Cellular Traffic Load Prediction with LSTM and Gaussian Process Regression

Wei Wang*†, Conghao Zhou†, Hongli He‡, Wen Wu†, Weihua Zhuang†, Xuemin (Sherman) Shen†
*College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, China
†Department of Electrical and Computer Engineering, University of Waterloo, Canada
‡School of Information Engineering, Zhejiang University, China
Email: wei_wang@nuaa.edu.cn; hongli_he@zju.edu.cn; {c89zhou, wen.wu, wzhuang, sshen}@uwaterloo.ca.

***Abstract*—Accurate cellular traffic load prediction is a prerequisite for efficient and automatic network planning and management. Considering diverse users' activities at different locations and times, it is technically challenging to characterize the network resource demands at different time scales via traditional prediction methods. In this paper, we propose to combine the long short-term memory (LSTM) and Gaussian process regression (GPR) to achieve accurate single-cell level cellular traffic prediction, using the open Milan cellular traffic dataset provided by Telecom Italia. Firstly, the dominant periodic components of the cellular data are extracted, and then the small components are fed to the LSTM network. To further improve the prediction accuracy, GPR is used to recover the residual components. Extensive experiments are conducted based on the dataset, and it is shown that the proposed LSTM-GPR scheme outperforms the benchmark schemes, especially for a relatively long time and burst traffic prediction.**

I. Introduction

Accurate and timely network traffic load prediction plays an important role in efficient and automatic planning and deployment of ultra-dense base stations (BSs), multi-dimensional resource management, and on/off BS switching in future wireless networks [1]–[3]. It has a great potential in reducing the network congestion, improving users' quality of experience (QoE), as well as reducing network operation cost [4]. For example, the predicted traffic load information of a cellular network in a certain area allows the operator to flexibly dispatch some unmanned aerial vehicles (UAVs) as moving access points (APs) to tackle the congested traffic during rush hours or in some special events, such as sport events, assembly, etc. The operator may also choose to close some idle BSs in the night to reduce the energy expenditure. Moreover, the traffic load directly affects the optimal caching policy and optimal allocation of communication and computing resources [5]. With the information of predicted traffic demand at different BSs, an optimal allocation policy for caching, communication, and computing resources at macro-cell and small-cell BSs can be executed to reduce the transmission delay, as well as to improve the network throughput.

Based on different use cases, network data traffic prediction can be conducted at different time scales: The long-term prediction aims to predict the evolution trend of whole network in the next few days or weeks for network infrastructure planning and deployment; the short-term prediction usually targets at the network load variations in a second or smaller scale for real-time network resource management; and the mid-term prediction focuses on a relatively large time scale at minutes or hours level with the prediction results used for dynamic resource adjustment among different network slices [6]. There exist plenty of studies on traffic prediction in the literature [7], [8]. As a linear regressive model, the autoregressive integrated moving average (ARIMA) model and its variants have been proposed for forecasting short-term temporal evolution of network traffic [9]. However, the ARIMA model tends to converge to the mean value of the past data, and cannot capture large nonlinear variations underlying the traffic flow. Support vector regression (SVR) has also been applied to traffic prediction [8], but it is difficult to determine the key parameters in the model.

Considering the large variations of cellular traffic flow and high accuracy requirements, it is extremely challenging to accurately predict the data traffic at different time scales and granularity using the conventional approaches. Firstly, the traffic demand is highly affected by many factors such as day of the week (weekday or weekend), time of the day (rush hour or off-peak hour), public holidays. According to the dataset in Milan, Italy, recorded in Nov. 1, 2013, the Internet traffic volume during peak hours can reach more than 10 times of that in a normal time. Secondly, the traffic volume is highly location dependent and has a strong spatial correlation among nearby cells, but such spatial correlation is nontrivial to be incorporated in existing approaches [10]. Moreover, the traffic data is exploding with the growth of smart applications and smart devices, and the traditional methods become more and more inefficient in the era of big data networks. However, the fast development of machine learning (ML) and big data techniques, provides us an opportunity to rethink how to exploit the big cellular traffic data for advanced traffic prediction approaches under the ML framework. In recent years, artificial neural networks (ANNs) have been adopted for traffic prediction [11], to achieve high accuracy. As a deep learning model, long short-term memory (LSTM) leverages the strengths of the convolutional layer to discover the local dependency patterns among multi-dimensional input variables and the recurrent layer to captures complex long-term dependencies, and has been used for traffic prediction in [10]. It is worth noting that, even though LSTM is capable of modeling the long-term dependencies by using the memory

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

block, it is ineffective to deal with the components whose period is longer than the maximum period of the LSTM network. Therefore, the LSTM prediction accuracy can be limited in practice.

To improve the prediction performance, various time-domain series and frequency-domain analysis methods have been proposed to decompose the complex time series [12], which naturally inspires us to adopt similar methodology to deal with the complex real cellular traffic flow. Specifically, we first extract the dominant periodic components by examining the traffic patterns of the data flow, and then use LSTM to learn the small values after subtracting the dominant periodic components. To further improve the prediction performance, we propose a compensation step with Gaussian process regression (GPR) after the LSTM network by estimating the residual random components. As a machine learning regression method, GPR is capable of dealing with a nonlinear and high dimensional system, and can capture the residual prediction errors caused by burst traffic. Therefore, the network load prediction performance can be improved significantly.

The remainder of the paper is organized as follows. Section II provides a description of the dataset and research problems under consideration. Section III presents the details of our proposed scheme, followed by experimental results and conclusions in Section IV and Section V, respectively.

II. Dataset Description and Problem Statement

A. Dataset Description

We use an open cellular traffic dataset to conduct the traffic prediction and analysis. Specifically, this dataset was collected by Telecom Italia from Nov. 1, 2013 to Jan. 1, 2014 with a time interval of 10 minutes in the city of Milan[1]. The whole area is divided into a grid of 100×100 squares. In each cell, three kinds of cellular traffic are recorded by the service provider, i.e., the short message service (SMS), call service, and Internet service. Without loss of generality, we focus on the Internet traffic prediction, and the approaches presented can be applied to the prediction of SMS and call services. Moreover, to facilitate the learning process, we only extract the data of the central 20×20 cells in the first month for our processing.

B. Problem Statement

Traffic load prediction can be represented as a time series model. Partition time into intervals of constant duration. Let $X_i(t)$ denote the observed traffic flow quantity during the tth time interval at the ith cell. Given sequence $X_i(t)$ of observed traffic flow data, $i = 1, 2, \ldots, N$, $t = 1, 2, \ldots, T$, where N is the number of cells, T is the length of the past observation, the problem of cellular traffic load prediction is to predict the traffic flow, $\hat{X}_i(t+l)$, at time interval $t+l$ for some prediction horizon l, where l is a positive integer.

[1]The whole dataset is collected over 62 days, with approximately 300 million records, totally about 19 GB. The dataset is available at https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV

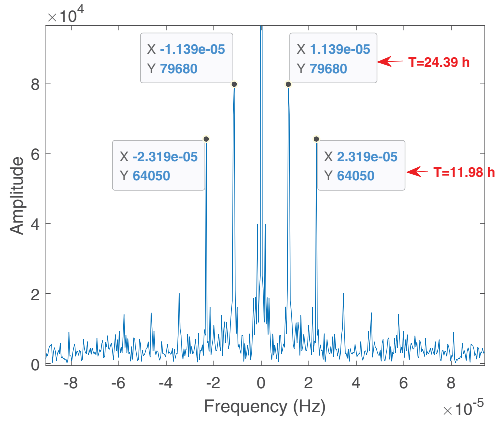


Fig. 1. The Fourier analysis of the cellular traffic data of cell 250.

Stochastic approaches have been used for time series prediction, but the performance deteriorates severely with an increase of the prediction horizon. Moreover, the burst traffic component cannot be captured with the stochastic approaches. In the following, we propose to predict the traffic load of a certain cell with advanced machine learning techniques, with an aim to improve the long time and burst data prediction performance. For the ease of notification, we omit subscript i, and use X_t to denote the traffic volume at time interval t.

III. Proposed Prediction Techniques

In this section, we present the proposed prediction scheme. Specifically, to facilitate the learning process, we first extract the large periodic components with Fourier analysis, and then the remaining small random components are fed to the LSTM network. To further improve the prediction performance, GPR is adopted to recover the residual components.

A. Analysis of periodic components of traffic data

As mentioned in Section II, due to the users' periodic behavior (e.g., the user's daily habit to use an APP or watch the videos), the cellular traffic has strong periodic patterns. The Fourier transform of the cellular traffic data of cell 250 is shown in Fig. 1. We can observe that there are several large periodic components, around frequencies $f = 0, 2.3148 \times 10^{-5}, 1.1574 \times 10^{-5}$ Hz corresponding to the constant component and periodic components with a period of 12 hour, and 24 hour, respectively. The dominant periodic components greatly affect the learning of the LSTM network. In this subsection, we aim to analyze the periodic characteristics of the cellular traffic and then extract such components to facilitate the learning process of LSTM.

The periodic components of the cellular traffic series can be modeled as a summation of a finite periodic signals with

$$X_t^{\mathrm{p}} = \sum_{m=1}^{M} \alpha_m \sin(2\pi f_m t + \theta_m) \tag{1}$$

where M is the number of periodic signals, f_m and θ_m are the frequency and phase of the mth component, respectively, with

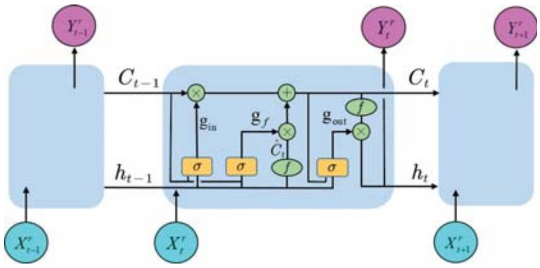


Fig. 2. An illustration of the LSTM network.

α_m being the corresponding magnitude. Given a K value, the parameters α_m, f_m, and θ_m can be obtained by least square (LS) estimation.

B. LSTM Prediction

As a special kind of RNN, LSTM can learn the long-term dependency and thus can model the sequence information of the cellular traffic [13]. The core idea behind LSTM is to design the structure of cells to regulate information flow in different cells. The LSTM is capable of removing or adding information to the cell state through three self-parameterized controlling gates, i.e., the input gate, output gate, and forget gate.

As shown in Fig. 2, the input sequence and output sequence are denoted by X_t^{r} and Y_t^{r}, respectively, and the long-term memory and short-term memory are denoted by C_t and h_t, respectively. The new long-term memory C_t includes two information components: old long-term memory C_{t-1} and new information candidate $\hat{C}_t$ needs to be stored, which is given by

$$C_t = g_{\mathrm{f}} \odot C_{t-1} + g_{\mathrm{in}} \odot \hat{C}_t \tag{2}$$

where g_{f} and g_{in} are the forget gate and input gate, respectively. The input gate is used to decide what new information needs to be stored. The forget gate is used to scale the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. Note that $\odot$ is the element-wise product or Hadamard product. New information candidate $\hat{C}_t$ can be calculated as

$$\hat{C}_t = f(W_c \cdot \left[h_{t-1}, X_t^{\mathrm{rand}}\right] + b_c) \tag{3}$$

where $f(\cdot)$ is the activation function which limits the output value to be between -1 and 1, such as tanh function. The short-term memory h_t is calculated as $h_t = g_{\mathrm{out}} \odot f(C_t)$, where g_{out} is an output gate value, which controls the output flow of cell activations into the rest of the network. In addition, the modern LSTM architecture contains peephole connections [14] to address the weakness that the cell output is close to zero, as long as the output gate is closed. Therefore, every gate is connected directly with the input information flow. Generally, all gates are controlled by old input X_{t-1}^{r} and old output Y_{t-1}^{r}. The forget gate and input gate are controlled by old long-term memory C_{t-1}, and the output gate is decided by new long-term memoryC_t. All gates are given by

$$g_{\mathrm{in}} = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, X_t^{\mathrm{r}}] + b_i) \tag{4}$$

$$g_{\mathrm{f}} = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, X_t^{\mathrm{r}}] + b_f) \tag{5}$$

$$g_{\mathrm{out}} = \sigma(W_o \cdot [C_t, h_{t-1}, X_t^{\mathrm{r}}] + b_o) \tag{6}$$

where W_i, W_f,W_o are weights of neural networks, b_i, b_f,b_o are biases of neural networks, and $\sigma(\cdot)$ represents the sigmoid function which maps the input to the range between 0 and 1. Based on (2) - (6), the output of neural network is $Y_t^{\mathrm{r}} = h_t$.

Generally, data size In_Size of the input data sequence depends on different practical applications (e.g., the length of input traffic data). The repeating module number is equal to the input size, and all of these modules form an LSTM layer in the neural network. The number of LSTM layers is denoted by depth $Dept\_Size$ of the neural network. The updating of the neural network parameters (weights and biases) is based on the Back propagation algorithm.

C. Gaussian Process Regression

By subtracting the LSTM predicted components from X_t^{r}, the residual components are mainly composed of the random/burst components, which follows a Gaussian distribution in a high probability based on the central limit theorem. To further improve the prediction performance of burst traffic, we use the Gaussian process regression approach to predict the residual components. The Gaussian process (GP) approach is a Bayesian nonparametric model that generalizes the Gaussian distributions to functions, and any finite number of random variables in a GP have joint Gaussian distributions [15]. The GP is capable of dealing with high-dimensional nonlinear problems, and is fully specified by its mean function and covariance function. If function y is distributed as a GP with mean function m and covariance function k, then we have $y \sim \mathcal{GP}(m, k)$.

Let $\mathbf{y} = \{y_1, y_2, \ldots, y_t\}$ be the known function values of the training set $X = \{x_1, x_2, \ldots, x_t\}$, then $\mathbf{y}$ follows a joint Gaussian distribution, i.e., $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = m(x_t), t = 1, \ldots, T$, $\boldsymbol{\Sigma}$ is the training set covariance. Let $\mathbf{y}_*$ be a set of function values corresponding to the test set input, X_*, then

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_* \\ \boldsymbol{\Sigma}_*^T & \boldsymbol{\Sigma}_{**} \end{bmatrix}\right) \tag{7}$$

where $\boldsymbol{\mu}_*$ is the mean of test data set X_*, $\boldsymbol{\Sigma}_* = [\mathbf{k}(X_*, x_1), \mathbf{k}(X_*, x_2), \ldots, \mathbf{k}(X_*, x_t)]$, $\boldsymbol{\Sigma}_{**}$ is the test set covariance. Since $\mathbf{y}$ is known, we can derive the conditional distribution of $\mathbf{y}_*$ given $\mathbf{y}$, which is given by [16]

$$\mathbb{P}(\mathbf{y}_* \mid \mathcal{D}_{1:t}, X_*) = \mathcal{N}\left(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right) \tag{8}$$

with

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \boldsymbol{\mu}_* + \boldsymbol{\Sigma}_*^T \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \\ \hat{\boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}_{**} - \boldsymbol{\Sigma}_*^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}_*. \end{aligned} \tag{9}$$

This is the posterior distribution for a specific set of test cases. Notice that to make the GP useful in practice, the mean and covariance need to be chosen properly, which can be achieved by hyper-parameters training [17]. Moreover, the covariance matrix used here is a squared exponential function (Gaussian kernel), but in fact any positive definite function can be used as covariance function.

By combining the predicted results with LSTM and GPR and the dominant periodic components, we can obtain the predicted traffic volume. The detailed procedure is summarized in Algorithm 1.

Algorithm 1 Proposed scheme for cellular traffic prediction

1: Initialization: divide the whole dataset into training and test data;
2: Do Fourier transform using the training data X_t, and extract the K largest frequency;
3: Estimate the parameters of the sinusoid signal corresponding to the extracted frequencies via LS estimation or curve fitting tools;
4: Subtract the periodic components from the training data, and obtain $X_t^r = X_t - X_t^p$;
5: Train the LSTM network by using X_t^r, Y_t^r;
6: Calculate the residual component, i.e., $X_t^o = X_t^r - Y_t^r$;
7: Train the GPR with the residual components X_t^o;
8: Predict the large periodic component with the extracted frequency, and obtain $\hat{X}_t^p$;
9: Predict the small component with the trained LSTM network $\hat{X}_t^r$;
10: Conduct GPR for residual components $\hat{X}_t^o$;
11: Combine the results in Step 8-10, and obtain the predicted traffic, i.e., $\hat{X}_t = \hat{X}_t^p + \hat{X}_t^r + \hat{X}_t^o$;
12: Calculate the RMSE and MAE.

IV. Experiment Results

In this section, we present extensive experiments conducted using the open dataset obtianed in the city of Milan, to demonstrate the effectiveness of the proposed prediction scheme. The data preprocessing, experiment settings, and performance metrics are described first, and then the prediction performance is evaluated and compared with different schemes.

A. Data Preprocessing

The whole city of Milan is divided into 100×100 grids during data recording, due to the limited storage of the computer, we only extract the data in the central 400 grids in the first month as an example for traffic prediction. The 400 grids (cell) are numbered sequentially from 1 to 400 from the bottom-left corner to the up-right corner. Moreover, to facilitate the training process, we ignore the spatial correlation among different cells and only consider the temporal series prediction. The data of different cells are processed separately and independently.

TABLE I
The hyper-parameters of the LSTM network.

Parameters	Values
Number of features	1
Gradient threshold	1
Number of hidden units	100
Maximum epoches	50

Since the traffic volumes of some cells at certain time intervals are missing due to data storage errors, data completion should be done before proceeding to cellular traffic prediction. We first fill the missing values of traffic data by the mean values of its surrounding cells or time intervals, as in [18]. The granularity of the original cellular traffic is 10-minute, which is the determined by the sampling frequency. This fine-granularity of prediction is useful for dynamic resource management among different network slices. The traffic volume is normalized to the interval of $[0, 1]$ to facilitate the training process, and the predicted results is rescaled back to its real value. The whole dataset is divided into two part, where the first 95% constructs the training dataset and the remaining is used for the test dataset.

B. Performance Metrics and Experiment Settings

We use two metrics to evaluate the prediction performance of different schemes. The first one is root mean square error (RMSE), which measures the difference between predicted values and the values of ground truth, i.e.,

$$RMSE = \sqrt{\sum_{t=T+1}^{T+l} \frac{(X_t - \hat{X}_t)^2}{l}}. \tag{10}$$

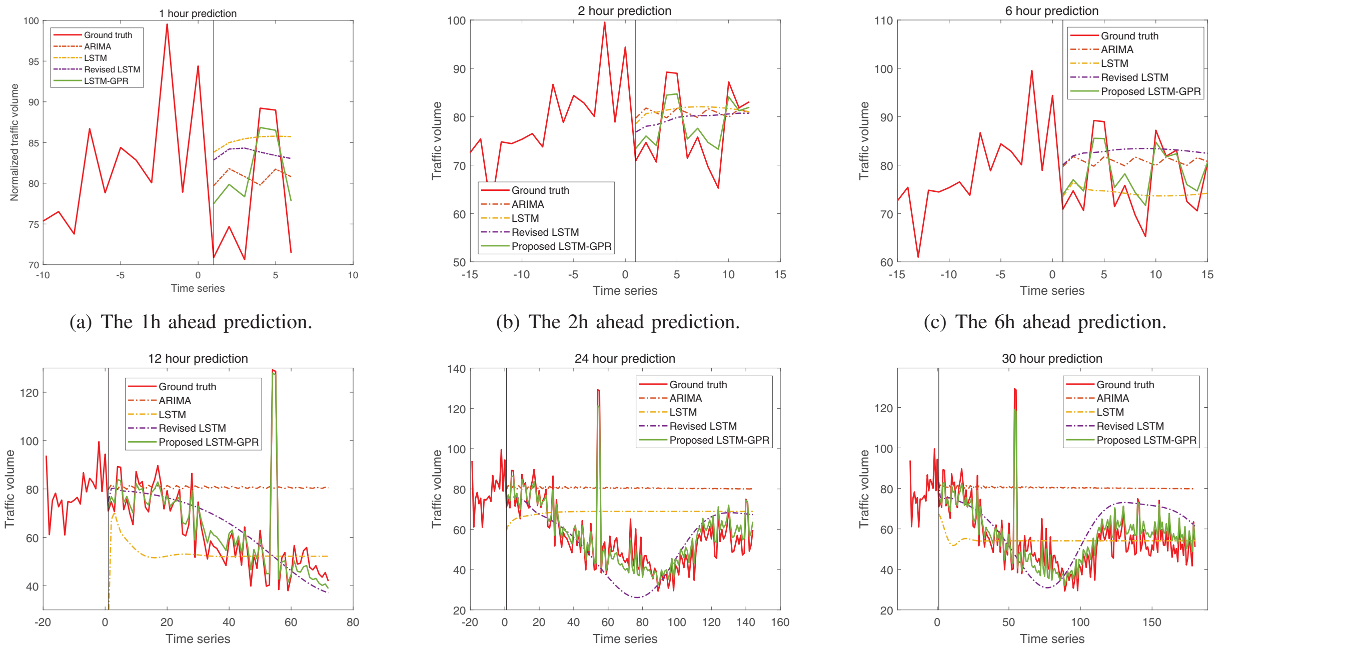
The other one is mean absolute error (MAE), which denotes the average of the absolute differences between prediction and ground truth and is given by

$$MAE = \sum_{t=T+1}^{T+l} \frac{\left| X_t - \hat{X}_t \right|}{l}. \tag{11}$$

To show the performance of the proposed scheme, several benchmark schemes including ARIMA and the conventional LSTM approaches are presented. For the ARIMA model, the model order identification is omitted [19]. During the LSTM training, the hyper-parameters are given in Table I.

C. Prediction Performance

Figure 3 shows the traffic prediction results under different time horizon prediction with different prediction schemes. We observe that the ARIMA can achieve relatively good prediction performance for short time prediction, and the performance deteriorates with an increase of the prediction



(a) The 1h ahead prediction.
(b) The 2h ahead prediction.
(c) The 6h ahead prediction.
(d) The 12h ahead prediction.
(e) The 24h ahead prediction.
(f) The 30h ahead prediction.

Fig. 3. The prediction results of the sequence of cellular traffic with different schemes for cell 375.

horizon. Specifically, when the time horizon is larger than 6 hours, the prediction results of ARIMA scheme converges to a constant value, which is consistent to existing studies. For the LSTM scheme, the prediction performance also deteriorates with the increase of prediction time horizon. Due to the large variance of the cellular traffic, the stochastic characteristics of the training data and test data vary significantly, resulting in large errors during the data standardization process. We also observe that the proposed LSTM-GPR scheme outperforms the other schemes under different prediction time horizons and can properly capture the burst data, due to the GPR compensation step. The RMSE and MAE of different schemes with cell 375 data are given in Table II for 2 hour ahead prediction and 24 hour ahead prediction, respectively. The RMSE and MAE can be reduced by 55.76% and 63.63%, respectively, with the proposed prediction scheme as compared with the ARIMA scheme for 2 hour ahead prediction. For the case of 24 hour ahead prediction, the RMSE and MAE can be reduced by 85.24% and 73.08%, respectively.

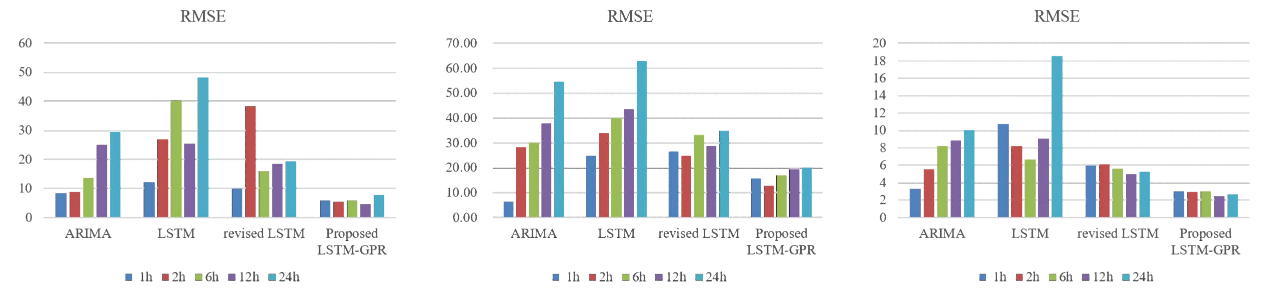
The RMSE and MAE performance of different schemes for various predictions are compared in Figs. 4 and 5. We can observe that, for the other benchmark schemes, both the RMSE and MAE performance deteriorate with an increase of prediction time horizon. Moreover, the performance varies significantly with different cells' data. This is because the cellular traffic volume at different locations varies significantly due to the different user activity patterns. For the proposed LSTM-GPR scheme, it achieves the best RMSE and MAE performance with different prediction horizon values, and thus is more applicable in practice. This is because, after the large periodic components have been extracted, the remaining small random values show similar stochastic properties for different cell data.

TABLE II
THE PERFORMANCE COMPARISON OF DIFFERENT SCHEMES.

Scheme	RMSE		MAE	
	2 h	24 h	2 h	24 h
ARIMA	8.77	29.41	0.11	0.26
LSTM	8.81	21.09	0.13	0.11
Revised LSTM	8.05	16.01	0.09	0.20
Proposed LSTM-GPR	3.88	4.34	0.04	0.07

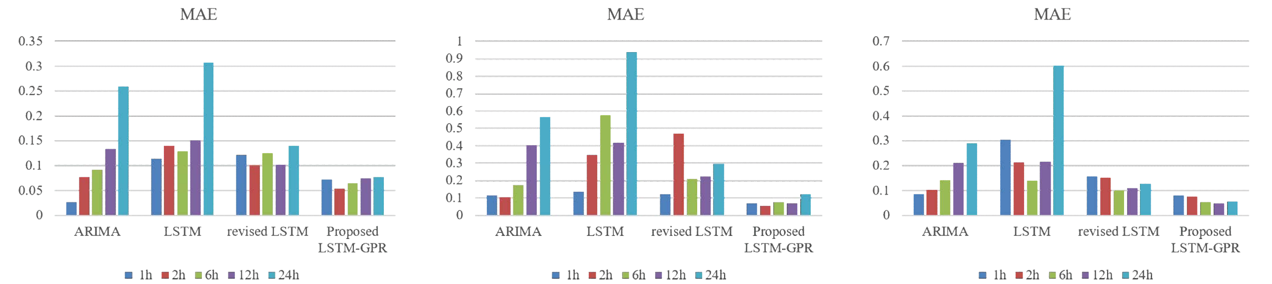
V. CONCLUSIONS

In this paper, we have proposed a machine learning based traffic prediction scheme to predict the cellular traffic volume with large burst data in a relatively large time scale. We have analyzed the cellular traffic periodic property and extracted the dominant periodic components through Fourier analysis. Then, we have employed the LSTM network to learn the long-term dependency in the small random values after subtracting the dominant periodic traffic components. A Gaussian process regression method has also been applied to predict the residual random components raising from the burst traffic. Extensive experiments on open cellular traffic dataset have been conducted to validate the effectiveness of the proposed scheme, and it has been shown that the proposed scheme can significantly improve the prediction accuracy as compared



(a) The RMSE performance with data of cell 375. (b) The RMSE performance with data of cell 250. (c) The RMSE performance with data of cell 23.

Fig. 4. The RMSE performance comparison of different schemes.



(a) The MAE performance with data of cell 375. (b) The MAE performance with data of cell 250. (c) The MAE performance with data of cell 23.

Fig. 5. The MAE performance comparison of different schemes.

with traditional statistical schemes or the conventional LSTM schemes.

In the future, we will consider the spatial-temporal correlation of the traffic to further improve the prediction performance.

References

[1] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, no. 1, pp. 45–66, Jan. 2020.

[2] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Trans. Multim.*, vol. 21, no. 4, pp. 915–929, Apr. 2018.

[3] L. Han, K. Zheng, L. Zhao, X. Wang, and X. Shen, "Short-term road traffic prediction based on deep cluster at large-scale networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12 301–12 313, Dec. 2019.

[4] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2014.

[5] N. B. Hassine, D. Marinca, P. Minet, and D. Barth, "Caching strategies based on popularity prediction in content delivery networks," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, pp. 1–8.

[6] C. Gutterman, E. Grinshpun, S. Sharma, and G. Zussman, "RAN resource usage prediction for a 5G slice broker," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 231–240.

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

[8] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700–1707, 2013.

[9] Y. Shu, M. Yu, O. Yang, J. Liu, and H. Feng, "Wireless traffic modeling and prediction using seasonal ARIMA models," *IEICE Trans. Commun.*, vol. 88, no. 10, pp. 3992–3999, 2005.

[10] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE Conference on Computer Communications (INFOCOM'17)*, 2017, pp. 1–9.

[11] A. Y. Nikravesh, S. A. Ajila, C.-H. Lung, and W. Ding, "Mobile network traffic prediction using MLP, MLPWD, and SVM," in *Proc. IEEE International Congress on Big Data*, 2016, pp. 402–409.

[12] P. Bloomfield, *Fourier analysis of time series: an introduction*. John Wiley & Sons, 2004.

[13] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communi. Mag.*, vol. 57, no. 6, pp. 114–119, June 2019.

[14] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, 2000, pp. 189–194.

[15] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at gaussian process regression through Kalman filtering," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 51–61, July 2013.

[16] M. Al-Shedivat, A. G. Wilson, Y. Saatchi, Z. Hu, and E. P. Xing, "Learning scalable deep kernels with recurrent structure," *Journal of Machine Learning Research*, vol. 18, pp. 1–37, 2017.

[17] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[18] 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, June 2019.

[19] B. Zhou, D. He, and Z. Sun, "Traffic modeling and prediction using ARIMA/GARCH model," in *Modeling and Simulation Tools for Emerging Telecommunication Networks*. Springer, 2006, pp. 101–121.