Traffic Forecasting in Cellular Networks using the LSTM RNN

Anestis Dalgkitsis
Department of Informatics & Telecommunications
University of Western Macedonia
Kozani, Greece
anestisdalgkitsis@gmail.com

Malamati Louta
Department of Informatics & Telecommunications
University of Western Macedonia
Kozani, Greece
louta@uowm.gr

George T. Karetsos
Department Computer Engineering
TEI of Thessaly
Larissa, Greece
karetsos@teithessaly.gr

ABSTRACT

In this work we design and implement a Neural Network that can identify recurrent patterns in various metrics which can be then used for cellular network traffic forecasting. Based on a custom architecture and memory, this Neural Network can handle prediction tasks faster and more accurately in real life scenarios. This approach offers a solution for service providers to enhance cellular network performance, by utilizing effectively the available resources. In order to provide a robust conclusion about the performance and precision of the proposed Neural Network, multiple predictions were made using the same data-set and the results were compared against other similar algorithms from the literature.

KEYWORDS

Cellular Networks, Traffic forecasting, Recurrent Neural Networks, Long-Short Term Memory

ACM Reference format:

Anestis Dalgkitsis, Malamati Louta and George T. Karetsos. 2018. Traffic Forecasting in Cellular Networks using the LSTM RNN. In *Proceedings of Pan-Hellenic Conference on Informatics (PCI'18). ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3291533.3291540

 ACM ISBN 978-1-4503-6610-6/18/11...$15.00
https://doi.org/10.1145/3291533.3291540

1 Introduction

Traffic forecasting is very important for cellular networking service providers, in order to strategically manage their resources in an efficient and future proof way. Resources like available energy and link bandwidth are becoming more and more valuable, due to the exponential increase in cellular data usage. Due to the explosively growing demand for radio access, there is an urgent need to design a traffic-aware energy-efficient networking architecture [1].

There have been several works in the past that study the subscriber generated traffic forecasting in cellular networks. To understand the network usage pattern and subscriber behavior, a large scale comprehensive prediction algorithm comparison and analysis of a variety of metrics must be performed. Such a detailed performance and accuracy study of data traffic forecasting in cellular networking environments is still not available.

Owing to the flexibility of resource allocation and its considerable agility to meet explosively increasing traffic demands [2], traffic-aware networks could be the most suitable future cellular architecture [1], in which traffic prediction acts as one of the dominant factors for on-demand network management [3]. Due to the rich multi-fractal behavior of cellular Internet traffic [4], non-linear forecasting techniques like Artificial Neural Networks (ANNs), can easily outperform model based techniques. Instead of modelling the given data, ANNs are used as an alternative mathematical tool for classification, pattern recognition, predictions and other tasks performed in auto-correlated data. In particular, the Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) architecture remembers values over arbitrary intervals. LSTM is well-suited to classify and predict time series given time lags of unknown size and duration between important events. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods.

A big advantage of applying LSTM RNN in modern cellular networking architectures is that part of their computation can be offloaded to a centralized system. Since a server can perform the

training phase, all base stations can perform predictions in simple embedded hardware, allowing service providers the option to completely automate strategic resource planning in base stations themselves. Obviously, this is energy, resource and cost efficient way for service providers to address the problem of resource planning and allocation. In this work, we analyze and forecast subscriber generated Internet traffic data, collected by a 4G network. Experimental results will be presented and discussed.

The rest of this paper is organized as follows. Section 2, presents related research in traffic modeling and forecasting as well as on the basics of Long-Short Term Memory Recurrent Neural Networks and their advantages for being used in cases like the one presented in this work. Section 3, presents the design and implementation choices for the realization of the traffic forecasting platform. Section 4, gives the performance results of the proposed forecasting scheme over real data and a comparison with similar approaches from the literature. Finally, Section 5 concludes our work.

2 Related Work

Traffic forecasting in cellular networks is becoming increasingly important, as the rapid demand for radio resources requires an energy-efficient, network-driven architecture. Indeed, traffic forecasting allows for a more efficient configuration and management of cellular networks [3].

2.1 Traffic Modelling and Forecasting

The authors in [5] study the behavior of subscribers regarding their generated traffic using data collected over three weeks at hundreds of base stations. In particular a call arrival model and a random walk model are derived that directly model the aggregate load. In [6] the authors study the predictability of user mobility and find a 93% potential of predicting a user's mobility pattern due to its inherent regularity. In [7] the authors provide an extensive analysis of network resource usage and subscriber behavior. It is shown that there exists significant traffic imbalance among both users and base stations. Also a big portion of the subscribers have limited mobility which also exhibits periodicity regarding the places they visit on certain time of the day. In [8] the authors deal with the spatial distribution of the traffic density and a model is proposed that generates large-scale spatial traffic variations that encompasses the characteristics of log-normally distributed and spatially correlated cellular traffic.

The aforementioned results can be employed as training data for forecast models and similar mechanisms. Furthermore energy efficiency can be enhanced if some Base Stations or elements of Base Stations are tuned into sleeping mode when the predicted traffic is negligible, while other Base Stations may expand their coverage in a coordinated manner [9]. In [10] some legacy time-series forecasting models are presented that due to the lack of a dynamic learning mechanism, cannot fit data sequences very well in particular for irregular or non-periodic time-series. In direct comparison, our proposed scheme is capable to adapt accordingly to the given data and it can outperform competing forecasting models, by range and accuracy as it will be shown subsequently. Finally in [11], traffic-aware energy-efficient radio access networking is proposed that can adapt to traffic fluctuations. However, this proposal does not utilize any forecasting model and is studied entirely on theoretical modeling and simulation.

Time series forecasting methodologies used for traffic prediction, fall into two main categories and the combination of them: Linear, Non-Linear or Hybrid models. In the linear models family we have AutoRegressive Moving Average (ARMA) [12] and its variants i.e. AutoRegressive Integrated Moving Average (ARIMA), Fractional AutoRegressive Integrated Moving Average (FARIMA) [4] and Seasonal AutoRegressive Integrated Moving Average (SARIMA). The Non-Linear models family includes the Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) Model, the Artificial Neural Networks (ANNs) [13] and the Support Vector Machines (SVM) [14]. In particular the authors in [14] employ the Support Vector Machine (SVM) method to predict small-scale network traffic. However, SVM is very expensive in terms of time and memory. They mainly use data during a short range of time near a predicted time point Finally we have hybrid models like the ARIMA/GARCH which aim in combining advantages from both families of models [15].

In summary traffic forecasting is based on the periodic similarity of traffic itself and requires a certain amount of previous information to reduce uncertainty.

2.2 Long Short-Term Memory Recurrent Neural Networks

Studies show that non-linear prediction based on Neural Networks, is more appropriate for network traffic forecasting, than linear prediction models [13]. In general, Neural Networks are widely used for modeling and forecasting based on previously observed data and not on a detailed mathematical model. The architecture and parameters of the neural network are determined solely by the data-set. Neural networks consist of interconnected nodes, called neurons. Each connection is characterized by a weight. The neural network comprises several layers of neurons i.e. a) an input layer, b) one or more hidden layers and c) an output layer.

The most popular neural network architecture is the feed- forward flow, in which the information moves through the network only forward, in direction from the input to the output layer. The use of a neural network as a forecasting tool involves two phases: i) the training phase and ii) the forecasting phase.

Through the training phase, the training data-set is presented in the input layer and the neural network parameters are dynamically adjusted in order to achieve the desired output value for the input set. The most commonly used learning algorithm is the back propagation algorithm, where weights are continuously adjusted until the output error falls below the predetermined value [16]. The forecasting phase represents the testing of the neural network. A new input, not included in the training set, is presented to the

neural network and the output is calculated, thus predicting the outcome of the new input data.

A Recurrent Neural Network (RNN) is a class of Neural Networks (NN) where connections between units form a directed graph along a sequence. RNNs can use their internal state to process sequences of inputs. RNNs are able to learn temporal dependencies in the input data, without the need to specify a fixed set of lagged observations. A Long Short-Term Memory (LSTM) Network is a type of RNNs that addresses the long-term dependency problem by remembering information for long time periods. LSTM Networks are composed of LSTM units and each LSTM unit is composed of: 1) a Cell, 2) an Input Gate, 3) an Output Gate and 4) a Forget Gate.

The Cell is responsible for remembering values over arbitrary time intervals. Input, Output and Forget Gates compute an activation function of a weighted sum. They act as regulators of the flow of values that goes through the connections of the Neural Network. The addition of sequence is a new dimension to the function being approximated. Instead of mapping inputs to outputs alone, the network is capable of learning a mapping function for the inputs over time to an output. The structure of the typical LSTM unit is shown in figure 2.

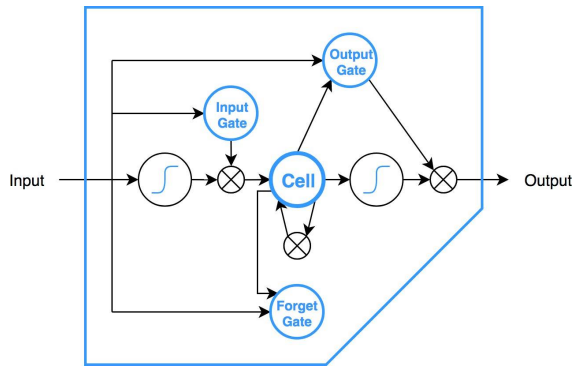

Figure 1: Long Short-Term Memory Unit Structure

Due to the nature of cellular networks, an enormous data-set of measurements can be extracted. That data-set can be used by a single or layers of multiple Neural Networks in order to harvest more detailed and complex conclusions about subscribers' activity and usage trends. The most significant factor to enable this advanced prediction functionality, is an appropriate training set [13]. A small training set can result into a prediction model that cannot extend over new unseen data [17]. On the other hand, a large training set can result into a more accurate prediction model at the expense of a prohibitively high computational cost [13]. Hence, there is a trade-off concerning the selection of the training set for the prediction model.

Linear time-series prediction algorithms like ARIMA and its extensions, seem to be affected by unwanted artifacts in the series [18]. These algorithms use a technique called moving average to calculate the trend in the series and this is directly projected to the forecasted results. Due to their ability to memorize and forget weights [17] according to their activation function, LSTMs can forget and re-evaluate their weights according to the correlation with the rest of the series. This ability makes LSTM RNNs more versatile and resistant to errors, noise and sample gaps. Since unwanted artifacts in the series can cause forecasting algorithms to lose accuracy, data filtering is required before forecasting. Consequently such a provision should be carefully incorporated in the respective forecasting scheme.

3 Neural Network Traffic Forecasting Platform

The main concept behind using Artificial Neural Networks (ANNs) for traffic forecasting, is a centralized overlying system over the current cellular network infrastructure. Consider an Intelligent Agent module located inside every Base Station, that is responsible for monitoring the Base Station's operations and store all necessary data, which will be subsequently used for the training process of the Neural Network.

Then, based on the prediction model, the Intelligent Agent exchange sections of it's extracted measurements to a centralized system, a server, for the training phase and gets back the trained model. The Intelligent Agent data exchange should take place only when the network is underutilized, most likely at dawn. Some times, due to congestion or malfunction, the Intelligent Agent could perform the training phase itself, but with the trade-off lower accuracy. This approach allows for fast training and faster prediction, by offloading the resource demanding task of the training phase to proper centralized hardware and leaves forecasting phase to the energy efficient Intelligent Agent of the base station. Base stations can become completely autonomous as the Intelligent Agent estimates the forthcoming needs for resources and proactively requests their commitment from the back-haul network.

Since this proposed architecture is built as an overlying technology, there is no need to change the structure of the existing network in order to implement such a system. The hardware needed for the proposed traffic forecasting system, consists of the following:

A. Intelligent Agents are small network devices, that collect measurements and execute the training phase of neural network. They use the back-haul network to send collected values to the Service Provider for the training phase and get back the trained model for instantaneous and energy efficient forecasting.

B. Centralized Intelligent System is a server, spatially located near the Service Provider and is used solely for the training phase of the Neural Network. Powerful and efficient, GPU or CPU, hardware can be used to fit the Neural Network of every Intelligent Agent quickly and transfer the trained model back to them for the next phase.

In the event of failure, Intelligent Agents can resolve the problem via certain actions taken autonomously. On the other hand

transition to that model is fast, efficient and safe with respect to the current network.

An initial statistical processing of the collected data and the subsequent selection of the training set can efficiently improve the performance of the prediction model. In similar literature [19], the authors use the notion of the Relative Standard Deviation (RSD) so as to depict and exploit the statistical properties of the collected data. The RSD is a measure of precision regarding the collected data and is defined as:

$$RSD(\%) = \frac{\sigma}{\mu} * 100 \quad (1)$$

Where σ is the standard deviation and μ is the average value of the data set. Practically, small RSD for a set of collected data implies that the measurements are averaged around their mean value, while a high RSD refers to collected data with great variations. The former may correspond, for example, to busy times in a crowded cell where the bandwidth is shared among all the subscribers, while the latter may refer to quiet times where a small amount of subscribers exploits a large portion of the available bandwidth [19]. We split our data-set in even blocks and select the one with the lowest Relative Standard Deviation. This method produces, in most cases, higher forecasting accuracy than the standard way of training the Neural Network with the whole data-set. Multiple libraries were used to reduce the complexity of building the proposed Neural Network. The most important are listed below:

1) TensorFlow: An open-source software library for Machine Intelligence made by Google.
2) Keras: A high-level neural networks API, capable of running on top of TensorFlow, CNTK or Theano.
3) Pandas: An open source library providing high- performance, easy-to-use data structures and data analysis tools.
4) scikit-learn: Simple and efficient tools for data mining and data analysis.
5) NumPy: Fundamental package for scientific computing with Python.
In particular, Google's TensorFlow enabled us to achieve very good performance, even on embedded low-power devices.

According to Keras library documentation, supervised learning data should be divided into input and output components. In a time- series problem, we achieve this division by using the samples from the last time-step as the input and the sample at the current time-step as the output. We require a shift of one step, which will become the input variables. The time-series as it stands will be the output variables. Then we concatenate those two series together to create a Data-Frame, ready for supervised learning. The series will now have a new position at the top without any value. This is called a Not a Number (NaN) which is a numeric data type value representing an undefined or unrepresentable value and will be used in this position. Later, we replace NaN values with zeros, which the LSTM model will have to learn.

In order to proceed, the trend must be removed from the samples, later added back to forecasts to return the prediction of the original scale and calculate an error score in order to evaluate our forecast. A quick and reliable way to remove the trend from our data-set is by differencing the data. Simply, we subtract the previous time-step from the current sample. This provides us with a difference series which corresponds to the changes to the samples from one time-step to the next. Pandas library includes a completely automatic way to implement differencing, but in our proposed application a custom differencing function is implemented. This is preferred for achieving flexibility and further control over data.

Similar to other Neural Networks, LSTM Neural Networks expect data to be within the scale of the activation function used by the network. The default activation function for LSTM Neural Networks is the hyperbolic tangent, which outputs values between 1 and 1. By using scikit-learn transform classes, we transform our data-set to the range [1, 1] using the *MinMaxScaler* class. This class requires data provided in a matrix format with rows and columns, so we reshape our arrays before transforming. The data transformation flow is depicted in figure 2.

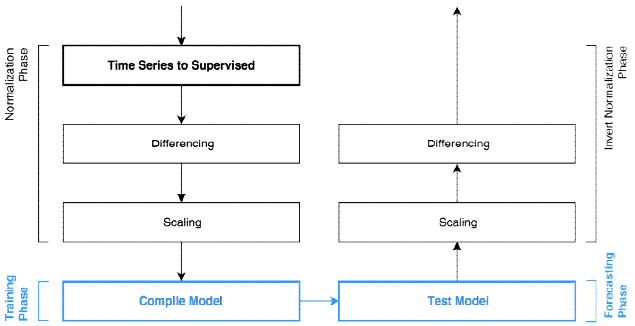

Figure 2: Data transformation flow

The main reason we prefer LSTM over other types of Neural Networks, is because it has the ability to learn and remember over long sequences and does not rely on a pre-specified window of samples as input. Another important parameter used for defining the LSTM layer is the number of neurons. This is a relatively simple problem so any number between 1 and 5 should be sufficient. The number of Neurons is a parameter that is chosen mostly empirically. Keep in mind, that there is a trade-off between optimal training time and forecast accuracy involved in the selection of this parameter. The network requires a single neuron in the output layer with a linear activation to predict the given metric at the next time-step.

Once the network is specified, it must be compiled into an efficient symbolic representation using a backend mathematical library. In this proposal, we use TensorFlow as backend, due its state-of-the-art algorithms employed and flexibility and performance it offers. In order to compile the network, we must specify a loss function and optimization algorithm. In this work we use *mean_squared_error* as the loss function and *adagrad* as an optimization algorithm due to the efficiency it offers in time-

series forecasting. According to Keras documentation *adagrad* optimization algorithm is ideal for RNNs as the LSTM we use. In figure 3 we present the logic diagram of the LSTM RNN training procedure that we employ.

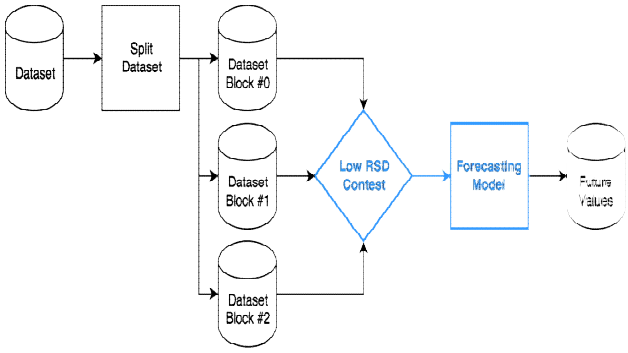

Figure 3: LSTM training procedure

4 Results

In this section we present the results obtained from fitting an available data-set via the implemented LSTM NN and demonstrate its performance.

Subscribers' traffic data were kindly provided by Vodafone, which is a major mobile operator. User personal information were removed from the data-set, in respect to user privacy. The data-set corresponds to traffic logs from 573, 4th Generation, Network Base Stations spatially distributed in the Island of Crete. All data were extracted from March 2016 till June of the same year i.e. for a period of four months.

First we select a random Base Station from our data-set and forecast it's future throughput values. In the beginning we split a set of samples corresponding to 122 days in half. The first 61 days are used for training and the last 61 are the forecasted results used for comparison. We also divide the first 61 days in 5 data blocks and select the one with the lowest Relative Standard Deviation (RSD), as stated in the previous section. That procedure ensures us that we train our model with the most condensed data, which are averaged around their mean value. A part of the obtained results is shown in figure 4.

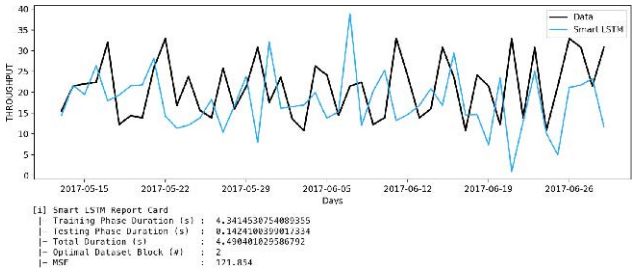

Figure 4: Long-term traffic forecasting at a single BS

From the results, we conclude that the proposed forecasting mechanism continues to forecast the expected traffic with high accuracy for almost 60 days without knowledge any of new samples.

As a next experiment we made a week long forecast of an other Base Station, by splitting 114 days in 4 data blocks and training the Neural Network with the one that has the lowest again RSD. The obtained fitting is shown in figure 5. Obviously, this forecast attempt looks almost identical to the real data and the MSE is minimal. This is because training data-set that we used is longer and thus it returns even more accurate results compared to the one presented in figure 4.

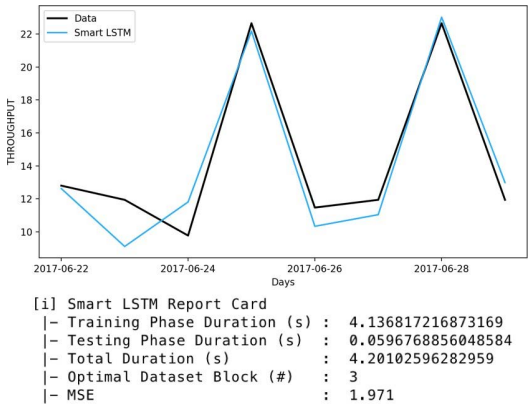

Figure 5: Short-term traffic forecasting at a single BS with longer data-blocks.

The forecasts presented in figure 6 are from two entirely different Base Stations, but share some similarities in their behavior. They are again a week long like the previous forecasts but the data-blocks employed are shorter. Despite the fact that the top forecast belongs to a Base Station located in a more popular location than the bottom one, both present accurate forecasts except from the fact that they fail to predict a sudden transition. This means that a longer data-set with a higher sample rate can train the neural network better so that it can make predictions with higher accuracy.

Since we used the Mean Squared Error (MSE) to calculate the deviation between forecasted and real values, we can compare the accuracy of other techniques against the proposed one. In particular, in table 1, we provide the MSE achieved and the time required to accomplish the forecasting procedure among the Radial Basis Function (RBF) of the Support Vector Machine (SVM), the Linear ARIMA, the Seasonal ARIMA (SARIMA) and LSTM.

SARIMA provides less accurate results in less time than ARIMA and RBF gives more accurate results in less time than Seasonal and Linear ARIMA. The proposed technique outperforms all the relative forecasting models in both accuracy and execution time. Since it can produce more accurate guesses for longer time periods of time and the training of the data-set can be performed in a fast and more hardware efficient way, the proposed mechanism appears to be the best option.

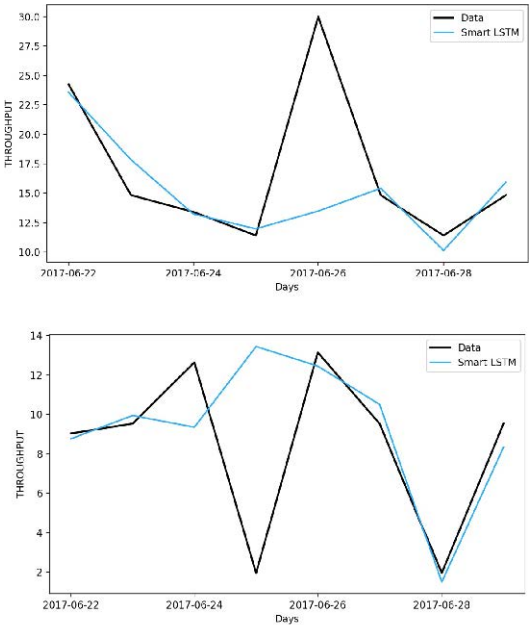

Figure 6: Short-term traffic forecasting at a single BS with shorter data-blocks

Table 1: Comparative results among different forecasting techniques

```
[i] Benchmark Report
 |- RBF     MSE: 3.24234140323 @ 18.16293692588806 (s)
 |- ARIMA   MSE: 6.53373309305 @ 25.01144289970398 (s)
 |- SARIMAX MSE: 11.2641659602 @ 9.040218114852905 (s)
 |- LSTM    MSE: 1.68576362799 @ 0.23576593399047852 (s)
```

5 Conclusions

In this paper we studied, implemented and evaluated a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) for forecasting traffic in cellular networks. The main advantage that our proposed solution has, is that once the Neural Network (NN) is trained, forecasting is instant, even on low power hardware. Furthermore it outperforms similar forecasting techniques. Another benefit of our scheme, is that the demanding part of the forecasting procedure can be offloaded to a server. Since training is performed at a centralized system, all Base Stations can perform forecasts in simple embedded hardware, thus allowing service providers the option to completely automate strategic resource planning in Base Stations themselves. That offers, an energy, resource and cost efficient way for service providers to address the problem of resource planning and allocation. Forecasting cellular network traffic data, can have a crucial impact on the operation and Quality of Service (QoS) provided to the users. In a future work, we intend to take into account seasonality and spatial distribution in order to achieve an even more accurate insight in future traffic predictions.

REFERENCES

[1] E. Oh and B. Krishnamachari. Energy savings through dynamic base station switching in cellular wireless access networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, Dec 2010.

[2] X. Zhou, Z. Zhao, R. Li, Yifan Zhou, and H. Zhang. The predictability of cellular networks traffic. In *2012 International Symposium on Communications and Information Technologies (ISCIT)*, pages 973–978, Oct 2012.

[3] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. 51:114–119, 02 2013.

[4] Jian Liu. *Fractal network traffic analysis with applications*. PhD thesis, School of Electrical and Computer Engineering at Georgia Institute of Technology, 2006.

[5] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary users in cellular networks: A large-scale measurement study. In *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 1–11, Oct 2008.

[6] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi. Limits of predictability in human mobility. 327:1018–21, 02 2010.

[7] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das. Under- standing traffic dynamics in cellular data network. In *2011 Proceedings IEEE INFOCOM*, pages 882–890, April 2011.

[8] Xiaofeng Zhong Zhisheng Niu Xuan Zhou Honggang Zhang Dongheon Lee, Sheng Zhou. Spatial modeling of the traffic density in cellular networks. *IEEE Wireless Communications*, 21(1):88, February 2014.

[9] Eunsung Oh, Bhaskar Krishnamachari, Xin Liu, and Zhisheng Niu. Toward dynamic energy-efficient operation of cellular network infras- tructure. 49:56–61, 06 2011.

[10] A. Adas. Traffic models in broadband networks. *IEEE Communications Magazine*, 35(7):82–89, Jul 1997.

[11] Z. Niu. Tango: Traffic-aware network planning and green operation. *IEEE Wireless Communications*, 18(5):25, October 2011.

[12] M. Joshi and T. H. Hadi, "A review of network traffic analysis and prediction techniques," *CoRR*, vol. abs/1507.05722, 2015..

[13] Andrei B. Rus Virgil Dobrota Melinda Barabas, Georgeta Boanea. Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition.

[14] Meng Qing-Fang, Chen Yue-Hui, and Peng Yu-Hua. Small-time scale network traffic prediction based on a local support vector machine regression model. 18:2194, 06 2009.

[15] Bo Zhou, Z Sun, and Wee Hock Ng. Network traffic modeling and prediction with arima/garch. 07 2008.

[16] G Thimm, P Moerland, and E Fiesler. The interchangeability of learning rate and gain in backpropagation neural networks. 8:451–60, 03 1996.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. 15:1929–1958, 06 2014.

[18] A S. Lapedes and Robert Farber. Nonlinear signal processing using neural networks: Prediction and system modelling, 06 1987.

[19] Ioannis Loumiotis, Evgenia Adamopoulou, Konstantinos Demestichas, Theodora Stamatiadi, and M Theologou. On the predictability of next generation mobile network traffic using artificial neural networks. 28, 12 2013.