Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA

Amin Azari[1](✉), Panagiotis Papapetrou[1], Stojan Denic[2], and Gunnar Peters[2]

[1] Department of Computer and Systems Sciences, Stockholm University,
Stockholm, Sweden
{amin.azari,panagiotis}@dsv.su.se

[2] Huawei, Stockholm, Sweden
{stojan.denic,gunnar.peters}@huawei.com

Abstract. Prediction of user traffic in cellular networks has attracted profound attention for improving the reliability and efficiency of network resource utilization. In this paper, we study the problem of cellular network traffic prediction and classification by employing standard machine learning and statistical learning time series prediction methods, including long short-term memory (LSTM) and autoregressive integrated moving average (ARIMA), respectively. We present an extensive experimental evaluation of the designed tools over a real network traffic dataset. Within this analysis, we explore the impact of different parameters on the effectiveness of the predictions. We further extend our analysis to the problem of network traffic classification and prediction of traffic bursts. The results, on the one hand, demonstrate the superior performance of LSTM over ARIMA in general, especially when the length of the training dataset is large enough and its granularity is fine enough. On the other hand, the results shed light onto the circumstances in which, ARIMA performs close to the optimal with lower complexity.

Keywords: Statistical learning · Machine learning · LSTM · ARIMA · Cellular traffic · Predictive network management

1 Introduction

A major driver for the beyond fifth-generation (5G) wireless networks consists in offering the wide set of cellular services in an energy and cost-efficient way [22]. Toward this end, the legacy design approach, in which resource provisioning and operation control are performed based on the peak traffic scenarios, are substituted with predictive analysis of mobile network traffic and proactive network resource management [5,9,22]. Indeed, in cellular networks with limited and highly expensive time-frequency radio resources, precise prediction of user traffic arrival can contribute significantly in improving the resource utilization [5]. As a result, in recent years, there has been an increasing interest in leveraging

P. Kralj Novak et al. (Eds.): DS 2019, LNAI 11828, pp. 129–144, 2019.
https://doi.org/10.1007/978-3-030-33778-0_11

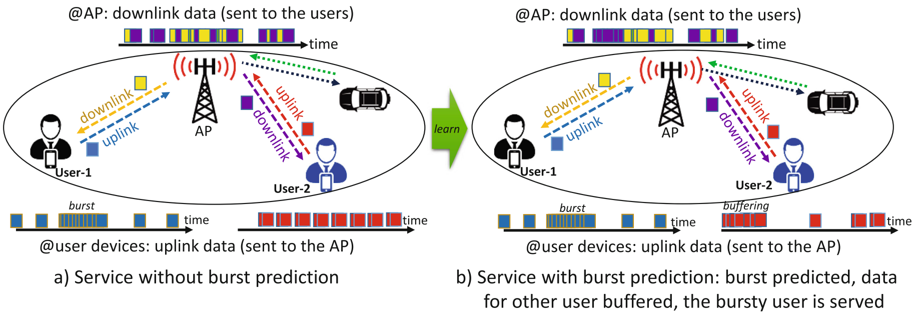
@AP: downlink data (sent to the users)
time
downlink
uplink
AP
uplink
downlink
User-1
User-2
burst
time
time
@user devices: uplink data (sent to the AP)
a) Service without burst prediction
learn
@AP: downlink data (sent to the users)
time
downlink
uplink
AP
uplink
downlink
User-1
User-2
burst
buffering
time
time
@user devices: uplink data (sent to the AP)
b) Service with burst prediction: burst predicted, data for other user buffered, the bursty user is served

Fig. 1. A communication network including access point, users, and uplink and downlink data communications. (a) service is offered without prediction of bursts, (b) service is adapted to the probability of occurrence of bursts.

machine learning tools in analyzing the aggregated traffic served in a service area for optimizing the operation of the network [1,28,30,32]. Scaling of fronthaul and backhaul resources for 5G networks has been investigated in [1] by leveraging methods from recurrent neural networks (RNNs) for traffic estimation. Analysis of cellular traffic for finding anomaly in the performance and provisioning of on-demand resources for compensating such anomalies have been investigated in [32]. Furthermore, prediction of light-traffic periods, and saving energy for access points (APs) through sleeping them in the respective periods has been investigated in [28,30]. Moreover, Light-weight reinforcement learning for figuring out statistics of interfering packet arrival over different wireless channels has been recently explored [4]. While one observes that analysis of the aggregated traffic at the network side is an established field, there is lack of research on the analysis and understanding at the user level, i.e., of the specific users' traffic arrival. In 5G-and-beyond networks, the (i) explosively growing demand for radio access, (ii) intention for serving battery- and radio-limited devices requiring low-cost energy-efficient service [4], and (iii) intention for supporting ultra-reliable low-latency communications [5], mandate studying not only the aggregated traffic arrival from users, but also studying the features of traffic arrival in all users, or at least for critical users. A critical user could be defined as a user whose quality-of-service (QoS) is at risk due to the traffic behavior of other devices, or its behavior affects the QoS of other users. Let us exemplify this challenge in the sequel in the context of cellular networks.

Example. Figure 1(a) represents a communication network in which, an AP is serving users in the uplink (towards AP) and downlink (towards users). One further observes that traffic from user-2 represents a semi-stable shape, which is usually the case in video streaming, while the traffic from user-1 represents a bursty shape, which could be the case in surfing and on-demand file download. One observes that once a burst in traffic of user-1 occurs, the server (i.e. AP) will have difficulty in serving both users in a timely manner, and hence, QoS degradation occurs. Figure 1(a) represents a similar network in which, AP predicts the arrival of burst to user-1, immediately fills the buffer of user-2. Thus, at the time of arrival of burst for user-1, user-2 will require minimal data transfer from the AP, and hence, QoS degradation for user-2 will be prevented. Backed to this motivation, the remainder of this paper is dedicated to investigating the feasibility of exploiting the traffic history at the user level and employing it for future traffic prediction via machine learning and statistical learning approaches.

Research Problem. Let us assume time in our problem is quantized into intervals of length τ seconds. The research problem tackled in this work could be stated as follows: *Given the history of traffic arrival for a certain number of time intervals, how accurately can we estimate (a) the intensity of traffic in the next time intervals, (b) the occurrence of burst in future time intervals (c) the application which is generating the traffic?*

This problem can be approached as a time series forecasting problem, where for example, the number of packet arrivals in each unit of time constitutes the value of the time series at that point. While the literature on time series forecasting using statistical and machine learning approaches is mature, e.g., refer to [24,31] and references herein, finding patterns in the cellular traffic and making the decision based on such prediction is never an easy task due to the following reasons [33]. First, the traffic per device originates from different applications, e.g. surfing, video and audio calling, video streaming, gaming, and etc. Each of these applications could be mixed with another, and could have different modes, making the time series seasonal and mode switching. Second, each application can generate data at least in two modes, in active use and in the background, e.g. for update and synchronization purposes. Third, each user could be in different modes in different hours, days, and months, e.g. the traffic behavior in working days differs significantly from the one in the weekends. Fourth, and finally, the features in the traffic, e.g., the inter-arrival time of packets, vary significantly in traffic -generating applications and activity modes.

Contributions. Our contributions in this paper are summarized as follows:

- We present a comprehensive comparative evaluation for prediction and classification of network traffic; autoregressive integrated moving average (ARIMA) against the long short-term memory (LSTM);
- We investigate how a deep learning model compares with a linear statistical predictor model in terms of short-term and long-term predictive performance, and how additional engineered features, such as the ratio of uplink to downlink packets and protocol used in packet transfer, can improve the predictive performance of LSTM;
- Within these analyses, the impact of different design parameters, including the length of training data, length of future prediction, the feature set used in machine learning, and traffic intensity, on the performance are investigated;
- We further extend our analysis to the classification of the application generating the traffic and prediction of packet and burst arrivals. The results presented in this work pave the way for the design of traffic-aware network planning, resource management, and network security.

The remainder of this paper is organized as follows: In Sect. 2, we outline the related work in the area and introduce the knowledge gaps of state-of-the-art. In Sect. 3, we formulate the problem studied in this paper, while Sect. 4 presents the two methods used for solving it. Section 5 presents the experimental evaluation results for different methods and feature sets, as well as provides a conclusive discussion on the results. Finally, concluding remarks and future direction of research are provided in Sect. 6.

2 Related Work and Research Gap

We summarize state-of-the-art research on cellular traffic prediction and classification, and introduce the research gaps which motivate our work.

Cellular Traffic Prediction. Understanding dynamics of cellular traffic and prediction of future demands are, on the one hand, crucial requirements for improving resource efficiency [5], and on the other hand, are complex problems due to the diverse set of applications that are behind the traffic. Dealing with network traffic prediction as a time series prediction, one may categorize the state-of-the-art proposed schemes into three categories: statistical learning [8,19], machine learning [26,27], and hybrid schemes [12]. ARIMA and LSTM, as two popular methods of statistical learning and machine learning time series forecasting, have been compared in a variety of problems, from economics [10,19,23] to network engineering [6]. A comprehensive survey on cellular traffic prediction schemes, including convolutional and recurrent neural networks, could be found in [13,15]. A deep learning-powered approach for prediction of overall network demand in each region of cities has been proposed in [2]. In [18,27], the spatial and temporal correlations of the cellular traffic in different time periods and neighboring cells, respectively, have been explored using neural networks in order to improve the accuracy of traffic prediction. In [14], convolutional and recurrent neural networks have been combined in order to further capture dynamics of time series, and enhance the prediction performance. In [6,26], preliminary results on network traffic prediction using LSTM have been presented, where the set of features used in the experiment and other technical details are missing. Reviewing the state-of-the-art, one observes there is a lack of research of leveraging advanced learning tools for cellular traffic prediction, selection of adequate features, especially when it comes to each user with a specific set of applications and behaviors.

Cellular Traffic Classification. Traffic classification has been a hot topic in computer/communication networks for more than two decades due to its vastly diverse applications in resource provisioning, billing and service prioritization, and security and anomaly detection [20,29]. While different statistical and machine learning tools have been used till now for traffic classification, e.g. refer to [16] and references herein, most of these works are dependent upon features which are either not available in encrypted traffic, or cannot be extracted in real-time, e.g. port number and payload data [16,20]. In [25], classification of traffic using convolutional neural network using 1400 packet-based features as well as network flow features has been investigated for classification of encrypted traffic, which is too complex for a cellular network to be used for each user. Reviewing the state-of-the-art reveals that there is a need for investigation of low-complex scalable cellular traffic classification schemes (i) without looking into the packets, due to encryption and latency, (ii) without analyzing the inter-packet arrival for all packets, due to latency and complexity, and (iii) with a few numbers of features as possible. This research gap is addressed in this work (Fig. 3).

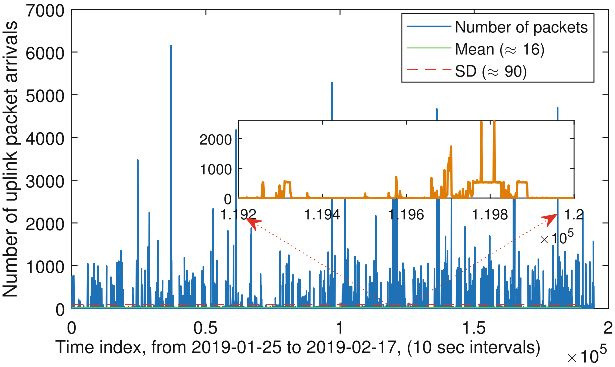

Fig. 2. The number of uplink packet arrivals for 24 days in 10-s intervals

3 Problem Description and Traffic Prediction Framework

In this section, we first provide our problem setup and formulate the research problem addressed in the paper. Then, we present the overall structure of the traffic prediction framework, which is introduced in this work.

We consider a cellular device, on which a set of applications, denoted by $\mathbf{A}$, are running, e.g., User-1 in Fig. 1. At a given time interval $[t, t+\tau]$ of length τ, each application could be in an *active* or *background* mode, based on the user behaviour. We further consider a set of features describing the aggregated cellular traffic in $[t, t+\tau]$ for a specific user, such as the overall number of uplink/downlink packets and the overall size of uplink/downlink packets, which don't require decoding the packets. Let vector $\mathbf{x}_i(t)$ denote the set of features describing the traffic in interval $[t-i\tau, t-(i-1)\tau]$ for $i \geq 1$, and in interval $[t-(i+1)\tau, t-i\tau]$ for $i < 0$ respectively. Furthermore, $\mathbf{X}_m(t)$ is a matrix containing m feature vectors of the traffic, including $\mathbf{x}_1(t)$:$\mathbf{x}_m(t)$ for $m > 0$, and $\mathbf{x}_{-1}(t)$:$\mathbf{x}_{-m}(t)$ for $m < 0$. Further, denote by $\mathbf{s}$ an indicator vector, with elements either 0 or 1. Then, given a matrix $\mathbf{X}_m(t)$ and a binary indicator vector $\mathbf{s}$, we define $\mathbf{X}_m^s(t)$ the submatrix of $\mathbf{X}_m(t)$, such that all respective rows, for which $\mathbf{s}$ indicates a zero value, are removed. For example, let $\mathbf{X}_m(t) = [1, 2; 3, 4]$ and $s = [1, 0]$. Then, $\mathbf{X}_m^s(t) = [1, 2]$.

Now, the research question in Sect. 1 could be rewritten as:

$$\begin{aligned} &\text{Given} \quad \mathbf{X}_m(t), m \geq 1; \\ &\text{minimize} \quad L\big(\mathbf{X}^{\mathbf{s}}_{-n}(t), \mathbf{Y}(t)\big) \end{aligned} \tag{1}$$

where $n > 0$ is the length of the future predictions, e.g., $n = 1$ for one τ future prediction, $\mathbf{Y}(t)$ is of the same size as $\mathbf{X}^{\mathbf{s}}_{-n}(t)$ and represents the predicted matrix at time t, while $L(\cdot)$ is the desired error function, e.g., it may compute the mean squared error between $\mathbf{X}^{\mathbf{s}}_{-n}(t)$ and $\mathbf{Y}(t)$.

4 Time Series Prediction

In this section, we give a short description of the two methods benchmarked in this paper to be used within the proposed prediction framework in Sect. 4.1.

4.1 The Proposed Traffic Prediction Framework

Recall the challenges described in the previous section on the prediction of cellular traffic, where the major challenge consists of dependency of traffic arrival to user behavior and type of the application(s) generating the traffic. Then, as part of the solution to this problem, one may first predict the application(s) in use and behavior of the user, and then use them as extra features in the solution. This approach for solving (1) has been illustrated in Fig. 3. In order to realize such a framework, it is of crucial importance to first evaluate the traffic predictability and classification using only statistics of traffic with granularity τ, and then, to investigate hybrid models for augmenting predictors by online classifications, and finally to investigate traffic-aware network management design.

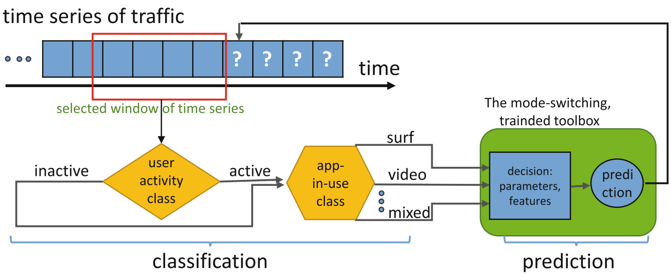

Fig. 3. The proposed framework for cellular traffic prediction

4.2 Statistical Learning: ARIMA

The first method we consider in our work is Autoregressive integrated moving average (ARIMA), which is essentially a statistical regression model. The predictions performed by ARIMA are based on considering the lagged values of a given time series, while at the same time accommodating non-stationarity. ARIMA is one of the most popular linear models in statistical learning for time series forecasting, originating from three models: the autoregressive (AR) model, the moving average (MA) model, and their combination, ARMA [7].

More concretely, let $\mathcal{X} = X_1, \ldots, X_n$ define a uni-variate time series, with $X_i \in \mathbb{R}$, for each $i \in [1, n]$. A *p-order* AR model, $AR(p)$, is defined as follows:

$$X_t = c + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \ldots + \alpha_p X_{t-p} + \epsilon_t, \tag{2}$$

where X_t is the predicted value at time t, c is a constant, $\alpha_1, \ldots, \alpha_p$ are the parameters of the model and ϵ_t corresponds to a white noise variable.

In a similar, a q-order moving average process, $MA(q)$, expresses the time series as a linear combination of its current and q previous values:

$$X_t = \mu + \epsilon_t + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + \ldots + \beta_q \epsilon_{t-q}, \tag{3}$$

where μ is the mean of X, $\beta_1, \ldots, \beta_q$ are the model parameters and ϵ_i corresponds to a white noise random variable. The combination of an AR and an MA process coupled with their corresponding p and q order parameters, respectively, defines an ARMA process, denoted as $ARMA(p, q)$, and defined as $X_t = AR(p) + MA(q)$. The original limitation of ARMA is that, by definition, it can only be applied to stationary time series. Nonetheless, non-stationary time series can be stationarized using the d^{th} differentiation process, where the main objective is to eliminate any trends and seasonality, hence stabilizing the mean of the time series. This process is simply executed by computing pairwise differences between consecutive observations. For example, a first-order differentiation is defined as $X_t^{(1)} = X_t - X_{t-1}$, and a second order differentiation is defined as $X_t^{(2)} = X_t^{(1)} - X_{t-1}^{(1)}$. Finally, an ARIMA model, $ARIMA(p, d, q)$, is defined by three parameters p, d, q [17], where p and q correspond to the AR and MA processes, respectively, while d is the number of differentiations performed to the original time series values, that is X_t is converted to $X_t^{(d)} = \nabla^d X_t$, with $X_t^{(d)}$ being the time series value at time t, with differentiation applied d times. The full $ARIMA(p, d, q)$ model is computed as $X_t^{(d)} = \sum_{i=1}^{p} \alpha_i X_{t-i}^{(d)} + \epsilon_t + c + \sum_{i=1}^{q} \beta_q \epsilon_{t-q} + \mu$.

Finding Optimized Parameters. In this study, the ARIMA parameters, including p, d, and q, are optimized by carrying out a grid search over potential values in order to locate the best set of parameters. In experimental results, Fig. 5, we represent the root mean square error (RMSE) results for different ARIMA (p, d, q) configurations, when they are applied to the dataset for prediction of the number of future packet arrivals. From these results and Bayesian information criterion (BIC), the best performance is achieved by ARIMA(6,1,0).

4.3 Machine Learning: LSTM

Next, we consider is a long short-term memory (LSTM) architecture based on a Recurrent Neural Network (RNN), a generalization of the feed forward network model for dealing with sequential data, with the addition of an ongoing internal state serving as a memory buffer for processing sequences. Let $\{X_1, \ldots, X_n\}$ define the input (features) of the RNN, $\{Y_1, \ldots, Y_n\}$ be the set of outputs, and let $\{Y'_1, \ldots, Y'_n\}$ denote the actual time series observations that we aim to predict. For this study the internal state of the network is processed by Gated Recurrent Units (GRU) [11] defined by iterating the following three equations:

$$r_j = sigm([W_r X]_j + [U_r h_{t-1}]_j), \tag{4}$$

$$z_j = sigm([W_z X]_j + [U_z h_{t-1}]_j)), \tag{5}$$

$$h_j^t = z_j h_j^{t-1} + (1 - z_j) h_{new}, \tag{6}$$

$$h_{new}^t = tanh([WX]_j + [U(r \circ h_{t-1})]_j), \text{where} \tag{7}$$

- r_j: a reset gate showing if a previous state is ignored for the j^{th} hidden unit,
- h_{t-1}; the previous hidden internal state h_{t-1},
- W and U: parameter matrices containing weights to be learned,
- z_j: an update gate that determines if a hidden state should be updated,
- h_j^t: the activation function of hidden unit h_j,
- $sigm(\cdot)$: the sigmod function, and
- $\circ$: the Hadamard product.

Finally, the loss function we optimize is the squared error, defined for all inputs as $\mathcal{L} = \sum_{t=1}^{n} (Y_t - Y'_t)^2$. The RNN tools leveraged in this work for traffic prediction consist of 3 layers, including the LSTM layer, with 100 hidden elements, the fully connected (FC) layer, and the regression layer. The regression layer is substituted with the softmax layer in the classification experiments.

5 Experimental Evaluation

In this section, we investigate the performance of the proposed prediction and classification tools over a real cellular dataset.

5.1 Dataset

We generated our own cellular traffic dataset and made part of it available online [3]. The data generation was done by leveraging a packet capture tool, e.g. WireShark, at the user side. Using these tools, packets are captured at the Internet protocol (IP) level. One must note that the cellular traffic is encrypted in layer 2, and hence, the payload of captured traffic is neither accessible nor intended for analysis. The latter is due to the fact that for the realization of a low-complexity low-latency traffic prediction/classification tool, we are interested in achieving the objectives just by looking at the traffic statistics. For generating labels for part of the dataset, to be used for classification, a controlled environment at the user-side is prepared in which, we filter internet connectivity for all applications unless a subset of applications, e.g., Skype. Then, the traffic labels will be generated based on the different filters used at different time intervals. In our study, we focus on **seven packet features**: (i) time of packet arrival/departure, (ii) packet length, (iii) whether the packet is uplink or downlink, (iv) the source IP address, (v) the destination IP address, (vi) the communication protocol, e.g., UDP, and (vii) the encrypted payload, where only the first three features are derived without looking into the header of packets. We experimented with different values for the interval length parameter τ, and for most of our experiments τ was set to 10 s. Table 1 provides the set of features for each time interval in rows, and the subsets of features used in different feature sets (FSs). It is straightforward to infer that τ tunes a tradeoff between complexity and reliability of the prediction. If τ tends to zero, i.e., $\tau = 1\,\text{ms}$, one can predict traffic arrival for the next τ interval with high reliability at the cost of extra effort for keeping track of data with such a fine granularity. On the other hand, when τ tends to seconds or minutes, the complexity and memory needed for prediction decrease, which also results in lower predictive performance during the next intervals.

5.2 Setup

The experimental results in the following sections are presented within 3 categories, i.e., (i) prediction of the number of packet arrivals in future time intervals, (ii) prediction of burst occurrence in future intervals, and (iii) classification of applications generating the traffic. In the first two categories, we performed a comprehensive set of Monte Carlo MATLAB simulations [21], over the data set, varying different data parameters, such as length of the training set, length of future prediction, feature sets used in learning and prediction. Each RMSE result in Fig. 5 for each scheme has been derived by averaging over 37 experiments. In each experiment, each scheme is trained using a training dataset, and then tested over 2000 future time intervals (non-overlapping with the training dataset). For the classification performance evaluation, we have leveraged 16 labeled datasets, each containing traffic from 4 mobile applications. Then, we constructed 16 tests, where in each test, one dataset is used for performance evaluation. The notation of the schemes used in the experiments, extracted from the basic ARIMA and LSTM methods described in Sect. 4, is as follows: (i) AR(1), representing the traffic prediction based on the last observation; (ii) optimized ARIMA, in which the number of lags and coefficients of ARIMA are optimized using a grid search for RMSE minimization; and (iii) LSTM(FS-x), in which FS-x for $x \in \{1, \cdots, 6\}$ represents the feature set used in the LSTM prediction/classification tool. The overall configuration of experiments can be found in Table 2.

Reproducibility. All experiments can be reproduced using the anonymized GDPR-compliance traffic dataset available at the supporting repository [3].

5.3 Empirical Results

In this section, we present the prediction and classification performance results.

Prediction of Traffic Intensity. Figure 5 depicts the RMSE results for different ARIMA and LSTM configurations versus AR(1), when the number of uplink packets in intervals of 10 s is to be estimated. Towards this end, the right y-axis represents the absolute RMSE of AR(1), the left y-axis represents the relative performance of other schemes versus AR(1), and the x-axis represents the standard deviation (SD) of the test dataset. The results are insightful and shed light to the regions in which ARIMA and LSTM perform favorably, as follows. When the SD of traffic from its average value is more than 30% of the long-term SD of the dataset[1], which is almost the case in the active mode of phone usage by human users, LSTM outperforms the benchmark schemes. On the other hand, when there is only infrequent light background traffic, which is the case on the right-end side of Fig. 5, ARIMA outperforms the benchmark schemes. When we average the performance over a 24-days dataset, we observe that LSTM(FS-6), LSTM(FS-5), LSTM(FS-3), and optimized ARIMA outperform the AR(1) by 16%, 14.5%, 14%, and 12%, respectively, for $\tau = 10$ s. Recall that LSTM(FS-6) keeps track of the number of uplink and downlink packets, as well as statistics of the communication protocol used by packets in each time interval, while LSTM(FS-5) does not care about the protocol used by packets. The superior performance of LSTM(FS-6) with regards to LSTM(FS-5), as depicted in Fig. 5, represents that how adding features to the LSTM predictor can further improve the prediction performance in comparison with the linear predictors.

Table 1. Feature sets.

Feature sets (FSs)	1	2	3	4	5	6
Num. of UL packets	1	1	1	1	1	1
Num. of DL packets	1	0	0	1	1	1
Size of UL packets	1	0	0	0	0	0
Size of DL packets	1	0	0	0	0	0
UL/DL packets	1	1	0	1	0	0
Comm. protocol	0	0	0	0	0	1

Table 2. Parameter configuration.

Parameters	Description
Traffic type	Cellular traffic
Capture point	IP layer, device side
Length of dataset	48 days traffic
RNN for prediction (eq. classification)	[LSTM, FC, regression(eq. softmax)]
Time granularity, τ	Default: 10 s

We investigate if LSTM can further outperform the benchmark schemes by increasing time-granularity of the dataset, decreasing length of future observation, and increasing length of the training set. First, let us investigate the performance impact of τ, i.e. the time granularity of dataset. Figure 5 (left) represents the absolute (left y-axis) and rational (right y-axis) RMSE results for the proposed and benchmark schemes as a function of time granularity of

[1] The long-term SD of the dataset is 90.

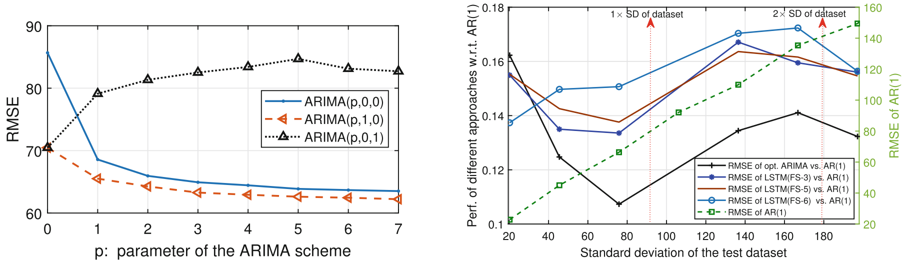

Fig. 4. Prediction of the future number of uplink packets ($\tau = 10\,\text{s}$). (left) Finding optimized ARIMA(p, d, q) configuration; (right) The RMSE performance comparison of LSTM and ARIMA.

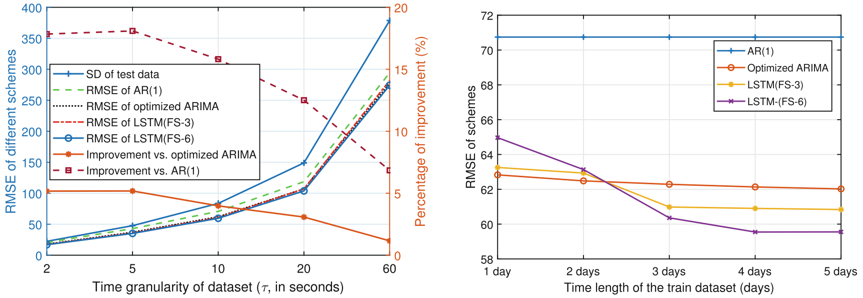

Fig. 5. (left) RMSE of prediction as a function of τ (time granularity of dataset); (right) RMSE of prediction of number of uplink packets as a function of length of the training dataset (as well as length of future prediction).

dataset (τ, the x-axis). One must further consider the fact that τ not only represents how fine we have access to the history of the traffic, but also represents the length of future prediction. It is clear that the best results for the lowest τ, e.g. when $\tau = 1$, the LSTM (FS-6) outperforms the optimized ARIMA by 5% and the AR(1) by 18%. One further observes that by increasing the τ, not only the RMSE increases but also the merits of leveraging predictors decrease, e.g. for $\tau = 60$, LSTM(FS-6) outperforms AR(1) by 7%. Now, we investigate the performance impact of the length of the training set on the prediction in Fig. 5 (right). One observes that the LSTM(FS-6) with poor training (1 day) even performs worse than optimized ARIMA. However, as the length of training data set increases, the RMSE performance for the LSTM predictors, especially for LSTM(FS-3) with further features, decreases significantly.

Prediction of Event Bursts. We investigate the usefulness of the proposed schemes for burst prediction in future time intervals. For the following experiments, we label a subset of time intervals as *bursts*, based on the underlying traffic intensity, i.e., the number and length of packets. Then, based on this training dataset, we aim at predicting whether a burst will occur in the next time interval. As a benchmark to the LSTM predictors, we compare the performance against AR(1), i.e., we estimate a time interval as burst if the previous time interval was labeled as a burst. In Fig. 6 (left) we see the recall of bursty and normal (non-bursty) intervals for a burst definition in which, time intervals with more than 90 uplink packet arrivals are treated as burst when the SD of packet arrivals in the dataset is 90. The LSTM predictor developed in this experiment returns the probability of burst occurrence in the next time interval. In order to declare the decision as burst or non-burst, we set a probability threshold value. The x-axis of Fig. 6 (left) represents the decision threshold, which tunes the weight of recall and accuracy of decisions. In this figure, we observe that the probability of missing a burst is very low on the left side, while the accuracy of decisions is low (it can be inferred from the recall of normal intervals). Furthermore, on the right side of the figure, the probability of missing a burst has decreased, however, the accuracy has increased (high recall of normal intervals). The crossover point, where the recall values of bursty and non-bursty intervals match, could be an interesting point for investigating the prediction performance. In this figure, one observes that when the decision threshold is 0.02, 91% of bursts could be predicted, while only 9% of normal intervals are labeled as bursty (false alarm).

In Fig. 6 (right) we observe some insightful results on the coupling between recall of predictions and degree of rareness of the bursts. The x-axis represents the definition of bursts, e.g. for $x = 90$, we label time intervals with more than 90 packets as a burst. From this figure, it is clear that LSTM outperforms the benchmarks in recalling the burst with a reasonable non-burst recall cost. For example, for $x = 1 (\approx 0.01 SD)$, we aim at predicting if the next time interval will contain a packet or not, i.e., time intervals with a packet transmission are defined as bursts. One observes that 78% of bursts could be predicted using LSTM(FS-5), while only 28% of non-bursts are declared as bursts. Having the information that 20% of time internals contain bursts, we infer that the accuracy of prediction is 78%. As the frequency of burst occurrence decreases, i.e., we move to the right side of the figure, the recall performance of LSTM increases slightly up to some point beyond which, it starts decreasing. On the other hand, the accuracy of prediction by moving from left to right decreases substantially due to the rareness of the bursts. Clearly, LSTM outperforms AR(1), especially when bursts are occurring infrequently.

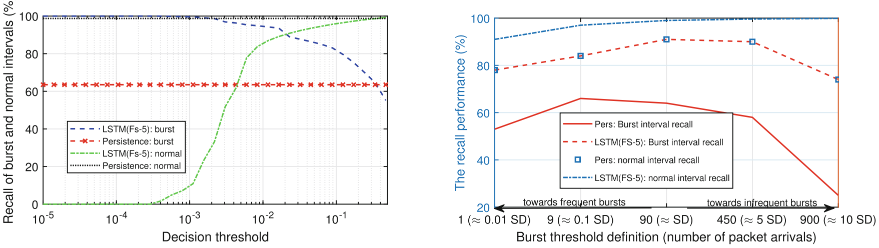

Fig. 6. (left) Prediction of bursts as a function of decision threshold; (right) Prediction of bursts as a function of frequency of occurrence of bursts. ($\tau = 10$ s)

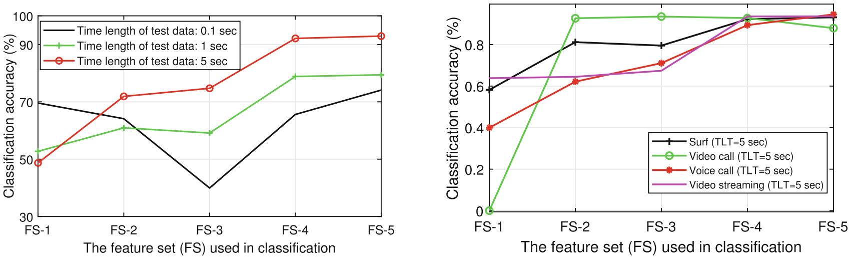

Fig. 7. (left) The overall accuracy of classification as a function of the feature set used in the experiment; (right) Per application accuracy of classification.

Traffic Classification. We investigate leveraging machine learning schemes for classification of the application generating the cellular traffic in this subsection. For the classification purpose, a controlled experiment at the user-side has been carried out in which, 4 popular applications including surfing, video calling, voice calling, and video streaming have been used by the user. Figure 7 (left) represents the overall accuracy of classification for different feature sets used in the machine learning tool. One observes that the LSTM(FS-5) and LSTM(FS) outperform the others significantly in the accuracy of classification. Furthermore, in this Fig. 3 curves for different lengths of the test data, to be classified, have been depicted. For example, when the length of the test data is 0.1 s, the time granularity of dataset (τ) is 0.1 s, and we also predict labels of intervals of length 0.1 s. It is clear here that as the length of τ increases, the classification performance increase because we will have more evidence from the data in the test set to be matched to each class. To further observe the recall of classification for different applications, Fig. 7 (right) represents the accuracy results per each application. One observes that the LSTM(FS-4) and LSTM(FS-5) outperform the others. It is also insightful that adding the ratio of uplink to downlink packets to FS-5, and hence constructing FS-4 (based on Table 1), can make the prediction performance more fair for different applications. It is further insightful to observe that the choice of feature set to be used is sensitive to the application used in the traffic dataset. In other words, FS-3, which benefits from one feature, outperforms the others in the accuracy of classification for video calling, while it results in classification error for other traffic types.

5.4 Discussion

The experimental results represent that the accuracy of prediction strongly depends on the length of the training dataset, time granularity of dataset, length of future prediction, mode of activity of the user (standard deviation of test dataset), and the feature set used in the learning scheme. The results, for example, indicate that the proposed LSTM(FS-3) is performing approximately 5% better than optimized ARIMA, and 18% better than AR(1) for $\tau = 10$ s. The results further indicated that the performance of LSTM could be further improved by designing more features related to the traffic, e.g. the protocol in use for packets, and the ratio of uplink to downlink packets. Moreover, our experiments indicated that the design of a proper loss function, and equivalently the decision threshold, can significantly impact the recall and accuracy performance. Furthermore, we observed that the frequency of occurrence of bursts (definition of burst), the time granularity of dataset, and length of future prediction, can also significantly impact the prediction performance. The results, for example, indicated that a busy interval, i.e. an interval with at least one packet, could be predicted by 78% accuracy as well as recall. The experimental results represented the facts that, first, accuracy and recall performance of classification is highly dependent on the feature set used in the classification. For example, a feature set that can achieve an accuracy of 90% for classification of one application may result in a recall of 10% for another application. Then, the choice of feature set should be in accordance with the set of applications used by the user. Second, if we can tolerate delay in the decision, e.g. 5 s, the classification performance will be much more accurate when we gather more information and decide on longer time intervals. The overall accuracy performance for different applications using the developed classification tool is approximately 90%.

6 Conclusions

In this work, the feasibility of per-user traffic prediction for cellular networks has been investigated. Towards this end, a framework for cellular traffic prediction has been introduced, which leverages statistical/machine learning units for traffic classification and prediction. A comprehensive comparative analysis of prediction tools based on statistical learning, ARIMA, and the one based on machine learning, LSTM, has been carried out, under different traffic circumstances and design parameter selections. The LSTM model, in particular, when the length of training data is long enough and the model is augmented by additional features like the ratio of uplink to downlink packets and the communication protocol used in prior packet transfers, exhibited demonstrable improvement over the benchmark schemes for future traffic predictions. Furthermore, the usefulness of the developed LSTM model for classification of cellular traffic has been investigated, where the results represent high sensitivity of accuracy and recall of classification to the feature set in use. Additional investigations could be performed in the future works regarding making the prediction tool mode-switching, in order to reconfigure the feature set and prediction parameters based on the changes in the behavior of user/applications in an hourly/daily basis.

References

1. Alawe, I., et al.: Smart scaling of the 5G core network: an RNN-based approach. In: Globecom 2018-IEEE Global Communications Conference, pp. 1–6 (2018)

2. Assem, H., Caglayan, B., Buda, T.S., O'Sullivan, D.: ST-DenNetFus: a new deep learning approach for network demand prediction. In: Brefeld, U., et al. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11053, pp. 222–237. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10997-4_14
3. Azari, A.: Cellular traffic analysis (2019). https://github.com/AminAzari/cellular-traffic-analysis
4. Azari, A., Cavdar, C.: Self-organized low-power IoT networks: a distributed learning approach. In: IEEE Globecom 2018 (2018)
5. Azari, A., et al.: Risk-aware resource allocation for URLLC: challenges and strategies with machine learning. IEEE Commun. Mag. **57**, 42–48 (2019)
6. Azzouni, A., Pujolle, G.: NeuTM: a neural network-based framework for traffic matrix prediction in SDN. In: NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–5 (2018)
7. Box, G.E., Jenkins, G.M.: Time Series Analysis: Forecasting and Control, Revised edn. Holden-Day, San Francisco (1976)
8. Brockwell, P.J., Davis, R.A., Calder, M.V.: Introduction to Time Series and Forecasting, vol. 2. Springer, New York (2002). https://doi.org/10.1007/b97391
9. Chen, M., et al.: Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks. arXiv preprint arXiv:1710.02913 (2017)
10. Chen, T., et al.: Multivariate arrival times with recurrent neural networks for personalized demand forecasting. arXiv preprint arXiv:1812.11444 (2018)
11. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014). http://arxiv.org/abs/1406.1078
12. Choi, H.K.: Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. arXiv preprint arXiv:1808.01560 (2018)
13. Huang, C.W., Chiang, C.T., Li, Q.: A study of deep learning networks on mobile traffic forecasting. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1–6 (2017)
14. Lai, G., Chang, W.C., Yang, Y., Liu, H.: Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104 (2018)
15. Li, R., et al.: The prediction analysis of cellular radio access network traffic. IEEE Commun. Mag. **52**(6), 234–240 (2014)
16. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access **5**, 18042–18050 (2017)
17. Mills, T.C.: Time Series Techniques for Economists. Cambridge University Press, New York (1991)
18. Qiu, C., Zhang, Y., Feng, Z., Zhang, P., Cui, S.: Spatio-temporal wireless traffic prediction with recurrent neural network. IEEE Wirel. Commun. Lett. **7**(4), 554–557 (2018)
19. Rebane, J., Karlsson, I., Denic, S., Papapetrou, P.: Seq2seq RNNs and ARIMA Models for cryptocurrency Prediction: A Comparative Study (2018)
20. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: an overview. arXiv preprint arXiv:1810.07906 (2018)
21. Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo Method, vol. 10. Wiley, New York (2016)
22. Saad, W., et al.: A vision of 6G wireless systems: applications, trends, technologies, and open research problems. arXiv preprint arXiv:1902.10265 (2019)

23. Skehin, T., et al.: Day ahead forecasting of FAANG stocks using ARIMA, LSTM networks and wavelets. In: CEUR Workshop Proceedings (2018)
24. Tealab, A.: Time series forecasting using artificial neural networks methodologies. Future Comput. Inf. J. **3**(2), 334–340 (2018)
25. Tong, V., Tran, H.A., Souihi, S., Melouk, A.: A novel QUIC traffic classifier based on convolutional neural networks. In: IEEE International Conference on Global Communications (GlobeCom), pp. 1–6 (2018)
26. Trinh, H.D., Giupponi, L., Dini, P.: Mobile traffic prediction from raw data using LSTM networks. In: 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1827–1832 (2018)
27. Wang, J., et al.: Spatio-temporal modeling and prediction in cellular networks: a big data enabled deep learning approach. In: IEEE Conference on Computer Communications, pp. 1–9 (2017)
28. Wang, K., et al.: Modeling and optimizing the LTE discontinuous reception mechanism under self-similar traffic. IEEE Trans. Veh. Tech. **65**(7), 5595–5610 (2016)
29. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Comput. Commun. Rev. **36**(5), 5–16 (2006)
30. Ye, J., Zhang, Y.J.A.: DRAG: Deep reinforcement learning based base station activation in heterogeneous networks. arXiv preprint arXiv:1809.02159 (2018)
31. Zhang, G.P.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing **50**, 159–175 (2003)
32. Zhang, Q., et al.: Machine learning for predictive on-demand deployment of UAVs for wireless communications. arXiv preprint arXiv:1805.00061 (2018)
33. Zhang, Y., Årvidsson, A.: Understanding the characteristics of cellular data traffic. In: Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design, pp. 13–18 (2012)