Hotspots Prediction Based on LSTM Neural Network for Cellular Networks

Lixia Zhou1,*, Xia Chen1,2, Runsha Dong3 and Shan Yang4

1School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China
2Beijing Engineering Research Center of High-speed Railway Broadband Mobile Communications, Beijing, China
3Research and Development Centre of Big Data, China Unicom Network Technology Research Institute, Beijing, China
4China Unicom Smart City Research Institute, Beijing, China

*Corresponding author email: lixiazhou@bjtu.edu.cn

Abstract. The tremendous growth in data traffic usage is a prominent challenge to cellular network operators. The aggregation of a small number of users’ services may lead to considerably high loads on base stations, thus generating traffic hotspots in the networks. To provide better quality of service in dynamic network scenarios efficiently, it is necessary to accurately distinguish and predict potential hotspots, so as to adjust resource configuration and allocation in advance and keep the network running smoothly. In this paper, a network traffic hotspots prediction method is proposed based on the Long Short-Term Memory (LSTM) neural network framework. To verify its effectiveness, real-world data are utilized in our experiments. Numerical results demonstrate that a time series model based on the LSTM framework can achieve 92.2% success rate for future hotspots prediction.

1. Introduction
With the rapid development of communication and Internet technologies, the unprecedented growth of mobile devices such as smartphones and tablets has led to a traffic surge in existing cellular networks. According to the data provided by a forecast report [1], the global mobile data traffic will exceed 30 Exabytes per month by 2020. Such explosive increase in traffic brings great challenges to network operators. At present, the capacity of static network architectures to handle traffic fluctuations in both temporal and spatial dimensions caused by user mobility and aggregation activities still needs to be improved [2]. In the case of special events, such as accidents (service interruption or road congestion), political events (elections or pageants), and entertainment venues (concerts or sport competitions), network load may increase dramatically and the positions of hotspots may also change rapidly [3]. As a result, the macro base stations tend to reach the upper limit of their traffic capacity. In view of this, a common solution is to deploy additional micro base stations in the serving area of the affected macro cells to increase capacity [4]. However, the additional increase in the number of deployed micro base stations will not only increase the operator’s expenditure costs, but also lead to poor timeliness due to its long deployment time. Therefore, it is vital to carefully plan the network growth and manage the usage of network resource reasonably for operators.

In recent years, operators have placed increasing emphases on network optimization, and have carried out extensive investigations to accurately monitor and comprehend network performance dynamics.
To systematically evaluate the complicated network system, a set of key performance indicators (KPIs) are relied on. Generally, KPIs provide observation of each cell sector in a certain time window, usually in minutes or hours [5]. In order to make adjustments to network resource timely when there is a congestion in the network, especially hotspots with underperformance, one particular challenge is to efficiently predict the hotspots in advance.

In this paper, a two-layer Long Short-Term Memory (LSTM) based model is proposed for predicting future hotspots in wireless cellular networks. The model consists of two hidden LSTM layers and a fully connected layer at the top, whose output is the classification result of predicting whether the cell is a hotspot. The rest of paper is summarized as follows. Section II reviews the related work. The collected dataset, data pre-processing methods and LSTM structure are introduced in Section III. Subsequently, LSTM-based time sequence prediction and experimental results are described in Section IV. Section V discusses some issues worth pondering, and the paper is concluded in Section VI.

2. Related Work
Over the past years, the topics on hotspots identification and detection have attracted quite some attention from both industry and academia. For instance, Ewe et al. [6] deployed a mobile pseudo node in the network, which can only receive user information but can’t send information, to obtain the number of users in its coverage in real time. Hence, the mobile user hotspots detection was achieved based on user density. A mobility modelling method for wireless big data was designed by Zhao et al. [7], which analyzed a large amount of traffic data from the perspective of user behaviours to find out user’s mobility features, changing of city hotspots, and their mobility pattern. In addition, the clustering algorithm like $K$-means could obtain clusters formed by different aggregations of users based on their location, where each cluster center corresponds to a hotspot center [8].

On the other hand, some existing works also attempted to forecast hotspots based on historical data. Göndör et al. [9] proposed a Movement Prediction System based on the pattern mining algorithm [10], which allows to determine the future movement of a user using precomputed movement patterns, with the objective to predict user mobility for proactively anticipate traffic hotspots. Another example is the Holt-Winter model [11], which used past traffic data to predict the data of every base station for future time intervals, and then determined hotspots location based on a threshold. Unfortunately, with the proposed method, the average Symmetric Mean Absolute Percentage Error of every base station still reached 33%. Similarly, Nika et al. [12] performed an experimental study on data hotspots using cellular dataset with more than 700k users, their analysis examined the static and dynamic characteristics of hotspots, and the predictability was also evaluated. Indeed, their work was the first to characterize traffic hotspots using real data. But only hotspots duplication at the exact time and location in the following week was considered. Finally, in [13], the authors studied the spatio-temporal patterns of the so-called hotspot score and discovered its regularities, and tree-based machine learning models were proposed to forecast future hotspots. However, it is clear that they did not take account of the context of features in the time series.

3. Dataset and LSTM Model
In this section, cellular dataset used in our analysis and methodology for forecasting traffic hotspots are introduced. Moreover, a simplified description of the basic structure of LSTM neural network is also provided.

3.1. Dataset Description
Our dataset is provided by a Chinese telecommunication service provider, and is divided into a feature set and a target set. The feature set includes real KPIs of 420 cells located in a northern city of China. These KPIs are hourly measurements over a period of three days, with a total of 387 parameters, containing cell_id, time stamps, Internet data usage (both download and upload), the number of successfully established RRC (Radio Resource Control) connection, and so on.

As for the target set, it is a binary label for whether the cell is a hotspot, where the label value 1 is meant for a hotspot, and 0 means the cell is currently having a relatively low traffic. According to [14],
a hotspot can be determined as the traffic demand exceeding a certain threshold, which can choose multiples of the mean scales in proportion to a scaling in the overall traffic by definition. By following the results provided in [11], we choose 5 as the scaling factor in our study.

3.2. Data Pre-processing

Considering the LSTM model only needs to apply simple data pre-processing, several commonly used steps are performed on the feature set as follows:

- **Manual elimination**: Because the original dataset contains a dozen of irrelevant features, they should be removed from set. This affects features such as province_id, flag, and start_time, etc. Both cell_id and enode_id are deleted accordingly, so that the model will not introduce any effect caused by the id’s range. Furthermore, a small portion of the features are all null, which indicates that they are worthless, and also need to be expurgated.

- **Variance threshold**: Before feature selection, the missing values need to be handled due to the fact that KPIs are not always present at every moment. We impute the missing data points using historical mean because the missing parts only account for a small portion of the above processed dataset. The variance threshold filters out the features having the same values primarily.

- **Standardization**: The values of KPIs vary widely, and these differences will have a negative impact on the model’s ability to learn. Thereby, the features fed into the expected model must be standardized to guarantee that they are of the same scale. This pre-processing ensures the parameters can converge steadily in the model proposed in our paper. The standardization formula is defined as:

\[
\chi_{(new)} = \frac{\chi - \bar{\chi}}{\sigma},
\]

where \(\bar{\chi}\) and \(\sigma\) are the average and standard deviation of each feature, respectively.

In short, the remaining 287 features are applied to the proposed LSTM-based model. When we partition the training set and the test set, the first 80% of time series data are used for training, and then rest of data are used to validate the accuracy of the model.

3.3. Long Short-term Memory Network

Long Short-Term Memory (LSTM) neural network, which is a special kind of Recurrent Neural Network (RNN), was initially proposed in 1997 for language models and well-known for its prominent ability to work out long-term dependencies [15]. An LSTM model possesses internal self-looping module in which connections between units compose a direct cycle enabling data to flow both forward and backward in the network [16]. Therefore, compared with traditional machine learning models, LSTM is able to “remember” previous information well and has a strong self-learning ability for time series tasks. Currently, LSTM model has been successfully utilized in the field of Deep Learning and Artificial Intelligence, such as language modelling, automatic image captioning, machine translation and speech recognition.

![Figure 1. LSTM structure of neural network model.](image)

Similar to RNN, LSTM also has chain-like structure, while its repeating module is more sophisticated. Particularly, the LSTM architecture prevents the so-called vanishing or exploding gradients problem
when the Back Propagation Through Time (BPTT) is applied to the training progress. The typical structure of an LSTM is shown in figure 1, where the main elements are a cell state and three gates. The cell state $C_t$ memorizes information about the past, and the three gates are particular multiplicative units named forget gate, input gate, and output gate, respectively. Each gate outputs a value between 0 and 1, where the value 0 means no information is allowed to pass, and the value 1 means all information is allowed to pass. They control the flow of information within the memory block primarily, that is, optionally allowing information to go through. Specifically, the forget gate $f_t$ controls what information should be thrown away from the cell state and thus selects the optimal time lag for input sequence. The input gate $i_t$ controls what new input information should be added to the cell state, while the output gate $o_t$ controls what parts of the cell state should be read and output into the next network.

The input time series are denoted as $X = (x_1, x_2, \ldots, x_T)$, the output vector $h_t$ of the LSTM memory block at the time step $t$ is calculated by using following formulas:

$$
f_t = \sigma(W_fx_t + U_fh_{t-1} + b_f),
$$

$$
l_t = \sigma(W_ix_t + U_ih_{t-1} + b_i),
$$

$$
o_t = \sigma(W_ox_t + U_oh_{t-1} + b_o),
$$

$$
C'_t = \tanh(W_cx_t + U_ch_{t-1} + b_c),
$$

$$
C_t = f_t \odot C_{t-1} + i_t \odot C'_t,
$$

$$
h_t = o_t \odot \tanh C_t,
$$

where $W_f, W_i, W_o$ represent the matrix of weights from input vector to the forget, input, output gate, respectively. Accordingly, $U_f, U_i, U_o$ represent the matrix of weights from last hidden state to the forget, input, output gate, respectively. $b_f, b_i, b_o$ denote the bias vectors of the forget, input, and output gate, respectively. $\sigma$ stands for a non-linear activation function: sigmoid, and the element-wise multiplication of two vectors is denoted with $\odot$.

**4. LSTM-based Prediction**

In this work, the measurements’ history of the cells over the past 24 hours are used to predict whether they will be hotspots. We design a two-layer LSTM and a fully connected layer at the top based model. The previous information can be saved by the LSTM layers, which is useful for enhancing the ability of the model to learn time series data. A fully connected layer contributes to improve the model’s fitting ability. Furthermore, the neurons in the LSTM layers are treated by dropout so as to prohibit overfitting [18], where the main idea is to randomly make part of neurons not work during the training. The dropout probability is set to 50% during the training eventually.

In our experiment, we select the Adam optimizer with adaptive learning rate, which is the alteration of the stochastic gradient descent (SGD) optimizer, in order to minimize training error and avoid local minimal points at the same time. In addition, the cross entropy function is applied to calculate the loss of the model, which usually combines with the softmax function. The softmax and the cross entropy function are defined below:

$$
S_i = \frac{e^{x_i}}{\sum e^{x_i}},
$$

$$
L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log y_i + (1 - y_i) \log(1 - y_i)].
$$

As demonstrated in the above formulas, obviously, $S_i$ ranges from 0 to 1, while the sum of all $S_i$ equals to 1, which is similar to probabilities. In the loss function, $Y_i$ represents the real label of the $i$-th sample, $y_i$ denotes the predicted value, $N$ is the samples size. The model updates its parameters along the direction of loss reduction. For a fair comparison, all of our computation is performed in the same environment: an NVIDIA TESLA K20m GPU.
To explain how the hyper-parameters affect the prediction results, different hyper-parameters are debugged for the proposed model. And two performance metrics are calculated in order to obtain the optimal model hyper-parameters. Results including the training loss and the accuracy in test are displayed in table 1, and the optimal hyper-parameters are shown in the first row. From table 1 it can be learned that too many training iterations may not ensure better results, such as an increase of only 0.1% in the accuracy, but as far as we know, it needs take much longer time to train; while too few iterations may make the model learning inadequate. In addition, too many hidden neurons in the LSTM layer are computationally expensive and often lead to overfitting, meanwhile, having an insufficient number of neurons may decrease the network’s learning ability. Therefore, 300 hidden neurons are chosen during our implementation. Moreover, it is optimal to choose a dropout probability to 0.5, while the model is prone to overfitting without dropout. This phenomenon also is depicted in figure 2, it is clear that the test accuracies do not fit training progress very well.

Table 1. Results of the proposed model with different hyper-parameters.

| Hidden neurons | Iterations | Dropout | Loss  | Accuracy |
|----------------|------------|---------|-------|----------|
| 300            | 215000     | 0.5     | 1.260 | 0.922    |
| 300            | 300000     | 0.5     | 1.105 | 0.923    |
| 300            | 107000     | 0.5     | 10.137| 0.874    |
| 300            | 215000     | 0.0     | 1.271 | 0.882    |
| 300            | 215000     | 0.25    | 1.269 | 0.901    |
| 500            | 215000     | 0.5     | 1.293 | 0.893    |
| 100            | 215000     | 0.5     | 1.352 | 0.876    |

Figure 2. Training progress without dropout.

In order to compare the proposed model with the traditional feed-forward neural network (FFNN) model, the FFNN model is also trained with the same hyper-parameters as mentioned above. What’s more, the precision, recall and F1 score are also added as evaluation indicators to comprehensively evaluate the performance of the prediction model. Results are summarized in table 2, one can see that the proposed model reaches more than 92% in the accuracy, precision and recall score. The four indicators of the proposed model are much higher than the FFNN model, which are only about 75%. These results indicate that the proposed model can achieve a high success rate in hotspots prediction, which reflects the ability of the LSTM network to memorize previous information and its superiority in processing time series data.

Note that the proposed model has two layers of LSTM and one fully connected layer at the top. To prove that the proposed model has sufficient learning ability, it is also compared with other models that have different structural scales, including one-layer (one-LSTM), three-layer (three-LSTM) or even four-layer LSTM structure (four-LSTM). The same hyper-parameters are used to train other models. Except for the above four indicators, the running time of these models is also compared to reflect complexity of the model. Table 2 displays the summarized results. It can be seen that the accuracy of the model with one layer of LSTM is only about 90.8%. When an additional layer of
LSTM is added (i.e. the proposed model), its accuracy increases, which can reach 92.2%. However, the number of layers is too high, such as four hidden LSTM layers, the accuracy is reduced to only about 89.6%. This phenomenon may be caused by overfitting. Besides, the performance of the proposed model and the three-layer LSTM model do not differ much as we can be seen in table 2. The proposed model is only slightly better than the three-LSTM, which may be the result of random initialization parameters. But the three-layer LSTM model needs to take longer for convergence (7.16 min), that is, its computational overhead is more expensive than the proposed model. Therefore, considering the results in table 2, one can come to the conclusion that the proposed model is more effective than the FFNN and other scale structure models in hotspots prediction.

**Table 2.** Comparisons of performance statistics for the proposed model, FFNN, one-LSTM, three-LSTM, and four-LSTM.

| Metrics     | Proposed model | FFNN   | One-LSTM | Three-LSTM | Four-LSTM |
|-------------|----------------|--------|----------|------------|-----------|
| Accuracy    | 92.16%         | 75.73% | 90.84%   | 91.24%     | 89.62%    |
| Precision   | 92.02%         | 70.49% | 91.16%   | 91.22%     | 90.29%    |
| Recall      | 92.16%         | 75.72% | 90.86%   | 91.24%     | 89.58%    |
| F1 score    | 91.93%         | 73.01% | 90.97%   | 91.23%     | 89.82%    |
| Time (min)  | 5.18           | --     | 3.25     | 7.16       | 8.81      |

Furthermore, we observe the presentation of the proposed model and the three-LSTM in detail, respectively. Specifically, the relationship between loss function and training iteration in the proposed model is depicted in figure 3. From figure 3, one can see that the training loss dropped from about 260 to nearly 1.26, which is ideal. Besides, for the proposed model, the accuracies of training and test for hotspots prediction are shown in figure 4. The training progress for accuracies about the three-LSTM is described in figure 5. From figure 4, we find that the prediction accuracies fitted well on both training and test set. It proves that dropout can effectively prevent overfitting and enhance the generalization ability of the model. The result in figure 5 is similar to the proposed model extremely, which is coincident with the fact reflected in table 2. Moreover, another indicator is also applied to the proposed model, namely the Receiver Operating Characteristic (ROC) curve to more intuitively reflect the experimental precision of the proposed model. In figure 6, it is clear that the ROC curve is close to point (1, 0), which means the model achieves good prediction accuracy.

**Figure 3.** Loss of training.  
**Figure 4.** Comparison of accuracies for the proposed model.
In addition, we also attempt to implement network traffic prediction, which has been achieved in existing research, and then convert the predicted traffic values into corresponding hotspot labels according to the hotspot threshold (indirect prediction). Both the model structure and parameters are based on the optimal selections of the direct hotspots prediction experiments (direct prediction) described above. The experimental results are correlated with real hotspot labels to generate relevant accuracy, precision, recall, and F1 score. They are compared with the previous best results displaying in table 3, where one can clearly see that the performance of the direct hotspots prediction is better than that of the indirect prediction. And its running time is slightly longer in the indirect prediction, which may be an additional increase for converting the traffic values to the hotspot labels. So for the two approaches, it is more preferable to predict hotspots directly, which not only takes less time but also is more explicit about the hotspots.

Table 3. Comparison of direct hotspots prediction and indirect prediction results.

| Metrics          | Accuracy | Precision | Recall | F1 score | Time (min) |
|------------------|----------|-----------|--------|----------|------------|
| Direct prediction| 92.16%   | 92.02%    | 92.16% | 91.93%   | 5.18       |
| Indirect prediction| 90.55%  | 90.46%    | 90.54% | 90.50%   | 5.25       |

5. Discussion
The above experimental results indicate that the proposed model works well, but there are still some questions needed to be further considered. For instance, based on the previous research, we chose the five-fold means of total cell traffic as the hotspot threshold. However, the theoretical proof of scale factor selection is still missing. What will happen if one chooses other values? One intuitive understanding is that it would affect the quantity of cells determined as hotspots, but without changes of the common attributes of hotspots. Besides, the 287 features were selected from the original dataset to feed into the neural network in our experiment. It is obvious that an increase in the number of features will lead to a more sophisticated model structure. Thus, a new question arises whether selecting these features is the optimal choice. Is it possible to implement other effective methods to reduce the data dimension, so that the complexity of model is reduced but still remain good results? We plan to further study these topics in the near future.

6. Conclusion
Aiming to deal with issue of the hotspots prediction in wireless cellular networks, a new method consisting of two hidden LSTM layers and one fully connected layer is proposed in this paper. The real data collected by the network operator are fed into the neural network. Compared with the traditional FFNN model, the experimental results demonstrate that LSTM has more satisfactory performance in time sequence prediction. Besides, the ability of the proposed model to predict hotspots are also evaluated and compared with other models, such as one-LSTM, three-LSTM, and...
even four-LSTM model. It is proved that the proposed model has adequate learning ability. With historical data to predict future hotspots, it has the potential to help handle possible network congestion. The operators may have sufficient time and motivation to adjust resource configuration accordingly.

References

[1] Cisco visual networking index http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[2] Mignardi S and Verdone R 2017 On the performance improvement of a cellular network supported by an unmanned aerial base station 29th International Teletraffic Congress pp 7-12

[3] Yalinz I B, Keyi A E and Yanikomeroglu H 2016 Efficient 3-D placement of an aerial base station in next generation cellular networks International Conference on Communications pp 1-6

[4] Razavi R and Claussen H 2012 Urban small cell deployments: impact on the network energy consumption Wireless Communications and Network Workshops pp 47-52

[5] Kreher R and Gaenger K 2010 Key performance indicators and measurements for LTE radio network optimization ed John Wiley & Sons, Hoboken

[6] Ewe L, Moedinger R and Bakker H 2016 Mobile user hotspot detection in LTE networks by moving pseudo pico cells European Wireless 2016, 22th European Wireless Conference pp 1-6

[7] Zhao Z, Zhang P, Huang H and Zhang X 2017 User mobility modeling based on mobile traffic data collected in real cellular networks 11th International Conference on Signal Processing and Communication Systems (ICSPCS) pp 1-6

[8] Santana T V, Galindo-Serrano A, Savrac B and López S M 2016 Dynamic network configuration: hotspot identification for virtual small cells 2016 International Symposium on Wireless Communication Systems (ISWCS) pp 49-53

[9] Göndör S, Uzun A, Rohrmann T, Tan J and Henniges R 2013 Predicting user mobility in mobile radio networks to proactively anticipate traffic hotspots 2013 International Conference on Mobile Wireless Middleware, Operating Systems, and Applications pp 29-38

[10] Yavas G, Katsaros D, Ulusoy O and Manolopoulos Y 2005 A data mining approach for location prediction in mobile environments Data & Knowledge Engineering vol 54(2) pp 121-146

[11] Zhang X, Zhou F, Ning J, Yu P and Li W 2018 Hotspot localization and prediction in wireless cellular networks via spatial traffic fitting 2018 IEEE/IFIP Network Operations and Management Symposium pp 1-6

[12] Nika A, Ismail A, Zhao B Y, Gaito S, Possi G P and Zheng H 2014 Understanding data hotspots in cellular networks 10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness pp 70-76

[13] Serrà J, Leontioadis I, Karatzoglou A and Papagiannaki K 2017 Hot or not? forecasting cellular network hot spots using sector performance indicators 33rd International Conference on Data Engineering(ICDE) pp 259-270

[14] Klessig H, Suryaprakash V, Blume O, Fehske A, Fettweis G 2014 A framework enabling spatial analysis of mobile traffic hot spots IEEE wireless communications letters vol 3(5) pp 537-540

[15] Hochreiter S and Schmidhuber J 1997 Long short-term memory Neural Computation vol 9(8) pp 1735-1780

[16] Zhang J, Zhu Y, Zhang X, Ye M and Yang J 2018 Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas Journal of Hydrology vol 561 pp 918-929

[17] Hochreiter S, Bengio Y, Frasconi P and Schmidhuber J 2001 Gradient flow in recurrent nets: the difficulty of learning long-term dependencies A field guide to dynamical recurrent neural networks eds Kremer S C and Klen J F

[18] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting Journal of Machine Learning Research vol 15 pp 1929-1958