Cellular Network Traffic Prediction Based on Correlation ConvLSTM and Self-Attention Network

Xuesen Ma, Biao Zheng, Gonghui Jiang, and Liu Liu

***Abstract*— Predicting the future dynamicity of the network traffic are crucially important to support the 5G intelligent system and automated network management. In this letter, we propose a Correlation-based ConvLSTM and Self-Attention-based Network (CCSANet) to accurately predict complex cellular network traffic. In the proposed CCSANet, the correlation layer is leveraged in ConvLSTM to improve the ability of extracting consecutive spatial features. Additionally, the self-attention is adopted to aggregate the ability of extracting the dependency between external factors feature and network traffic feature. Experimental evaluations on real-world cellular network traffic datasets demonstrate the effectiveness of CCSANet, which outperforms the state-of-the-art (SOTA) methods.**

***Index Terms*— Cellular traffic prediction, correlation layer, ConvLSTM, self-attention mechanism.**

I. Introduction

NETWORK traffic prediction (NTP) [1] refers to the estimation of traffic data volume in the future. With the predicted traffic data, proactive measures can be taken to mitigate the network congestion and outage caused by burst transmissions in the communication network. Therefore, NTP provides the decision basis of communication network management and optimization [2]. Predicting the future dynamicity of the network traffic is crucially important to support the 5G intelligent system and automated network management [3].

In recent works, NTP has been modeled as a time series analysis problem, which is generally categorized into classic prediction methods and Neural Network (NN) prediction methods. Classic prediction methods are mainly based on statistics or probability distributions, such as α-stable distribution [4], Autoregressive Integrated Moving Average (ARIMA) [5], and covariance function [6]. However, most of these methods generally rely on the mean value of historical traffic and often fail to predict complex network traffic accurately.

Compared with the classic prediction methods, the NN-based methods can better extract network traffic with complex characteristics, such as Long Short-Term Memory neural network (LSTM) [7], etc. In the NTP problem, these methods have comprehensively studied the characteristics of cellular networks, which have shown that changes in communication traffic have both temporal and spatial autocorrelation. However, while LSTM focuses mainly on temporal features, the ability to extract spatial correlations needs improvement.

To further model the spatial dependencies of network traffic, Convolutional Neural Network (CNN) is used in NTP. Zhang et al. [8] proposed a cellular traffic prediction method based on a convolutional neural network (STDenseNet). In [9], the authors proposed a new hybrid spatiotemporal network (HSTNet) and considered the time characteristics to enhance the NTP accuracy. While in [10], the author considered more external factors such as base stations (BSs), points of informations (POIs), and Socials. Besides, they proposed the STCNet based on convolutional LSTM (ConvLSTM) [11] to model temporal and spatial dependencies. Shen et al. [12] proposed a time-wise attention aided convolutional neural network (TWACNet) structure for citywide cellular traffic prediction. Experimental results demonstrated the effectiveness of the self-attention mechanism. These methods have comprehensively studied the spatial characteristics of cellular networks. Meanwhile, these methods have indicated that modeling correlation between the current and previous features is important for NTP. However, these methods primarily extract spatial correlations using CNN or ConvLSTM and may not fully capture the spatial dependence between adjacent features. This limitation could lead to inadequate characterization of consecutive features. Moreover, the existing work indicates that ConvLSTM often shows more effectiveness than the dense convolution-based methods [10].

This letter proposes a cellular NTP model called CCSANet, based on Correlation ConvLSTM and Self-Attention. The model addresses the issues mentioned earlier by leveraging the correlation layer to calculate the correlation between two consecutive cellular traffic features. This technique is widely used in computer vision to model consecutive spatial features [13]. In addition, the self-attention (SA) mechanism is used to aggregate the ability of extracting the dependency of external factors and network traffic features. Experimental results show the effectiveness of CCSANet over existing NTP methods.

II. Data Observation and Analysis

A. Citywide Cellular Traffic Dataset

The dataset used in this letter is the Telecom Italia Big Data Challenge, which is widely used in the field of NTP [14] and publicly accessible. The dataset contains three real network traffic data (SMS, Call, and Internet) recorded in Milan, Italy for a period of two months. The city of Milan is divided

Manuscript received 28 March 2023; revised 4 May 2023; accepted 5 May 2023. Date of publication 11 May 2023; date of current version 12 July 2023. This work was supported by the Hefei Municipal Natural Science Foundation (No. 2022015) and National Key R&D Program of China (No. 2020YFC1512601). The associate editor coordinating the review of this letter and approving it for publication was F. Kara. *(Corresponding author: Xuesen Ma.)*

Xuesen Ma and Biao Zheng are with the Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei 230009, China, and also with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China (e-mail: mxs@hfut.edu.cn).

Gonghui Jiang and Liu Liu are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China.

Digital Object Identifier 10.1109/LCOMM.2023.3275327

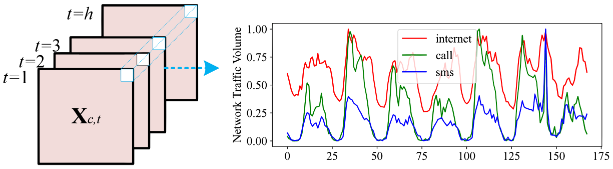
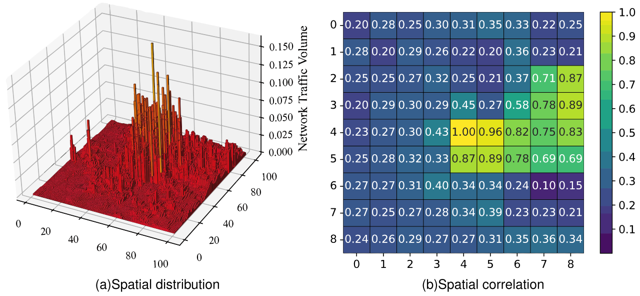


Fig. 1. The temporal correlation for network traffic.



(a)Spatial distribution (b)Spatial correlation

Fig. 2. The spatial distribution and correlation for network traffic.

into $H \times W$ sub-area of units, where H and W represent the number of rows and columns of cells. Specifically, the dataset covers an area of 0.0552 km^2 divided into a grid of 10,000 cells, and $H = W = 100$. The value of each cell represents the statistical value of traffic in the area.

B. Data Analysis

1) Temporal Domain: Fig. 1 shows the SMS, Call, and Internet traffic volume for a given cell as a function of time. It can be seen that the traffic dynamic patterns of the three are similar. The traffic follows obvious periodicity, especially the traffic value on weekends is lower than that on weekdays.

2) Spatial Domain: Fig. 2 illustrates the spatial distribution of a snapshot of internet traffic. As expected, the traffic is unevenly distributed throughout the city, with denser traffic in urban centers than in suburbs. Despite these differences, there is still a spatial correlation between traffic in different areas that a prediction model should be able to capture. To measure this correlation, we use the widely used Pearson correlation coefficient ρ [15] to assess the relationship between the target cell $x^{(i,j)}$ and its surrounding cell $x^{(i',j')}$.

$$\rho = \frac{\operatorname{cov}\left(x^{(i,j)}, x^{(i',j')}\right)}{\sigma_{x^{(i,j)}}\,\sigma_{x^{(i',j')}}}, \tag{1}$$

where cov denotes the covariance operator and σ represents the standard deviation. In Fig. 2(b), we choose the cell with coordinates (4,4) as the target cell and compute its Pearson correlations with other cells according to Eq. (1). It can be observed that there is indeed a correlation among cells, which has a great relationship with the distance. At the same time, although the cells (5,3) and (5,5) have the same distance from the target cell, they may have different correlation values 0.27 and 0.89 due to external factors. It shows that the correlation may relate to other factors besides spatial distance. Therefore, we need to investigate novel methods to capture the spatiotemporal latent correlations of cellular network traffic.

III. The Proposed Prediction Model

This section introduces the proposed CCSANet, which mainly consists of four modules: *Corr-ConvLSTM*, *Self-Attention Module*, *Time Embedding*, and *Fusion Output Module*, as shown in Fig. 3. The model takes $\mathbf{X}_{c,h}$, $\mathbf{X}_e$, and $\mathbf{X}_d$ as inputs. Specifically, $\mathbf{X}_{c,t}$ denotes network traffic flow at time t, where $c \in \{sms, call, internet\}$. To represent the historical network traffic flow, $\mathbf{X}_{c,h}$ is defined as $\mathbf{X}_{c,h} := [\mathbf{X}_{c,t-h}, \mathbf{X}_{c,t-(h-1)}, \ \ldots, \mathbf{X}_{c,t}] \in \mathbb{R}^{h \times H \times W}$. Then the NTP output $\hat{\mathbf{X}}_{c,p} \in \mathbb{R}^{p \times H \times W}$ can be defined as:

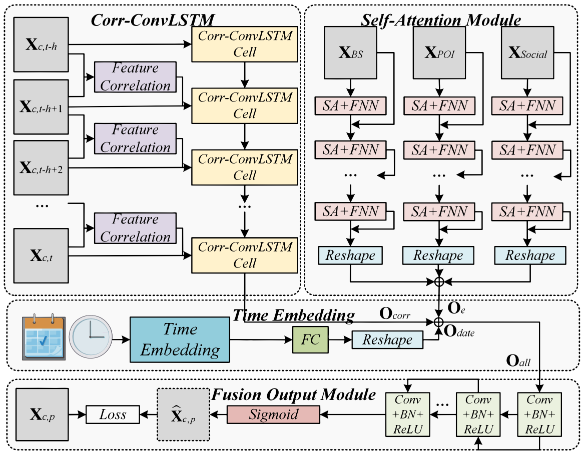


Fig. 3. Overview of the proposed CCSANet.

$$\hat{\mathbf{X}}_{c,p} = CCSANet\left(\mathbf{X}_{c,h}, \mathbf{X}_e, \mathbf{X}_d\right). \tag{2}$$

A. Corr-ConvLSTM

Firstly, we extract the multi-channel features $\mathbf{f}_{t,k}$ and $\mathbf{f}_{t-1,k}$ of the input $\mathbf{X}_{c,t}$ and $\mathbf{X}_{c,t-1}$ through the convolution operation, where $*$ denotes the convolution operator, *k* is the number of channels and $k \in \{1, 2, 3\}$.

Secondly, the correlation $\mathbf{C}_{t,k}$ of each channel of two consecutive cellular traffic features are calculated by correlation function (*CF*) along the channel dimension, and the *CF* is dot-product operator in this letter. The correlation of the *k* channels is concatenated to obtain the correlation map $\mathbf{C}_t$.

$$\mathbf{f}_{t,k} = \mathbf{W}_{t,k} * \mathbf{X}_{c,t} + \mathbf{b}_{t,k}, \tag{3}$$

$$\mathbf{C}_{t,k} = CF\left(\mathbf{f}_{t-1,k}, \mathbf{f}_{t,k}\right), \tag{4}$$

$$\mathbf{C}_t = concat\left(\mathbf{C}_{t,1}, \mathbf{C}_{t,2}, \mathbf{C}_{t,3}\right). \tag{5}$$

Thirdly, the correlation map $\mathbf{C}_t$ is then passed into a convolutional layer and a global average pooling layer (GAPL) [16]. Finally, we obtain the factor $\mathbf{u}_t$, which measures the dynamic change between two consecutive cellular traffic features.

$$\mathbf{C}_t^o = \mathbf{W}_c * \mathbf{C}_t + \mathbf{b}_c, \tag{6}$$

$$\mathbf{u}_t = \sigma(GAPL(\mathbf{C}_t^o)), \tag{7}$$

where $\mathbf{W}_{t,k}$,$\mathbf{W}_c$ and $\mathbf{b}_{t,k}$,$\mathbf{b}_c$ denote the weights and bias for the convolutional layer respectively, $\mathbf{C}_t^o \in \mathbb{R}^{H \times W}$.

We use $\mathbf{u}_t$ to decide the correlation information between two consecutive cellular traffic features and integrate it into ConvLSTM. Then Corr-ConvLSTM calculates input $\mathbf{X}_{c,t}$ by the following formula:

$$i_t = \sigma(\mathbf{W}_{xi} * (1 - \mathbf{u}_t)\mathbf{X}_{c,t} + \mathbf{W}_{hi} * \mathbf{u}_t H_{t-1} + \mathbf{b}_i),$$

$$f_t = \sigma(\mathbf{W}_{xf} * (1 - \mathbf{u}_t)\mathbf{X}_{c,t} + \mathbf{W}_{hf} * \mathbf{u}_t H_{t-1} + \mathbf{b}_f),$$

$$\begin{aligned} g_t &= \tanh(\mathbf{W}_{xc} * (1-\mathbf{u}_t)\mathbf{X}_{c,t} + \mathbf{W}_{hc} * \mathbf{u}_t H_{t-1} + \mathbf{b}_c), \\ C_t &= \mathbf{u}_t f_t \circ C_{t-1} + (1-\mathbf{u}_t) i_t \circ g_t, \\ o_t &= \sigma(\mathbf{W}_{xo} * (1-\mathbf{u}_t)\mathbf{X}_{c,t} + \mathbf{W}_{ho} * \mathbf{u}_t H_{t-1} + \mathbf{b}_o), \\ H_t &= o_t \circ \tanh(C_t), \end{aligned} \tag{8}$$

where $\sigma(\cdot)$ denotes the activation function, and $\circ$ denotes the Hadamard product. $\mathbf{W}_{*i}, \mathbf{W}_{*f}, \mathbf{W}_{*c}, \mathbf{W}_{*o}$ and $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o$ denote the learnable weights and bias respectively. i_t, f_t, g_t, C_t, o_t are referred to as *input, forget, input modulation, cell and output gates*. Finally, we can get the output $\mathbf{O}_{corr} \in \mathbb{R}^{p\times H\times W}$ of Corr-ConvLSTM module.

B. Self-Attention Module

We adopt the self-attention module to extract feature representations of external factors ($e \in \{BSs, POIs, Socials\}$). The external input $\mathbf{X}_e$ is mapped into different feature spaces as the *query*: $\mathbf{Q}_e = \mathbf{W}_q\mathbf{X}_e \in \mathbb{R}^{d_k\times N}$, the *key*: $\mathbf{K}_e = \mathbf{W}_k\mathbf{X}_e \in \mathbb{R}^{d_k\times N}$ and the *value*: $\mathbf{V}_e = \mathbf{W}_v\mathbf{X}_e \in \mathbb{R}^{d_v\times N}$, where $\{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\}$ is a set of weights for 1×1 convolutions, d_k and d_v are the number of channels, and where $N = H \times W$.

The similarity scores of each pair of points are calculated by applying the matrix multiplication as:

$$\mathbf{e} = \mathbf{Q}_e^T\mathbf{K}_e \Big/ \sqrt{d_k} \in \mathbb{R}^{N\times N}. \tag{9}$$

The similarity between the *i*-th point and the *j*-th point can be indexed as $e_{i,j} = \left(\mathbf{X}_{e,i}^T\mathbf{W}_q^T\right)\left(\mathbf{W}_k\mathbf{X}_{e,j}\right)/\sqrt{d_k}$, where the $\mathbf{X}_{e,i}$ and the $\mathbf{X}_{e,j}$ are feature vectors with the shape $d_k \times 1$. Then the similarity scores are normalized along with column:

$$\alpha_{i,j} = \exp\left(e_{i,j}\right) \Bigg/ \sum_{k=1}^{N} \exp\left(e_{i,k}\right), i,j \in \{1,2,\ldots,N\}. \tag{10}$$

The aggregated feature of the *i*-th location is calculated with a weighted sum across all locations:

$$\mathbf{Z}_i = \sum_{j=1}^{N} \exp\left(\alpha_{i,j}\left(\mathbf{W}_v\mathbf{X}_{e,j}\right)\right), \tag{11}$$

where $\mathbf{W}_v\mathbf{X}_{e,j} \in \mathbb{R}^{d_v\times 1}$ is the *j*-th column of the *value* $\mathbf{V}_e$. Then, $\mathbf{Z} \in \mathbb{R}^{d_v\times N}$ is input in the fully connected layer and the output is vector $\mathbf{o}_e \in \mathbb{R}^{pN\times 1}$. Finally, The vector is reshaped into a output $\mathbf{O}_e \in \mathbb{R}^{p\times H\times W}$.

$$\mathbf{o}_e = \sigma\left(\mathbf{W}_e\mathbf{Z} + \mathbf{b}_e\right), \tag{12}$$

$$\mathbf{O}_e = Reshape\left(\mathbf{o}_e\right), \tag{13}$$

where $\mathbf{W}_e$ and $\mathbf{b}_e$ are learnable parameters of the fully connected layer.

C. Time Embedding

We can see in the previous data analysis that data characteristics are strongly correlated with cellular network traffic. We extract four kinds of date data, $is\_weekday\,(1/0)$, $is\_weekend(1/0)$, $day\_of\_week(0-6)$, and $hour\_of\_day(0-23)$, and treat them as features. The 33-dimensional feature vector $\mathbf{X}_d$ combines the four kinds of data and is input to the two-layer fully connected layer, and the output is vector $\mathbf{v}_d \in \mathbb{R}^{pHW\times 1}$. The vector is reshaped into a output $\mathbf{O}_d \in \mathbb{R}^{p\times H\times W}$ through a reshape layer and merged with the input result of the *Fusion Output Module*. The calculation process is as follows:

$$\mathbf{v}_d = \sigma\left(\mathbf{W}_d^{(2)}\left(\sigma(\mathbf{W}_d^{(1)}\mathbf{X}_d + \mathbf{b}_d^{(1)})\right) + \mathbf{b}_d^{(2)}\right), \tag{14}$$

$$\mathbf{O}_d = Reshape\left(\mathbf{v}_d\right), \tag{15}$$

where $\mathbf{W}_d$ and $\mathbf{b}_d$ are learnable parameters of the fully connected layer.

D. Fusion Output Module

It can be seen from the above analysis that the traffic of different cells is not only related to the correlation of continuous traffic data but also related to the period. To capture this relationship, we first fuse the correlation and periodic features to obtain the fused feature $\mathbf{O}_{all}$. Then we extract the fused feature through the multiple *DenseBlock(Conv+BN+ReLU+DeformConv)*. Finally, the predicted $\hat{\mathbf{X}}_{c,p}$ is obtained through the sigmoid activation function.

$$\mathbf{O}_{all} = concat\left(\mathbf{O}_{corr}, \mathbf{O}_e, \mathbf{O}_d\right), \tag{16}$$

$$\hat{\mathbf{X}}_{c,p} = \sigma\left(DenseBlock\left(\mathbf{O}_{all}\right)\right). \tag{17}$$

IV. Experiments

A. Experimental Process and Parameter Settings

1) Dataset and Parameter Settings: The experimental dataset comes from Telecom Italia, and its preprocessing method in NTP area [10]. We split the three datasets (SMS, Call, and Internet) into training and test sets with 1320 (55 days × 24 hours) and 168 (7 days × 24 hours), respectively. To avoid overfitting in training, we randomly divide part data from the training set as the validation set. The CCSANet adopts the widely used Adam optimizer [17] and is trained for 300 epochs with a batch size of 32. The initial learning rate is set to 0.01, and when the epoch number increases to 50% and 75%, the learning rate is reduced by a factor of 10 and 100, respectively. In the convolution module, the last layer has 1 filter with kernel size of 1×1 and sigmoid activation function. Besides, the rest layer have 16 kernel sizes of 3×3 filter and *ReLU* activation function.

2) Baseline Methods and Evaluation Metrics: We compare CCSANet with the following widely used NTP methods to evaluate the performance. The baselines including ARIMA [5], LSTM [7], STDenseNet [8], STCNet [10], HSTNet [9], and TWACNet [12]. STCNet and TWACNet are the SOTA ConvLSTM-based and attention-based methods in cellular NTP respectively.

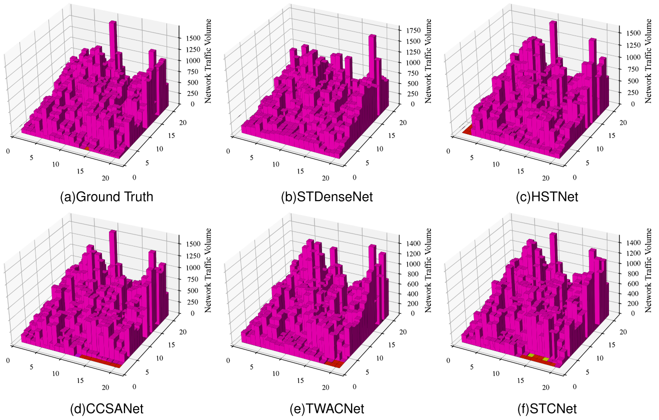
We adopt two evaluation metrics of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the model, which are widely used to measure the difference between the value of ground truth and prediction value [18].

B. Experiment Analysis

We experimentally compare the proposed CCSANet with baseline models on three different cellular network traffic datasets, and the evaluation results are plotted in Table I.

TABLE I
EXPERIMENTAL RESULTS OF RMSE AND MAE

Methods	RMSE			MAE		
	Call	SMS	Internet	Call	SMS	Internet
ARIMA	62.74	91.17	295.45	36.51	57.49	213.64
LSTM	49.42	72.11	229.38	25.08	38.28	152.84
DenseNet	45.09	68.20	212.88	23.36	38.68	131.88
TWACNet	42.41	60.23	181.80	22.52	35.38	109.36
HSTNet	44.09	62.11	184.02	22.61	37.28	108.53
STCNet	33.47	55.78	170.02	16.49	31.00	98.15
CCSANet	**31.85**	**53.49**	**164.06**	**15.88**	**30.07**	**90.76**



(a)Ground Truth (b)STDenseNet (c)HSTNet

(d)CCSANet (e)TWACNet (f)STCNet

Fig. 4. The snapshots of NTP results.

As illustrated in this Table, ARIMA has the highest MAE and RMSE on the three traffic datasets, because it only considers the historical temporal characteristics of the data without accounting for other dependencies. The performance of LSTM is better than statistical methods but worse than other deep learning methods. STDenseNet ignores the influence of external factors, while HSTNet only takes temporal attributes into account and neglects other external factors like BSs information and POIs distribution. TWACNet adopts a convolution-based network, but its performance is lower than that of the ConvLSTM-based methods. STCNet uses the ConvLSTM-based network but does not incorporate the Self-Attention and correlation layer to enhance the feature extraction. All of the above works primarily rely on networks (e.g. CNN, ConvLSTM, and SA) to extract hidden information, which may not model the consecutive spatiotemporal features. The CCSANet achieves the best performance compared to the baseline methods. On the one hand, CCSANet employs the correlation layer in ConvLSTM to improve its ability to extract consecutive spatiotemporal features. On the other hand, CCSANet incorporates the self-attention mechanism to aggregate the ability to extract the dependency between external factors and network traffic feature.

To give an additional perspective on the effectiveness of NTP, we plot a randomly selected snapshots of the Internet for NTP results in Fig. 4. Compared to the baseline methods, CCSANet shows better predictive performance, as evidenced by the very similar prediction results of every cell compared to the ground truth in Fig. 4(d).

V. CONCLUSION

In this letter, we propose a novel cellular NTP method called CCSANet, which enhances the ability of extracting consecutive spatial features by adopting the correlation layer in ConvLSTM. Additionally, the method aggregates the ability of extracting the dependency of external factors and network traffic feature by adopting the self-attention mechanism. Experimental results show that CCSANet outperforms the SOTA method in terms of RMSE and MAE on real-world cellular network traffic datasets. This demonstrates our proposed method can be used to improve the accuracy of cellular network traffic prediction.

REFERENCES

[1] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): A survey," *Comput. Commun.*, vol. 170, pp. 19–41, Mar. 2021.

[2] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10.

[3] F. Xu et al., "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 796–805, Sep./Oct. 2016.

[4] 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 Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3899–3912, Jun. 2017.

[5] Y. Shu, M. Yu, O. W. W. Yang, J. Liu, and H. Feng, "Wireless traffic modeling and prediction using seasonal ARIMA models," *IEICE Trans. Commun.*, vol. E88-B, no. 10, pp. 3992–3999, Oct. 2005.

[6] X. Chen, Y. Jin, S. Qiang, W. Hu, and K. Jiang, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 3585–3591.

[7] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 231–240.

[8] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, Aug. 2018.

[9] D. Zhang, L. Liu, C. Xie, B. Yang, and Q. Liu, "Citywide cellular traffic prediction based on a hybrid spatiotemporal network," *Algorithms*, vol. 13, no. 1, p. 20, Jan. 2020.

[10] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[11] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, Jun. 2015, pp. 802–810.

[12] W. Shen, H. Zhang, S. Guo, and C. Zhang, "Time-wise attention aided convolutional neural network for data-driven cellular traffic prediction," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1747–1751, Aug. 2021.

[13] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[14] G. Barlacchi et al., "A multi-source dataset of urban life in the city of Milan and the Province of Trentino," *Sci. Data*, vol. 2, no. 1, Oct. 2015, Art. no. 150055.

[15] J. Wang et al., "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[16] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[18] N. Zhao, Z. Ye, Y. Pei, Y.-C. Liang, and D. Niyato, "Spatial-temporal attention-convolution network for citywide cellular traffic prediction," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2532–2536, Nov. 2020.