delay communication could be achieved by dynamic content caching or service processing in vicinity [7–9].

However, the traditional network traffic prediction model gradually presents many deficiencies in processing of increasingly complex network traffic. In the context of the liner network traffic forecasting model such as autoregressive integrated moving average (ARIMA) model [10], support vector regression model (SVR) [11], and Bayesian model [12, 13], mainly through the linear correlation of traffic in temporal to achieve traffic modeling and prediction, with the emergence of new network technology and new scenarios, new factors that could greatly affect network traffic features should be considered:

(1) The trend of network traffic is constrained by network topology, every end-to-end traffic flow travels the network through a path composed of links, and a link converges flows from different paths. The performance change of any node on the transmission link will spread to the adjacent nodes. The booming 5G, the Internet of Things (IoT), MVR, and V2X bring more and more mobility to the network, which also bring great uncertainty in the spatial distribution of network traffic

(2) The statistical value of network traffic flow shows the characteristics of proximity, periodicity, and trend in time. The closer the time interval, the higher the correlation between the two statistical values. In the scenarios such as MVR, IoT, or V2X, the terminal behaviors and traffic demands are more dynamic, that make the temporal patterns of the traffic too complicated to be captured [14–16].

To better handle the problem of spatial feature processing caused by terminal mobility and spatial dependence of applications, many studies use neural networks to network traffic modeling and prediction based on the big data of the network traffic [17]. The convolutional neural network (CNN) and RNN are the most representative in the proposals [17–19]. However, the method based on CNN also has some shortcomings. Different from Euclidean data such as image and grid, the communication network usually uses graph representation which is not suitable for CNN [17]. Unlike the Euclidean data, the structure of graph is irregular. It is difficult for CNN to hold a stable kernel size to realize parameter sharing. This kind of model cannot describe the spatial correlation of network in essence; consequently, the prediction accuracy of the trained model and the applicability of different topologies are limited. To solve these problems, we propose GC-GRU, a novel network model could which could accurately forecast network traffic matrix by comprehensively understand the complex relation on traffic flow, node topology, and time. Our main contribution has three aspects:

(1) This model firstly uses GCN to process network topology relationship, node input traffic, and network topology that are combined as input of the model to achieve spatial features capturing. Then, hidden states are input to a recursive network based on GRU to find the temporal dynamics. Through the parameter sharing method of spatial and temporal features, traffic prediction has a better universal correlation model for the future network. Our model is able to generalize over arbitrary topologies, routing schemes, and variable traffic intensity

(2) We evaluated our model on real topology and traffic matrices of real traffic dataset. The result shows that our model can handle the time-varying information of graph structure well and has higher accuracy than all baseline methods, which proves the superiority of the model in network traffic prediction. In addition, through experimental comparison, we find that our GC-GRU model can maintain stable prediction performance under different forecast time granularity, and it shows that it has better robustness

2. Related

Network traffic prediction methods include analytic model method and data-driven method. The representative methods of the first kind include queuing theory model, cell transmission model, and random geometry model [20–23]. Through the model method, a clear relationship between network traffic and other network parameters could be established, so as to complete the network planning and scheduling. For example, Krishnan et al. [23] introduce an analysis framework of random geometry that describes the spatial-temporal interference of adjacent locations and calculates the joint coverage probability of them; in this contest, the Poisson process is used to model the mobile behavior of the UE under the station. Kamath et al. [24] proposed a framework to handle the QoS of heterogeneous services in the network, in which a multiclass queuing model is used to analyze the heterogeneous services performance demands and combined with SDN technology to complete service classification and network slicing. Such models are either based on some idealized assumptions or based on realistic simplification. However, in reality, it is difficult to model UE arrival process as Poisson process, and the diversity of applications also makes the link service feature dynamic. These theoretical models are difficult to model these mobility, time-varying, and spatial dependence elements comprehensively; consequently, it is hard to accurately predict the current complex network traffic in reality [10, 18]. In addition, the model-driven approach relies on specific idealized scenarios; as a result, it lacks the ability of generalization and migration [25–27].

The data-driven methods are based on the statistical characteristics of network traffic history data, by using its self-similarity, long-term relevance, and periodicity to forecast the trend in time domain [1, 28]. This kind of methods does not specifically analyze the specific dynamics and behaviors of network elements and has high flexibility. These methods are mainly including parametric prediction and nonparametric prediction [10, 29]. The parameter model is based on the regression function, through the analysis in mathematical statistics of the history data to determine the quantitative relationship of interdependence between two or more parameters, so as to forecast next step traffic volume according to the regression function.

More representative methods include autoregressive integral moving average model (ARIMA) and its improved variants [10, 30, 31]. For example, Paxson and Sally [32] used the autoregressive integral moving average model (ARIMA) to predict Ethernet traffic in 1994. Subsequently, in order to improve the prediction accuracy of the model, many researchers proposed different kinds of improvements based on it, such as seasonal-ARIMA (SARIMA), Kohonen-ARIMA, and subset ARIMA [10, 33]. Maciej and Anna [33] found that the latter SARIMA has better prediction accuracy than the ARIMA model through experimental. Furthermore, Chen et al. [34] used SARIMA to predict the short-term flow of the IEEE 802.11 network for the first time and achieved well prediction results on single traffic flow. Their advantages are that the mathematical foundations are relatively mature, and the system performance evaluation is easy to express mathematically. However, these methods are not robust. They assume that statistics and parameters' relationship of the traffic model hold steady which are not conform to the reality of network traffic nowadays. In addition, the nonparametric model has its advantages in this respect. The representative methods such as Bayesian network model [12, 13], support vector regression model (SVR) [35, 36], K-Nearest Neighbor model [37], and neural network model [38] use just historical traffic information to automatically learn statistics.

Recently, the deep learning model has been effectively devoted to promote complex pattern recognition and analysis in big data systems [28, 39–41]. Meanwhile, with the improvement of network big data collection ability, network traffic prediction methods based on the deep learning have attracted much attention because of its ability on capturing the nonlinear features of network traffic [1, 41]. For example, Azzouni and Pujolle [18] use recurrent neural networks (RNNs) to predict the traffic matrix (TM) and uses the prediction results to dynamically and actively allocate optical network resources to save network available capacity and alleviate the impact of peak traffic. To achieve traffic forecasting in data center, Laisen et al. [13] combined deep structure and deep trust networks to achieve traffic demand prediction through temporal dynamic features capturing. Tang et al. [42] proposed a prediction method of communication key indicators based on deep learning and combined software-defined network (SDN) to achieve intelligent dynamic channel allocation, which improves the channel utilization of wireless Internet of Things and avoids potential congestion. These methods mainly take the temporal correlation of network traffic sequence as prediction basis and the advantages recurrent neural network (RNN) structure to learn the characteristics in time to achieve adaptive traffic flow or traffic matrix prediction.

As mentioned before, the factors that affect network traffic are not only time but also in spatial. Without spatial relations such as network topology, dynamic content dependence, and client mobility, accurate forecasting and estimating could not be achieve. In order to explore the potential of spatial features on network traffic prediction, many studies had been proposed. For example, Zhao et al. [43] proposed a model that combined the multiscale wavelet analysis with the deep learning approach with; firstly, discrete wavelet decomposition was used to decompose the original TM sequence into multilevel time-frequency TM subsequences of different time scales; then, a CNN was used to find the spatial distribution pattern between network streams and finally, the LSTM in charge of exploring temporal dynamics in the TM sequences. In order to estimate the future location and resource usage of UEs, Siracusano and La Corte [44] proposed a deep regression (DR) model based on a tightly connected convolutional neural network (CNN) to simulate its complex spatiotemporal dynamics to capture the multiscale of mobile data. LA-ResNet [41] is a model which combined the residual network and RNN to capture spatiotemporal correlations of wireless traffic, and the experimental results showed that it had higher accuracy than RNN models.

Although the CNN method has made great progress in network traffic prediction, its data operation process makes it more suitable for processing Euclidean date rather than graph topology. The complex topological structures of communication network limited its application in this field. Based on this background, we propose a novel network GC-GRU, which unifies the network nodes, links, and traffic input in space, and analyzes the inherent spatiotemporal correlation characteristics combined with the dynamics in time, so as to complete the accurate network traffic prediction.

3. Method

3.1. Problem Definition. The task of network traffic prediction is to estimate the future network traffic status as accurately as possible according to the network historical traffic data. The traffic information includes traffic volume, time delay, and packets loss. In this paper, we take traffic volume as main example in experiment.

Definition 1. In our model, unweighted graph $\mathrm{G} = (V, E)$ was used to denote the network topology, where $V = \{v_1, v_2, \cdots, v_N\}$ denotes N nodes (they could be switches or routers) in the network, and E represents the set of communication links between nodes. The adjacency matrix A describes the detail relationship among network nodes, where $A \in R^{N\times N}$. Its element $a_{ij} = 1$ if node v_i is connected to v_j, $a_{ij} = 0$ otherwise.

Definition 2. We use $X^{N\times P}$ to represent the traffic feature matrix in the network, where N represents nodes counts as mentioned above, and P denotes the length of historical sampling sequence of traffic. $X_t \in R^{N\times i}$ means the traffic on node connection at time i.

In this way, our spatiotemporal network traffic prediction problem model is represented as equation (1):

$$[x_{t+1}, \cdots, x_{t+\mathrm{T}}] = f([x_{t-n}, \cdots, x_{t-1}, x_t]\,; G), \tag{1}$$

where f denotes the mapping function from G and historical X, to the next T step traffic information.

3.2. Model

3.2.1. Framework Overview. In this part, we will introduce our solution model GC-GRU, in which the temporal and spatial features are considered comprehensively. As shown in Figure 1, GC-GRU includes input layer, spatial feature processing, temporal feature processing, and a fully connected layer. Firstly, it takes n time series data between

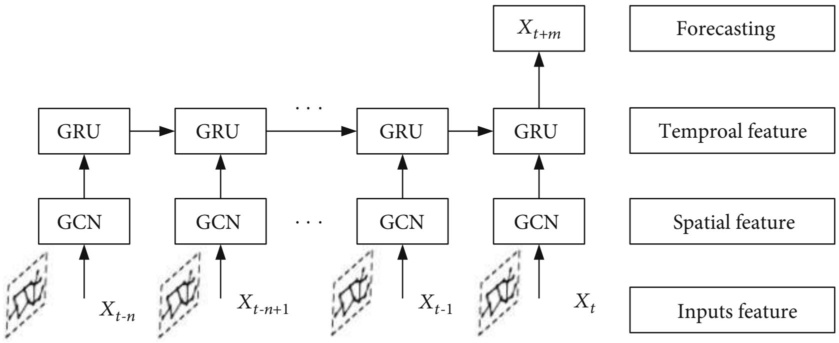


Figure 1: Framework overview of the GC-GRU model.

nodes in the network as the input and use graph convolution combined with the spatial topological relationship of communication network to obtain the spatial features of traffic; secondly, we use the processed time series with spatial features as the input of gated recursive unitary model and obtain the features in time domain through the information transmission between cells; finally, we use all connected network as the input. The prediction results are obtained through the layer.

As in Figure 1 shows, our model contains two main part: network topology GCN for spatial feature extraction and GRU for hidden spatial-temporal features capturing. Firstly, the GCN is used to lean the spatial features of the network topology, it takes the traffic feature matrix and network topology data both as input, in detail, X_t denotes the traffic on all node connections at time i, and the topology data is represented by adjacency matrix A introduced in Definition 1. Then, the output of the GCN model is fed into the second part which is constructed by gated recurrent units, to find out the traffic dynamics, so as to lean the temporal features. At last, a fully connected layer network is used to convert features mapping to original space; then, we get the predicted results.

3.2.2. Spatial Relation Feature Extraction Model. Accurately capturing the spatial dependence between nodes in the network is a key problem in this paper. As mentioned before, the CNN is suitable for Euclidean space data rather than non-Euclidean data like network topology [45]. In the graph representation of the network, each vertex has different numbers of adjacent vertices, and it is difficult to hold a convolution kernel with a same size. In recent years, the graph convolutional network (GCN) has been widely concerned, which can realize convolution operation on non-Euclidean structure, effectively learn the spatial characteristic information of nodes, and has been applied in action recognition, traffic network flow prediction, and so on [46–48].

From the perspective of graph signal processing (GSP), it applies discrete signal processing (DSP) to the field of graph signal. Through the transfer of basic concepts of signal processing such as Fourier transform and filtering, it studies the basic tasks of signal processing such as compression, transformation, and reconstruction of graph signal [49, 50]. In detail, given a graph $G = (V, E)$, graph signal is a kind of mapping from node field to real number field [50], which is expressed as vector in the form of $X = [x_1, x_2, \cdots, x_n]^T$, and x_i represents signal strength on vertex v_i strength. The Fourier transform of X is represented as $\mathcal{F}(X) = U^T X$, where $U = [u_0, u_1, \cdots, u_{n-1}] \in R^{n\times n}$ represented the eigenvector. Its inverse Fourier transform is $\mathcal{F}^{-1}(\widehat{X}) = U\widehat{X}$, where $\widehat{X}$ represents the frequency domain signal obtained from the graph Fourier transform [46]. The graph convolution operation of two groups of graph signals can be transformed into the corresponding graph filtering operation. From this point of view, the graph convolution is equivalent to the graph filtering. We take X as signal on vertexes and $g \in R^N$ as the filter, and the graph convolution could be expressed as equation (2):

$$x *_G g = \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(g)) = U\left(U^T x \odot U^T g\right), \qquad (2)$$

where $\odot$ represents the Hadamard product [46]. If we denote a filter as $g_\theta = \operatorname{diag}\left(U^T g\right)$, equation (2) could be simplified as

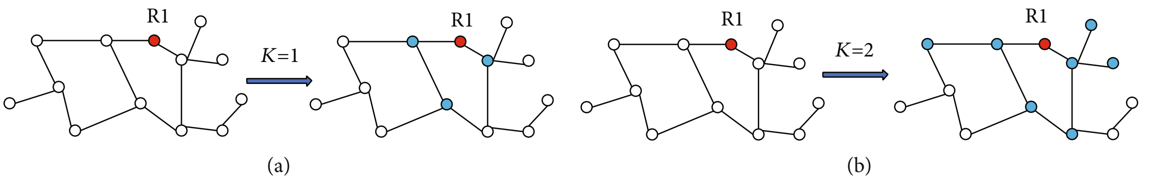
$$*_{\mathcal{G}} g_\theta = U g_\theta U^T X. \qquad (3)$$

This method was first proposed by [51], but it has many limitations in practical application. Firstly, the computation depends on the eigen decomposition of Laplacian matrix, which has high complexity. Secondly, when u is used for signal transformation, because u is dense, the complexity is very high, and the method is not localized. Therefore, the method has not attracted widespread attention.

In order to reduce the computational complexity, Chebyshev Spectral CNN (ChebNet) [45] used Chebyshev polynomials of the diagonal matrix of eigenvalues to approximate the filter g_θ. g_θ is a diagonal matrix composed of the parameter θ, and its polynomial expression is represented as equation (4):

$$g_\theta\left(\tilde{L}\right) = \theta_0 I + \theta_1 \tilde{L} + \theta_2 \tilde{L}^2 + \cdots + \theta_k \tilde{L}^k, \qquad (4)$$

where $\tilde{L} = I_n - D^{-1/2} A D^{-1/2}$, D denotes the degree matrix of A ($D_{ii} = \sum_j A_{i,j}$), and I_n denotes identity matrix. As the normalized Laplacian matrix has the property of real symmetric-positive semidefinite [51], $\tilde{L}$ can be decomposed into $\tilde{L} = U \Lambda U^T$, where Λ is the diagonal matrix of eigenvalues [50]



(a)

(b)

Figure 2: Graph convolutional networks. (a) $k = 1$. (b) $k = 2$.

$(\Lambda_{ii} = \lambda_i)$. Then, g_θ could expressed as a function of eigenvalue Λ:

$$g_\theta = g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k, \tag{5}$$

and then convolution of g_θ and graph signals could expressed as

$$X *_G g_\theta = U g_\theta(\Lambda) U^T x = \sum_{k=0}^{k-1} \theta_k L^k x. \tag{6}$$

In this way, the number of calculation parameters is reduced from n to k, and there is no need to explicitly calculate the feature matrix U in the calculation process; so, the computational complexity is greatly reduced. In addition, k-localized has local connectivity, because it is a k -order polynomial in Laplacian. As Figure 2 shows, each point represents a network node (router or switch), and the line represents link between nodes; for node R1, the GCN could get the spatial information directly linked to R1 and their links to other surrounding nodes and encode the network topology and the traffic attributes in nodes, so as to obtain the spatial correlation.

On the basis above, the first order approximation of chebNet is introduced into the graph convolution network to semplice the calculation process [45]. Moreover, in order to reduce parameters, Kipf [52] further assumes that $\theta = \theta_0 = -\theta_1$, the definition of graph convolution given as formula (7):

$$*_{\mathcal{G}} g_\theta = \theta\left(I_n + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)x. \tag{7}$$

Then, a high availability graph convolution network without too much matrix multiplication operation is established. It can be expressed as

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \hat{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \theta^{(l)}\right), \tag{8}$$

where σ denotes the sigmoid function, $\tilde{A} = A + I_n$, $\tilde{D}(\tilde{D} = \sum_j \tilde{A}_{ij})$ represents degree matrix, $H^{(l)}$ is the output of layer l, and $\theta^{(l)}$ denotes the parameters corresponding.

As shown in Figure 3, our spatial feature analysis model uses a two-layer graph convolution [52] structure. The first layer is $H^{(1)} = \sigma(\hat{A} X \theta_0), \theta_0 \in \mathscr{R}^{n_1 \times p}$, where n_1 is the number of hidden layer cells, and p is the length of input traffic matrices. $\theta_0 \in \mathscr{R}^{n_1 \times p}$ represents the feature matrix from the hidden layer to the output layer. The second layer is $H^{(2)} = \sigma(\hat{A} H^{(1)} \theta_1), \theta_1 \in \mathscr{R}^{N \times T}$, where N represents the node number, and T represents the length of the sequence to be predicted. The whole graph feature extraction process is expressed as follows:

$$f(X, A) = \sigma\left(\hat{A} \sigma\left(\hat{A} X \theta_0\right) \theta_1\right). \tag{9}$$

3.2.3. Temporal Dependence Modeling. This section introduces the method to obtain temporal correlation in network traffic prediction. At present, RNN is the most used model for time sequence data processing. In order to calculate the gradient, the time reverse propagation algorithm is used in the training RNN network. However, it has the defects of gradient disappearance and gradient explosion [53]. For network traffic prediction, it may lose the ability of weighted long-term dependence, which makes it having limitations in long-term prediction. Two famous architectures proposed to solve this problem are LSTM and GRU,and they are improved variants of RNN, which have been proved to solve the above problems [18, 28, 54]. Here, we take the GRU model to find the temporal correlation of network traffic data which is relatively simple, because of less parameters and faster training speed.

As Figure 4 shows, x_t represents the input at time t; h_{t-1} represents the hidden state at time $t-1$. The candidate activation $\tilde{\mathrm{h}}_\mathrm{t}$ contains the previous information h_{t-1} and input $\mathrm{x_t}$; h_t denotes output state at time t which is a liner combination of the activation on the h_{t-1} and the candidate activation $\tilde{h}_t$; r_t is the reset gate, which is in charge of controlling whether the previous state should be forgotten or not (e.g., if $\mathrm{r_t} \approx 0$, h_{t-1} will not be passed to $\mathrm{h_t}$); z_t is the update gate, and it decides the extent to which the unit updates its activation (e.g., if $z_t \approx 1$, h_{t-1} almost directly copied h_t. Instead, if $z_t \approx 0$, $\tilde{h}_t$ will be passed directly to h_t)

3.2.4. Spatiotemporal Correlation Process. To better predict the impression of complex spatial dependence and dynamic time dependence on the distribution of network traffic, we devote a network traffic spatiotemporal correlation model. In this model, the gate structure and hidden state of GRU are reserved, but we use the convolution feature of graph as input to find the hidden spatiotemporal features.

In the model, the traffic matrix dynamics on network are associated with the network topology information through the graph product to find the spatial correlation of network traffic. Then, the GRU unit is used to further handle the dynamics in temporal, so as to get spatiotemporal

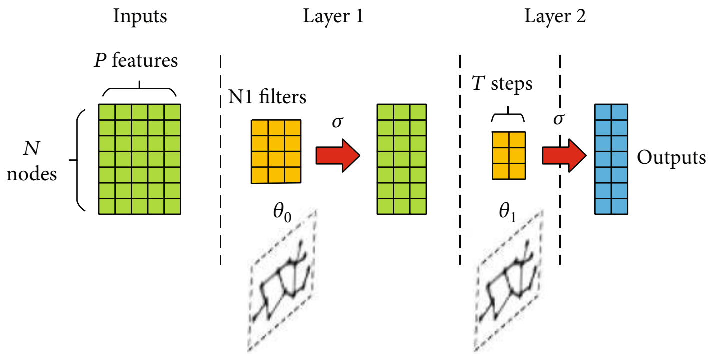


Figure 3: Spatial modeling in 2-layer graph convolutional networks (GCN).

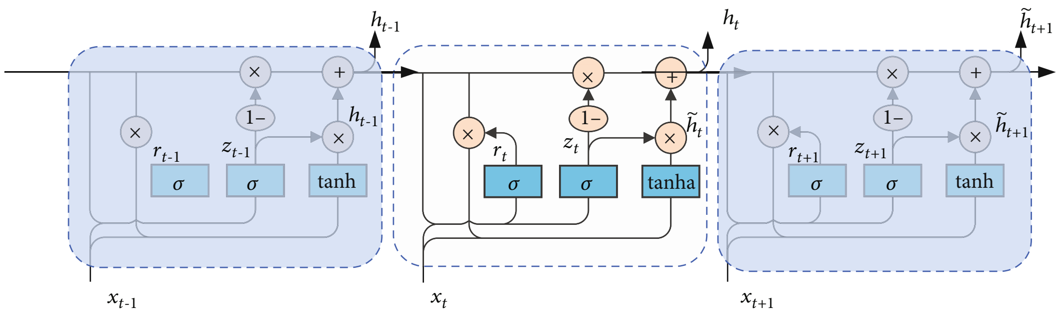


Figure 4: The structure of the GRU model.

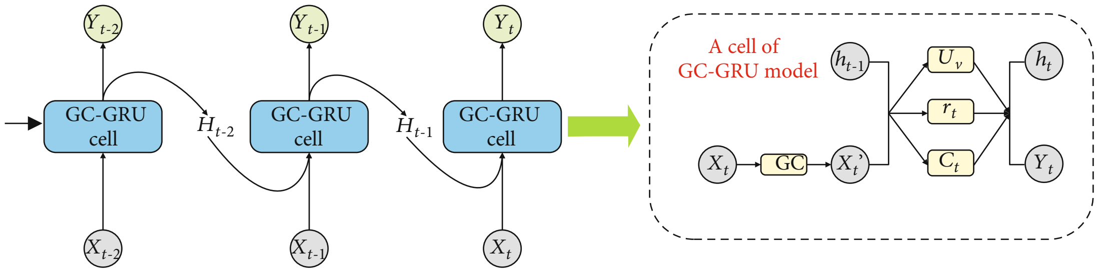


Figure 5: The process of spatial-temporal prediction.

correlation, and finally realizes the traffic prediction task. As Figure 5 shows, h_{t-1} denotes the output before time t, and h_t denotes the output at time t. X_t' is the spatial feature matrix after graph convolution, and GC represents the graph convolution process. z_t is the update gate, and r_t is the reset gate. The detail process is shown below:

$$\begin{aligned} r_t &= \sigma\left(W_r\left[X_t', h_{t-1}\right] + b_r\right), \\ z_t &= \sigma\left(W_z\left[X_t', h_{t-1}\right] + b_z\right), \\ \tilde{h}_t &= \tan h\left(W\left[X_t', r_t \odot h_{t-1}\right] + b\right), \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \end{aligned} \tag{10}$$

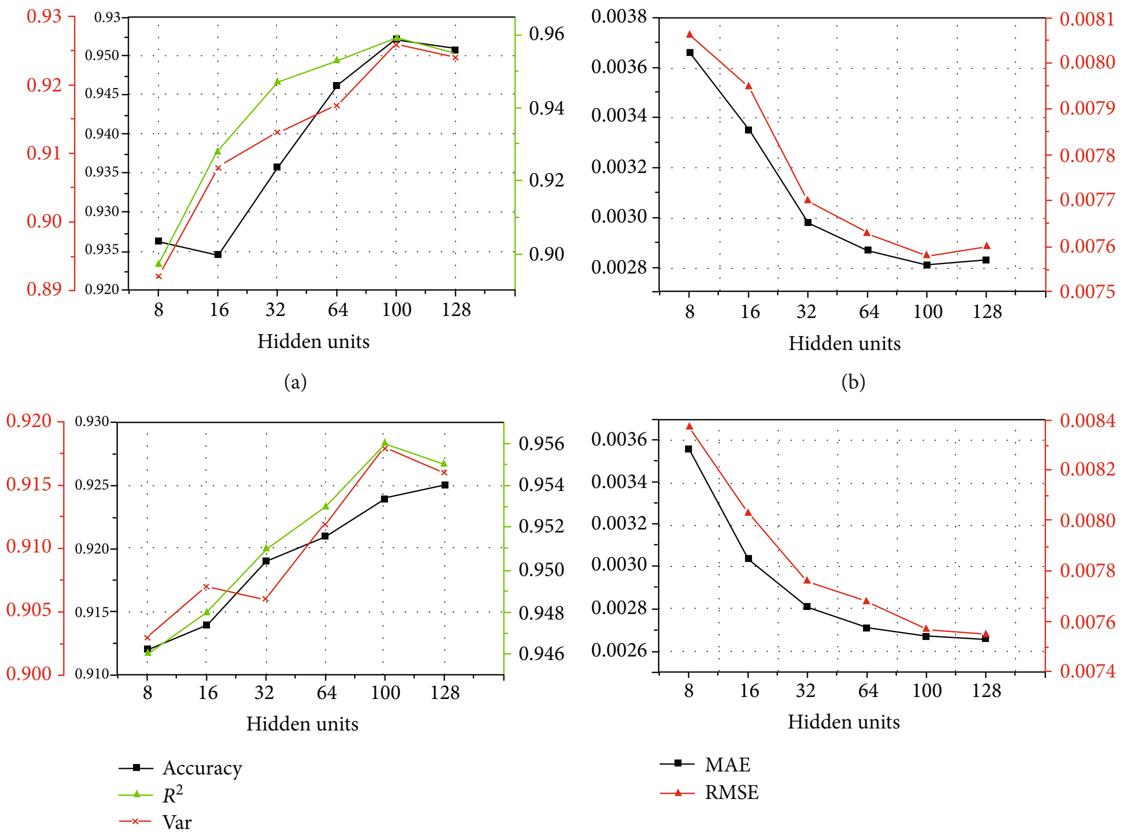
W represents the weights, and b represents biases, which needs to be trained. ⊙ represents the point-wise multiplication. The loss function is defined as

$$\text{Loss} = \left\|X_t - \widehat{X}_t\right\| + \lambda L_{\text{reg}}, \tag{11}$$

where X_t and $\widehat{X}_t$ represent the ground truth volume and the forecasted volume. λ is a hyperparameter, and λL_{reg} denotes the L2 regularization term to prevent overfitting.

4. Experiments

4.1. Data Description and Experimental Settings. We evaluate our prediction model by dataset from the Europe Research and Education Network (GÉANT) [55] . GÉANT



(a)

(b)

(c)

(d)

Figure 6: (a) Accuracy, R^2, and var in training set under different hidden units. (b) MAE and RMSE in the training set under different hidden units. (c) Accuracy, var, and R^2 in test set under different hidden units. (d) MAE and RMSE in the test set under different hidden units.

Table 1: Instructions of baseline models used for comparison.

No.	Model name	Brief intrude
1	ARIMA [10]	A well-known series data prediction model which the parameter model is consistent with the observation data.
2	SVR [11]	A version of support vector machine (SVM) for regression, it is mainly used in nonlinear regression estimation.
3	GCN	As mentioned in section 3.2.2
4	LSTM [38]	Long-short term memory (LSTM). Here, we set up two LSTM layers and a fully connected layer.

network contains 23 router nodes and 74 undirected links (as of 2005); in addition, 11460 traffic matrices were recorded in every 15 minutes. In the experiment, we build a sliding window on each data sequence for continuous input and learning. 70% of the data is used at the training phase, and the rest 30% is used at test phase.

All experiments adopt Pytorch deep learning architecture. The workstation used for experiment is configured with 64 core Intel Xeon 2.40GHz CPU, tow 8G NVIDIA GTX-2080 graphics, and 192 G RAM. The workstation uses the windows server operating system.

4.2. Evaluation Metrics. In the experiment, 5 metrics are used to evaluate the performance of GC-GRU model:

(1) Root mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{M}}\frac{1}{\mathrm{N}}\sum_{\mathrm{j}=0}^{M}\sum_{\mathrm{i}=0}^{N}\left(\mathrm{y}_{\mathrm{i}}^{\mathrm{j}} - \mathrm{y}\wedge_{\mathrm{i}}^{j}\right)^{2}}. \tag{12}$$

(2) Mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\frac{1}{\mathrm{M}}\sum_{\mathrm{j}=0}^{\mathrm{M}}\sum_{\mathrm{i}=0}^{N}\left|\mathrm{y}_{\mathrm{i}}^{\mathrm{j}} - \hat{y}_{\mathrm{i}}^{\mathrm{j}}\right|, \tag{13}$$

Table 2: The prediction results of the GC-GRU model and other baselines on GÉANT dataset.

Time	Metric	ARIMA	SVR	LSTM	GCN	GC-GRU
15 min	MAE	0.00742	0.00378	0.00317	0.00376	0.00279
	RMSE	0.0112	0.00898	0.00796	0.00833	0.00692
	Accuracy	0.4935	0.8337	0.8991	0.7635	0.9137
	R^2	*	0.8138	0.8536	0.7537	0.8811
30 min	MAE	0.00693	0.00366	0.00312	0.00384	0.00272
	RMSE	0.0098	0.00866	0.00762	0.0088	0.00708
	Accuracy	0.5375	0.8523	0.8903	0.7181	0.9113
	R^2	*	0.7835	0.8027	0.6957	0.8738
45 min	MAE	0.00726	0.00397	0.00299	0.00415	0.00259
	RMSE	0.01045	0.00857	0.00768	0.00855	0.00715
	Accuracy	0.5874	0.8497	0.8701	0.8499	0.8257
	R^2	*	0.7273	0.7745	0.6625	0.9023
60 min	MAE	0.00711	0.00341	0.00296	0.00409	0.00261
	RMSE	0.0976	0.00927	0.00748	0.00856	0.00712
	Accuracy	0.4973	0.8447	0.6894	0.5693	0.8932
	R^2	*	0.6438	0.6948	0.6204	0.8032

* means the value is very small to be ignored which indicates the model's prediction effect is poor.

where $\hat{y}_i^j$ denotes the forecasted traffic value of traffic flow i at time j, and y_i^j denotes ground truth value, respectively. N is the index of traffic flows, and M represents the index of time slot

(3) Accuracy:

$$\text{Accuracy} = 1 - \frac{\left\|\mathrm{Y} - \widehat{\mathrm{Y}}\right\|_{\mathrm{F}}}{\left\|\mathrm{Y}\right\|_{\mathrm{F}}}. \tag{14}$$

Y and $\widehat{Y}$ represent the set of real values and predicted values, respectively

(4) Coefficient of determination (R^2):

$$R^2 = 1 - \frac{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_i^j - y\wedge_i^j\right)^2}{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_i^j - \bar{Y}\right)^2}. \tag{15}$$

$\bar{Y}$ is the average of Y.

(5) Explained variance score (var):

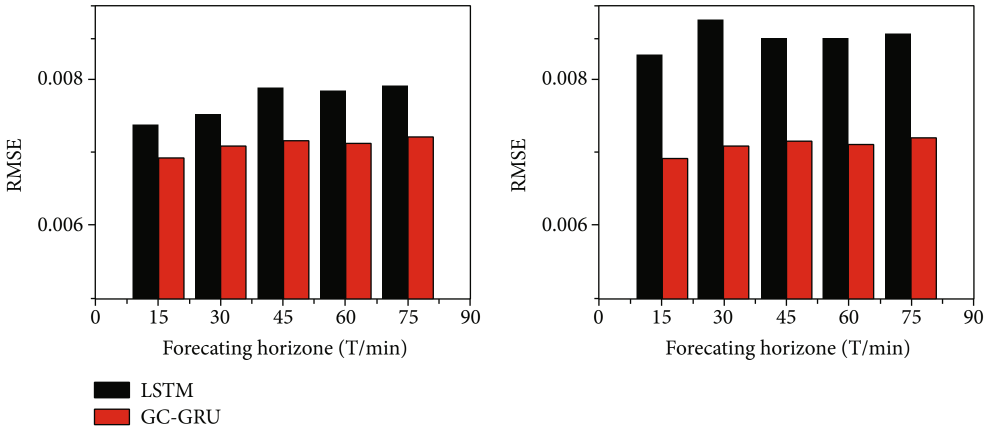
$$\text{var} = 1 - \frac{\mathrm{Var}\left\{Y - \widehat{Y}\right\}}{\mathrm{Var}\left\{Y\right\}}. \tag{16}$$

R^2 and var are the measures to represent fitting degree of the predicted value to observation value of the traffic.

4.3. Model Parameter Settings. In the experiment, the batch size is set to 32, and the training epoch is 200; the learning rate is set to 0.001. We set 8, 16, 32, 64, 100, and 128 as the number of hidden units and record and analyze the experimental results under each configuration. As shown in Figure 6(a), experiments show that the best prediction accuracy and correlation are obtained when the size of hidden units is 100. Specifically, with the increase of the number of hidden cells, the prediction accuracy, R^2, and var increase, but with the increase of hidden cells, the increase range of prediction accuracy and correlation declines, and when the number of hidden cells reaches 128, the prediction accuracy and R^2 drop. At the same time, Figure 6(b) shows the MAE and RMSE values under different units' numbers in the training stage. It can be seen that contrary to the accuracy and R^2, MAE and RMSE decrease with the increase of the number of hidden units, which indicates that the error between the predicted value and the observed value can be reduced by appropriately increasing the number of hidden units. Similarly, when the number of hidden units is 100, the error is the smallest. Figures 6(c) and 6(d) show the experimental results on the test set, which are basically consistent with the training phase. At the same time, when the number of hidden units is set to 100, several indicators perform best. In conclusion, the appropriate number of hidden units can improve the performance of the model, but not the higher the better, and too many hidden units may cause over fitting problem and led to higher computing overhead which would decrease prediction performance.

4.4. Results. We compared our GC-GRU model with other reprehensive baselines model, as Table 1 shows.

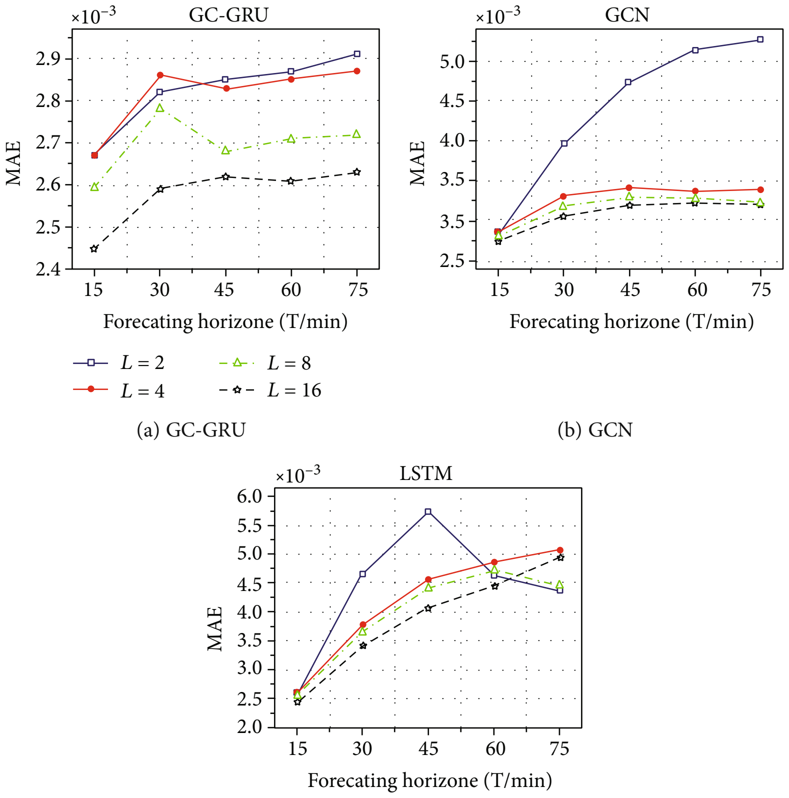
Based on the GÉANT dataset, we make traffic prediction at 15, 30, 45, and 60 minutes. As shown in Table 2, it is clearly observed that our GC-GRU model obtained the best



(a)

(b)

Figure 7: Spatiotemporal prediction capability of the GC-GRU model. (a) Comparison of the RMSE of GC-GRU model with the LSTM model. (b) Comparison of the RMSE of the GC-GRU model with the GCN model.



(a) GC-GRU

(b) GCN

(c) LSTM

Figure 8: MAE comparison between different sliding window lengths L for different perdition horizons.

performance under all evaluation metrics and different prediction time granularity. In general, the prediction performance of deep learning methods (LSTM, GC-GRU) is significantly better than other model baselines (ARIMA, SVR) under each prediction horizon. For example, compared with the ARIMA model, the prediction accuracy of the GC-GRU model and LSTM model is about 46% and 45% higher, and RMSE error is reduced by 38% and 29%, respectively. Also, compared with the SVR model, these metrics of the GC-GRU model and LSTM model are 22% and 11% lower, respectively, and the prediction accuracy is improved by 15% and 9%. Although the performance of the SVR model is better than ARIMA, it is also significantly lower than the LSTM and GC-GRU method. This may be caused by the lack of the ability of these methods to process complex nonstationary time series data. In essence, LSTM and GRU could capture the temporal features of traffic matric sequences than the time series model, because their cell states increase with time. Similarly, we can also clearly notice that the prediction effect of the GCN model is not good, which shows that it is not enough to only consider the spatial features of network traffic by ignoring the time-domain characteristics of network traffic.

4.4.1. Spatiotemporal Prediction Ability. In order to verify whether the GC-GRU model on the ability of handle spatiotemporal features from the dataset, we further analyzed the experimental results of the GC-GRU model, LSTM model, and GCN model. In detail, Figure 7(a) shows the RMSE of GC-GRU compared with the LSTM model which only considers the time domain features, the RMSE error of the GC-GRU model for 15-minute and 60-minute network traffic prediction is reduced by about 13% and 5%, respectively, and the prediction error is basically controlled at a low level, which indicates that the GC-GRU model can capture the time correlation well. In Figure 7(b), the multistep forecast result of GC-GRU is obviously better than the GCN model, and the RMSE of GC-GRU model is 16.9% lower than that of GCN model. For the 60-minute traffic volume forecast, the RMSE of GC-GRU model is reduced by 16.8%, which indicates that the GC-GRU model can capture the spatial correlation. It can be seen that GC-GRU has better prediction accuracy than the method based on spatial and temporal factors. Because GCN only considers the spatial feature of the network, the dramatic change of time domain may cause prediction error. LSTM can make a better one-step prediction because of its structure memory, but in the process of training, there is the possibility of modeling noise into the model, which leads to overfitting. In the GC-GRU model, we construct a local filter in the Fourier domain when dealing with the neighborhood node relationship and constantly move the filter to capture the spatial features, so that the local parameters can be shared. This process plays a role in noise suppression, which may reduce the possibility of overfitting issue.

4.4.2. Prediction Ability for Different Prediction Horizons. Figure 8 shows the difference of input sliding window lengths (L) for traffic prediction. Generally speaking, a longer input window length can improve the accuracy of long-term prediction because it contains more time-domain features. For the LSTM model, because of the memory of the hidden state, when the hidden state is updated for a long time, they will remember more information; so, the prediction accuracy is more affected by the history window. In contrast, the GCN model does not have explicit temporal modeling; consequently, it gets little responded to window size change. The MAE of our model for different perdition horizons under different history sliding windows holds small and stable, which means that the method has strong adaptability to both prediction horizon and input history horizon. In conclusion, we know that the GC-GRU model has strong adaptability in both long-term and short-term forecasting.

5. Conclusion

In view of the new characteristics of network traffic brought by the dynamic change of spatiotemporal correlation in the new application scenarios of mobile network, this paper proposes a novel network traffic prediction method based on traffic spatiotemporal features. It combines GCN and GRU, in which GCN is in charge of obtaining spatial correlation of the traffics on each node, and GRU is used to handle the spatiotemporal features hidden in spatial dynamic series, so as to achieve spatiotemporal network traffic prediction. Experimental results on real network traffic datasets show that the GC-GRU model has better prediction performance than other baseline models under different prediction horizons. It is not only more suitable for nonlinear network traffic prediction than ARIMA and SVR but also more suitable for nonlinear network traffic prediction than spatiotemporal model. The research results show that the model can capture the spatiotemporal correlation characteristics of network traffic, provide accurate network traffic distribution prediction for mobile network interactive applications, and provide more fine-grained and more accurate scheduling decision-making basis for wireless access management, content optimization cache, and computing resource scheduling tasks in these application scenarios [56].

In addition, we will further study the traffic characteristics of mobile networks. In reality, it is not only the traffic size but also the link delay, jitter, and other factors that reflect the network characteristics. In the future, we will explore the network demand forecasting method under multiparameter constraints.

Data Availability

All data, models, and code generated or used during the study appear in the submitted article.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This study was supported by (1) the Fundamental Research Funds for the Central Universities: 2020ZDPY0223 and supported by the Xuzhou Science and Technology Plan Funded Project-Research on Key Technologies for Three-dimensional Monitoring of Lake Water Environment based on 5G Internet of Things: KC19208.

References

[1] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, “A survey on big data for network traffic monitoring and analysis,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[2] P. Tsai, C. W. Tsai, C. W. Hsu, and C. S. Yang, "Network monitoring in software-defined networking: a review," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3958–3969, 2018.

[3] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "DeepTP: an end-to-end neural network for mobile cellular traffic prediction," *IEEE Network*, vol. 32, no. 6, pp. 108–115, 2018.

[4] B. K. J. al-Shammari, N. al-Aboody, and H. S. al-Raweshidy, "IoT traffic management and integration in the QoS supported network," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 352–370, 2018.

[5] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017.

[6] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): a review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144, pp. 17–39, 2018.

[7] B. Hu, L. Fang, X. Cheng, and L. Yang, "In-vehicle caching (IV-cache) via dynamic distributed storage relay (D^2SR) in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 843–855, 2019.

[8] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, and A. Nayyar, "SDN-based real-time urban traffic analysis in VANET environment," *Computer Communications*, vol. 149, pp. 162–175, 2020.

[9] C. Shu, Z. Zhao, G. Min, J. Hu, and J. Zhang, "Deploying network functions for multiaccess Edge-IoT with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9507–9516, 2020.

[10] R. Madan and P. S. Mangipudi, "Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–5, Noida, India, August 2018.

[11] Y. Liang and L. Qiu, "Network traffic prediction based on SVR improved by chaos theory and ant colony optimization," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 1, pp. 69–78, 2015.

[12] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Traffic estimation and optimal counting location without path enumeration using Bayesian networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 23, no. 3, pp. 189–207, 2008.

[13] L. Nie, D. Jiang, and Z. Lv, "Modeling network traffic for traffic matrix Estimation and anomaly detection based on Bayesian network in cloud computing networks," *Annals of Telecommunications*, vol. 72, no. 5-6, pp. 297–305, 2017.

[14] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions," *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.

[15] P. Duan, Y. Jia, L. Liang, J. Rodriguez, K. M. S. Huq, and G. Li, "Space-reserved cooperative caching in 5G heterogeneous networks for industrial IoT," *Journal of Network and Computer Applications*, vol. 14, no. 6, pp. 2715–2724, 2018.

[16] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: automation networks in the era of the Internet of Things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[17] X. Cheng, L. Fang, L. Yang, and S. Cui, "Mobile big data: the fuel for data-driven wireless," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1489–1516, 2017.

[18] A. Azzouni and G. Pujolle, "NeuTM: a neural network-based framework for traffic matrix prediction in SDN," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan, April 2018.

[19] X. Cheng, L. Fang, X. Hong, and L. Yang, "Exploiting mobile big data: sources, features, and applications," *IEEE Network*, vol. 31, no. 1, pp. 72–79, 2017.

[20] M. Kim, "Network traffic prediction based on INGARCH model," *Wireless Networks*, vol. 26, no. 8, pp. 6189–6202, 2020.

[21] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Performance modeling of Softwarized network services based on queuing theory with experimental validation," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1558–1573, 2021.

[22] E. Palkopoulou, C. Merkle, D. A. Schupke, C. G. Gruber, and A. Kirstädter, "Traffic models for future backbone networks – a service-oriented approach," *European Transactions on Telecommunications*, vol. 22, no. 4, pp. 137–150, 2011.

[23] S. Krishnan and H. S. Dhillon, "Spatio-temporal interference correlation and joint coverage in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5659–5672, 2017.

[24] S. Kamath, S. Singh, and M. S. Kumar, "Multiclass queueing network modeling and traffic flow analysis for SDN-enabled mobile core networks with network slicing," *IEEE Access*, vol. 8, pp. 417–430, 2020.

[25] A. Soule, A. Lakhina, N. Taft et al., "Traffic matrices: balancing measurements, inference and modeling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 362–373, 2005.

[26] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[27] M. Malboubi, S. Peng, P. Sharma, and C. N. Chuah, "A learning-based measurement framework for traffic matrix inference in software defined networks," *Computers & Electrical Engineering*, vol. 66, pp. 369–387, 2018.

[28] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): a survey," *Computer Communications*, vol. 170, pp. 19–41, 2021.

[29] D. Jiang, Z. Xu, and H. Xu, "A novel hybrid prediction algorithm to network traffic," *Annals of Telecommunications-Annales Des Télécommunications*, vol. 70, no. 9-10, pp. 427–439, 2015.

[30] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[31] Man-Chun Tan, S. C. Wong, Jian-Min Xu, Zhan-Rong Guan, and Peng Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 60–69, 2009.

[32] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[33] M. Szmit and A. Szmit, "Usage of pseudo-estimator LAD and SARIMA models for network traffic prediction: case studies,"

in *International Conference on Computer Networks*, Springer, 2012.

[34] C. Chen, Q. Pei, and L. Ning, “Forecasting 802.11 traffic using seasonal ARIMA model,” in *2009 International Forum on Computer Science-Technology and Applications*, Chongqing, China, December 2009.

[35] Z. Qian, D. Juan, P. Bogdan, C. Y. Tsui, D. Marculescu, and R. Marculescu, “A support vector regression (SVR)-based latency model for Network-on-Chip (NoC) architectures,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 471–484, 2016.

[36] C. Xiang, P. Qu, and X. Qu, “Network traffic prediction based on MK-SVR,” *Journal of Information and Computational Science*, vol. 12, no. 8, pp. 3185–3197, 2015.

[37] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, “An improved *K*-nearest neighbor model for short-term traffic flow prediction,” *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[38] N. Ramakrishnan and T. Soni, “Network traffic prediction using recurrent neural networks,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, USA, December 2018.

[39] Y. Li, X. Su, A. Y. Ding et al., “Enhancing the Internet of Things with knowledge-driven software-defined networking technology: future perspectives,” *Sensors*, vol. 20, no. 12, p. 3459, 2020.

[40] L. Pu, L. Jiao, X. Chen, L. Wang, Q. Xie, and J. Xu, “Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1751–1767, 2018.

[41] M. Li, Y. Wang, Z. Wang, and H. Zheng, “A deep learning method based on an attention mechanism for wireless network traffic prediction,” *Ad Hoc Networks*, vol. 107, p. 102258, 2020.

[42] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, “An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: a deep learning approach,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5141–5154, 2018.

[43] J. Zhao, H. Qu, J. Zhao, and D. Jiang, “Spatiotemporal mraffic ,atrix prediction: a deep learning approach with wavelet multiscale analysis,” *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 12, 2019.

[44] G. Siracusano and A. La Corte, *Forecasting Mobile Traffic with Spatiotemporal correlation using Deep Regression*, 2019.

[45] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering, advances in neural information processing systems,” in *Neural Information Processing Systems (NIPS)*, La Jolla, 2016.

[46] F. Dai, P. Huang, X. Xu, L. Qi, and M. R. Khosravi, “Spatio-temporal deep learning framework for traffic speed forecasting in IoT,” *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 66–69, 2020.

[47] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, *Structured Sequence Modeling with Graph Convolutional Recurrent Networks*, Springer International Publishing, Cham, 2018.

[48] L. Fang, X. Cheng, H. Wang, and L. Yang, “Idle time window prediction in cellular networks with deep spatiotemporal modeling,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1441–1454, 2019.

[49] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2019.

[50] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[51] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, *Spectral Networks and Locally Connected Networks on Graphs*, Computer ence, 2013.

[52] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” 2016, https://arxiv.org/abs/1609.02907.

[53] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[54] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[55] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, “Providing public intradomain traffic matrices to the research community,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[56] W. Gaihua, Z. Tianlun, D. Yingying, L. Jinheng, and C. Lei, “A serial-parallel self-attention network joint with multi-scale dilated convolution,” *IEEE Access*, vol. 9, no. 5, pp. 71909–71919, 2021.