

Forecasting Mobile Cellular Traffic Sampled at Different Frequencies

Armin Okic and Alessandro E. C. Redondi

Dip. Elettronica, Informazione e Bioingegneria, Politecnico di Milano

Email: name.surname@polimi.it

Abstract—We consider the problem of forecasting high-frequency sampled mobile cellular traffic starting from a lower-frequency sampled time series. We use a dataset of real downlink/uplink traffic traces obtained from a mobile cellular network and apply different methodologies for performing forecasts at different sampling frequencies. Through extensive evaluation we show that such type of forecasting is possible and in some cases is also able to outperform forecast results obtained starting directly from the high frequency time series. The outcomes of this work can be used for several scenarios of cognitive networking, including prediction of data traffic requests in specific locations, as well as for data storage optimization and improvement of BBU clustering tasks.

Index Terms—Mobile traffic forecasting, Neural Networks, Cognitive Networking, C-RAN

I. INTRODUCTION

The upcoming 5th generation (5G) of mobile networks will support an unprecedented growth in the number of connected devices and the volume of data traffic, thanks to a series of innovation in the way the entire network architecture is designed and operated. In particular, future networks are expected to strongly rely on network function virtualization (NFV) and software defined networking (SDN) at the core, while the radio access part will take advantage of innovative solutions such as Multi-access Edge computing (MEC) and Cloud-RAN (C-RAN). The latter is based on the aggregation of all Base Stations computational resources in a centralized baseband unit (BBU) pool, which is connected to a densely distributed set of Remote Radio Heads (RRH) through high capacity links. Compared to traditional architectures, the C-RAN approach is able to optimally allocate resources in a dynamic way to the individual RRH, whose hardware design is much simpler than legacy LTE eNodeBs. This approach allows to increase network resource utilization, decrease interference and at the same time lowering energy consumption and overall hardware costs [1]. In this scenario, it is envisioned that big data analysis and machine learning/data-driven methodologies will play a major role in all phases of the process. In particular, forecasting future downlink/uplink traffic demands is crucial for C-RAN operations: as an example, when the BBU functions are virtualized using virtual containers (e.g., Docker) which can be scaled up and down dynamically, the initialization time required for such horizontal scaling¹

operations is in the order of 1-5 minutes on modern hardware [2]. Therefore, it is important to forecast short-term traffic demands in order to be able to timely accommodate the needed BBU resources. At the same time, network operators have the common practice of collecting massive datasets from their networks in order to obtain a deep understanding of the communication patterns of users and its implications on social dynamics (user interactions, demographics, epidemics, etc.), user mobility (mobility models, traffic prediction, etc.) and network planning (resource management, energy efficiency, etc.). In particular, when it comes to traffic demands, operators are interested in forecasting traffic at different time scales: as an example, network planning and marketing operations require long term traffic forecasting in order to plan actions in the middle-long term (months or years), while radio access network optimization (including future C-RAN operations), require much finer time granularity. Such heterogeneity in the time scales of data to be handled increases the complexity of data management procedures for the operators, which need to decide how much resources to allocate for storing the massive amount of data coming from the network and at which time granularity [3], [4]. In any case, to limit the amount of storage resources needed, network operators aggregate their datasets either temporally (e.g., computing hourly or daily aggregates of traffic demands for each eNodeB) or spatially (e.g., aggregating data from several eNodeBs in a single traffic series). However, while such aggregations pose a limit on the needed storage resources, they have the drawback of possibly impairing fine-grained forecasting needed for future C-RAN operations. The objective of this paper is to quantify whether is possible to reuse temporally aggregated traffic data to perform short-term forecasting. In particular, we focus on uplink/downlink video traffic demand traces obtained from a real country-wide mobile cellular network and analyze the performance of short-term/high-frequency forecasting (e.g., 5 minutes-ahead predictions) starting from input data which is sampled at a lower frequency (e.g., hourly). We use machine learning methods to perform the forecast operations and comment on the performance obtained compared to other baselines methods commonly used in the literature. The outcomes of this work can be used for several scenarios of cognitive networking, including prediction of data traffic requests in specific locations (eNodeBs), as well as for data storage optimization and improvement of BBU clustering tasks.

¹<https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale>

II. RELATED WORK

The problem of forecasting time series sampled at different frequencies has been already tackled in the past, especially in the field of economics. In that area it is common to encounter time series sampled at both high-frequencies (e.g. quarterly sampled GDP data) or low-frequencies (e.g., annual data), together with the need of forecasting one series given the other. Early works in the area approach the problem using simple interpolation or distribution approaches, where the latter is used when the high-frequency data is ought to sum to the value of the low-frequency data in an observation period (e.g., four quarterly GDP samples sum up to an annual GDP sample) [5], [6], [7]. However, such works generally approach the problem in a formal way, and do not provide insights on the performance of the proposed methods when it comes to a real data forecast. Recent works providing performance analysis of forecast methodologies working at different frequencies can be found in the field of rainfall prediction [8] and tourism demand forecasting [9]. Conversely, increasing attention has been given in the last few years to the problem of predicting mobile network traffic demands. Stimulated by the growing field of context-aware anticipatory mobile networking [10], several studies have proposed different approaches to predict mobile traffic demands, ranging from classical statistical methods to more advanced machine learning-based ones. As an example, in [11] authors use ARIMA models to perform short-term traffic prediction, while the work in [12] applies Support Vector Regression and Multilayer Perceptron to forecast hourly-sampled traffic demands. In [13], authors compare the two different forecasting approaches. Other authors have focused on the capabilities of Neural Networks-based approaches in forecasting mobile traffic: in [14] Recurrent Neural Networks (RNN) are used to predict short/long-term samples of mobile data traffic, showing that RNNs outperform ARIMA based models. In [15] authors create Long-Short Term Memory (LSTM) Neural Networks to predict long-term traffic and show that they outperform ARIMA-based methods and simpler Neural Network approaches. However, to the best of our knowledge none of the works in the area of mobile traffic prediction tackles the problem of forecasting mobile traffic traces sampled at different frequencies: this is precisely the scope of our paper.

III. DATASET

The dataset under analysis in this paper is derived from the work in [16]: the dataset contains uplink/downlink activity from about 125k customers of one of the major European mobile network operators of a middle-sized European city during the entire month of April 2018. An ad-hoc Android application is installed on the user equipments (UE) after explicit consent of their owners: the application runs in background and logs statistics relative to the different applications run by a user, including the uplink/downlink 4G traffic volume. All data is reported to a central server using anonymous identifiers for users and then aggregated every five minutes at the eNodeB level to further ensure not to raise any privacy, ethical or legal

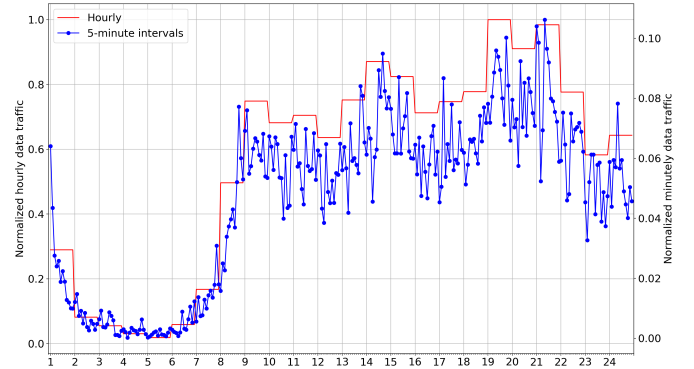


Fig. 1: Sampling of data in hourly and 5-minute intervals.

issues. In this paper, we focus only on uplink/downlink traffic volume produced by video streaming applications installed on the users' terminals (YouTube, Facebook Video, Netflix, etc.) The analysis is carried out on the subset of eNodeBs behaving similarly during the week, i.e., there is no difference in working days and weekends behaviour, in order not to raise further problems of analysis separation into working and weekend days. The traces of each eNodeB are then aggregated in a single traffic trace, similarly to the approach taken by [11] and somehow mimicking a scenario in which the traffic prediction must be performed on aggregated flows at the BBU pool, rather than on each individual base station. We refer to such spatially aggregated, 5-minutes sampled traffic trace as m_t . A lower-frequency sampled traffic series h_k is also created by summing together all samples of m_t falling in the same hour period, according to the following:

$$h_k = \sum_{i=1}^L m_{[(k-1)L+i]}, \quad (1)$$

with $L = 12$ in this case. Figure 1 shows the differences between 5-minutes sampled traffic and hourly aggregated traffic over one day. Data is scaled to the maximum value (note the different scales for the two time series). As one can see, high-frequency traffic data follows the same daily trend of hourly-aggregated traffic and is characterized by higher variability.

IV. PROBLEM STATEMENT

The objective of this paper is to perform prediction on the high-frequency time series m_t , starting from the availability of h_k only. Having in input N samples belonging to h_k (e.g., $h_1 \dots h_N$), we seek to forecast the next high-frequency samples m_t , with $t = NL + i$. We don't restrict the analysis to just one high frequency sample (i.e., $i = 1$), but we extend the study to all possible 5-minutes intervals between two hourly aggregated samples, (i.e., $i = 1 \dots 11$). As explained before, this problem arises in all those cases where a network operator stores low-frequency data (e.g., for storage capacity constraints) but still needs to perform high-frequency forecasts (e.g., for C-RAN optimizations). Two main approaches will be analyzed in this paper: time series distribution and direct

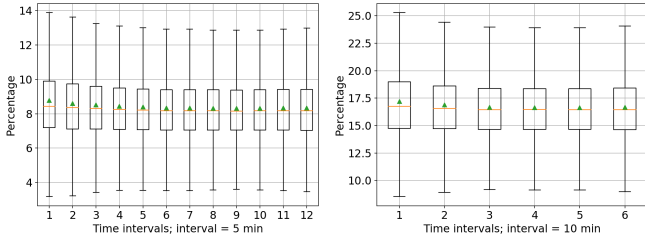


Fig. 2: Distributions of data traffic amounts with different time intervals: 5-minute (left) and 10-minute (right).

forecasting. Furthermore, several baseline approaches are also considered to provide benchmark performance.

A. Time series distribution

A first way of forecasting the higher-frequency series uses the concept of distribution [5], which builds on the assumption that each sample in the low-frequency series is equal to the sum of L higher-frequency samples, as stated in equation (1). Based on this assumption, the process of forecasting may be divided in two steps:

1) *Forecast*: predict the next low-frequency sample h_{N+1} starting from the input samples $h_1 \dots h_N$, using any forecasting method available for time series, e.g., find a function $f_1(\cdot)$ such that $h_{N+1} \sim f_1(h_1 \dots h_N)$. The following alternatives for learning $f_1(\cdot)$ will be explored:

- ARIMA (TSD-ARIMA): a basic option is to use the Autoregressive Integrated Moving Average model approach to predict h_{N+1} , which leads to use a model of the form:

$$h_{N+1} = \mu + \epsilon_{N+1} + \sum_{j=0}^{p-1} \phi_j h_{N-j} + \sum_{j=0}^{q-1} \theta_j \epsilon_{N-j} \quad (2)$$

where μ is a constant, ϵ_{N+1} is a zero mean Gaussian noise whose variance is equal to the variance of errors in the previous q estimated samples and the model parameters ϕ and θ weight the contribution of past samples and errors, respectively. Such parameters can be estimated using the well known Box-Jenkins method, while the order of AR and MA parts, p and q can be found using grid search. In particular we constraint $p \leq N$ to be fair in the usage of past samples across different methods.

- Linear Regression (TSD-MLR): In this case, forecasting is treated as a supervised machine learning problem and a linear regression model is used to predict the next hourly value h_{N+1} , that is:

$$h_{N+1} = \theta_0 + \sum_{j=0}^{N-1} \theta_{j+1} h_{N-j} \quad (3)$$

Parameters θ are estimated so that the training error is minimized.

- Neural Networks (TSD-NN): Recently, neural networks have been used to solve traffic forecasting problems [13], [14], [15]. Here we use a basic version of a Neural

Parameter	Value
Activation function	ReLu
Dropout probability	0.2
Batch size	32
Number of hidden layers	1
Number of nodes	32
Optimizer	Adam
Loss function	RMSE
Number of epochs	100

TABLE I: Neural Network Parameters

Network with N inputs, a single hidden layer with 32 nodes, and hyper-parameters tuned as reported in Table I.

2) *Distribution*: After having predicted the next low frequency value h_{N+1} with one of the aforementioned approaches, its value is distributed on the higher-frequency samples, using a function $f_2(\cdot)$ s.t. $m_t \sim f_2(h_{N+1})$. It is worth to look at how the higher frequency samples are distributed in one larger time interval in order to learn the function $f_2(h_{N+1})$. Figure 2 shows how 5 minutes ($L = 12$) and 10 minutes traffic ($L = 6$) samples are distributed in one hour, considering the entire dataset under analysis. As one can see, and as it is expected, both distributions are uniform: this means that each higher-frequency interval contributes equally to the low-frequency aggregation, so that $f_2(h_{N+1}) = \frac{h_{N+1}}{L}$.

B. Direct forecasting

The second option is to forecast directly the higher-frequency time series starting from the input samples $h_1 \dots h_N$, learning a function $g(\cdot)$ such that $m_t \sim g(h_1 \dots h_N)$. In this case, traditional forecasting algorithm such as ARIMA can't be used since the input and output of the model are sampled at different frequencies. Therefore, we limit the options for learning $g(\cdot)$ to the following:

- Linear Regression (DF-MLR): like before, the problem is treated as a supervised machine learning problem of the form:

$$m_t = \theta_0 + \sum_{j=0}^{N-1} \theta_{j+1} h_{N-j} \quad (4)$$

Here we don't restrict the analysis to $t = NL + 1$, but we train L different models to predict m_t with $t = NL + i$ and $i = 1 \dots L$. This allows to forecast any 5-minute interval in the future, given the past hourly-sampled series.

- Neural Networks (DF-NN): we use the same structure of the neural network reported in Table I and train L different neural networks to forecast the value m_t at any 5-minute interval within each hour.

C. Baselines

In order to provide some benchmarks, we also consider forecasting methods working directly on the high-frequency data inputs, i.e., we consider functions $b(\cdot)$ such that $m_t \sim b(m_{t-N} \dots m_{t-1})$. In this case, the following options for $b(\cdot)$ are considered:

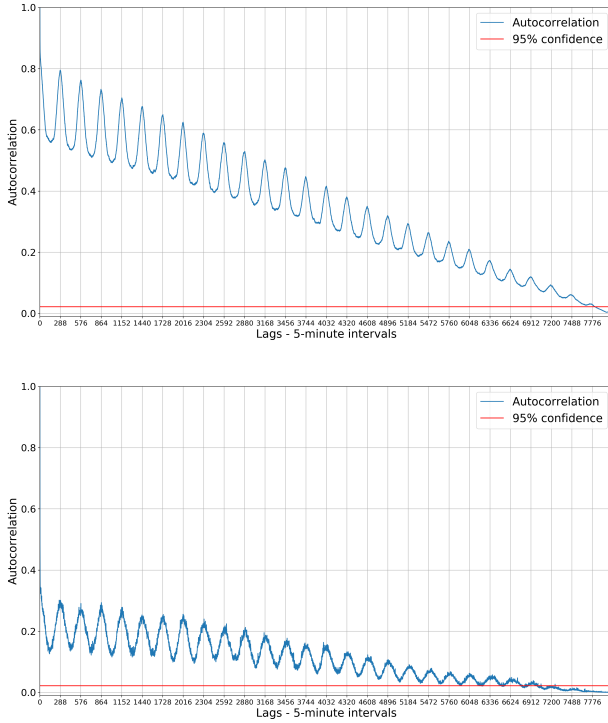


Fig. 3: Auto-correlation function of traffic data for the Downlink (top) and Uplink (bottom) scenario.

- **Last Value:** a simple baseline forecasting the next 5-minutes interval with the last observed value, that is $m_t = m_{t-1}$.
- **ARIMA:** Alternatively, the ARIMA method can be applied directly on the 5-minute series to forecast future values according to the following:

$$m_t = \mu + \epsilon_t + \sum_{j=1}^p \phi_j m_{t-j} + \sum_{j=1}^q \theta_j \epsilon_{t-j} \quad (5)$$

- **Linear Regression:** similar to before, the model used in this case is the following:

$$m_t = \theta_0 + \sum_{j=1}^N \theta_{j+1} m_{t-j} \quad (6)$$

- **Neural Networks:** again, the same Neural Network structure mentioned before is used and trained to perform 5-minute ahead forecasting starting from the 5-minute sampled time series.

V. EXPERIMENTS

Several experiments have been performed in order to compare the performance of TSD, DF and baseline approaches.

A. Data preprocessing

As a first step, the input time series h_k and m_t for both downlink and uplink data have been deseasonalized in order to make them stationary, a condition which is needed to

apply statistical forecasting methods such as ARIMA. Figure 3 shows the auto-correlation function of the time series, where a clear daily periodicity can be observed for both the downlink and uplink case (with the former showing much stronger autocorrelation values than the latter). In order to remove such daily seasonality, we take a simple differencing operation, which is illustrated in Figure 4. In details, first a median daily traffic pattern is created, by taking the median of all 5-minutes or hourly samples for both uplink or downlink traffic series. Then, the median pattern is removed from the original data, resulting in a deseasonalized traffic series. Forecasts have been performed on such latter time series.

B. Training and Test set creation

The application of the aforementioned forecasting methods requires training of the models and estimation of the corresponding parameters. The available one month data is therefore divided in two parts: two weeks of data are used as a training set to estimate parameters for all models. The remaining two weeks are used to evaluate each model performance.

C. Evaluation

Two different metrics are selected for evaluating the performance of the proposed methods:

- **Normalized RMSE (NRMSE):** The Root Mean Squared Error (RMSE) normalized to the standard deviation of the input data, that is:

$$NRMSE = \frac{\sqrt{\frac{1}{T} \sum_{t=1}^T (m_t - \hat{m}_t)^2}}{\sqrt{\frac{1}{T} \sum_{t=1}^T (m_t - \bar{m}_t)^2}}, \quad (7)$$

where T is the number of sample in the test set, m_t is the 5-minutes ground truth value, \hat{m}_t is the forecasted value and \bar{m}_t is the average traffic value computed over the test set. Such a normalization allows to understand how well forecasting is performing, compared to using the average value of the input data as a prediction. NRMSE values lower than 1 indicate that the prediction method is able to produce forecasts whose errors are lower than the data variability. NRMSE values greater than 1 indicate that the forecasts are worse than just using the average value of the input data as predictor. A similar approach is used in [13].

- **Normalized Underestimated RMSE (NUE-RMSE):** One drawback of the NRMSE is that it considers equally traffic overestimation (i.e., $m_t - \hat{m}_t < 0$) and underestimation (i.e., $m_t - \hat{m}_t > 0$). We observe that when the traffic forecasts need to be used as input for C-RAN operations (e.g., in order to reserve resources for the following 5-minute period), traffic underestimation may be much more critical than overestimation from a network operator perspective. In this case in fact, the allocated resources will not be able to manage the future demands, with potentially drastic effects on the network performance.

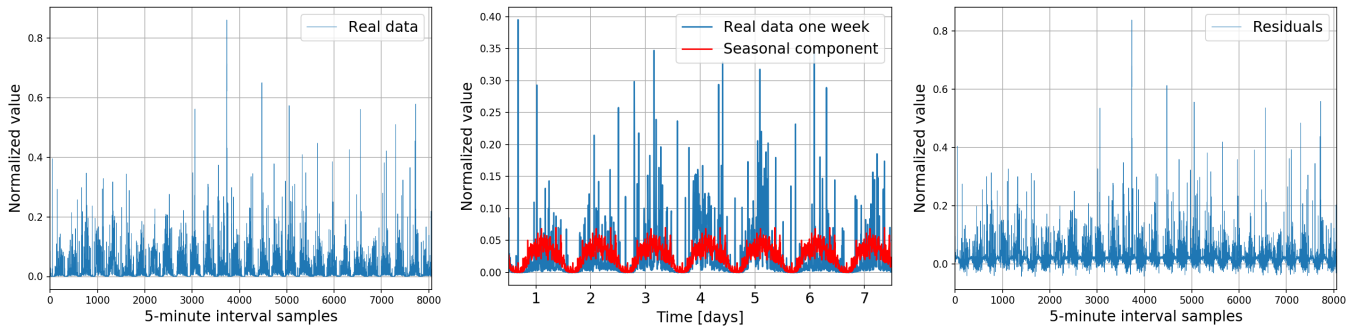


Fig. 4: Decomposition of the downlink data traffic: Full considered data (left), Seasonal component with one week real data (middle), and Residuals (right).

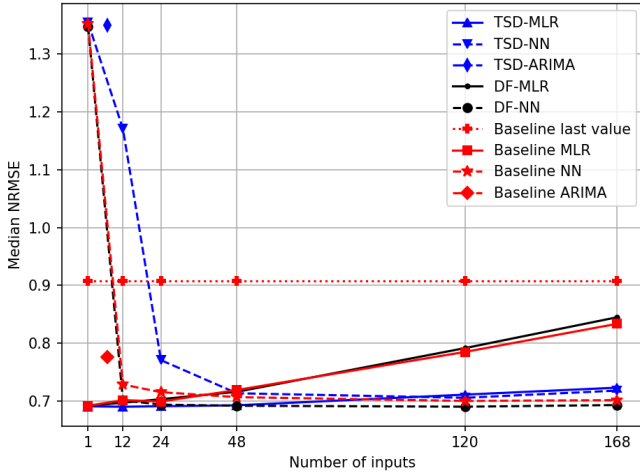


Fig. 5: NRMSE at different number of input samples - deseasonalized Downlink data traffic.

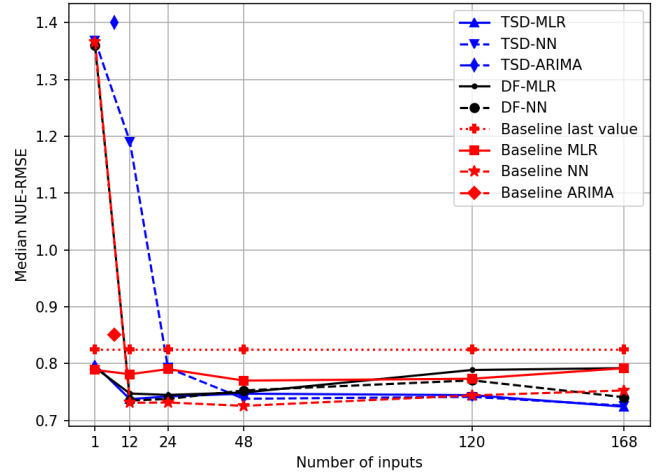


Fig. 6: NUE-RMSE at different number of input samples - deseasonalized Downlink data traffic.

To cope with this, we consider as a second performance metric the Normalized Underestimated RMSE, defined as:

$$NUERMSE = \frac{\sqrt{\frac{1}{T} \sum_{t=1}^T (m_t - \hat{m}_t)^2}}{\sqrt{\frac{1}{T} \sum_{t=1}^T (m_t - \bar{m}_t)^2}} \Big|_{(m_t - \hat{m}_t > 0)}, \quad (8)$$

i.e., the NRMSE restricted to those forecast samples which underestimate traffic.

The experiments have been performed with different number of past samples N : this allows to reason on the amount of samples to be stored in order to perform prediction. For each number of input samples N and each sample in the test data, we predict all possible future 5-minute intervals within each hour and we present median results. To clarify with an example, when $N = 2$ and the current test sample corresponds to the timestamp at 8:00, the hourly samples at 7:00 and 8:00 are used to forecast the values at 8:05, 8:10 up to 8:55, using the models explained before. In case of baselines methods, the samples at 7:55 and 8:00 will be used instead.

D. Downlink traffic forecast

Figure 5 illustrates the NRMSE obtained for the case of downlink traffic. Let's focus first on the baseline methods, that is forecasting methods applied to the 5-minutes sampled time series. The last value approach is the simplest method available and allows to obtain a NRMSE of approximately 0.9. The ARIMA baseline allows to increase the performance at the cost of a slightly higher number of input samples in the model: a grid search for the optimal AR parameters return $p = 7$ as best value, with a corresponding NRMSE of less than 0.8. Baselines using machine learning obtain the best results, with Neural Networks whose performances increase as the number of input samples increases, while Linear Regression models behave in the opposite way (e.g., the performance decrease as the number of input samples increase). This is explained observing that while the Neural Network used in this work has a fixed amount of parameters to be estimated (i.e. one single hidden layer with 32 nodes), the Linear Regression model has a number of parameters that grows with the number of input samples. Therefore, given the limited amount of training data (2 weeks), when increasing N

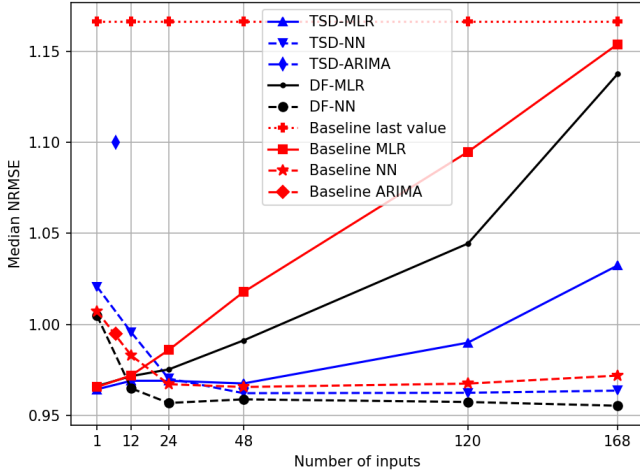


Fig. 7: NRMSE at different number of input samples - deseasonalized Uplink data traffic.

the number of observations available for training Linear Regression models decreases, and this explains the performance degradation. Moving to the results obtained by Time Series Distribution (TSD) and Direct Forecasting (DF) approaches, that is forecasting a high-frequency series starting from a low-frequency one, we can observe how the performance of such methods surprisingly approach the baselines, except when used in conjunction with ARIMA, for which baseline model behaves better. In both approaches, again the Neural Network methodologies increase their performance as the number of input samples increases. For what concerns linear regression models, again we notice how the performance tend to decrease as the number of input samples increases (for the same reason explained before). Interestingly, the approach based on Time Series Distribution allows to limit the performance degradation compared to the one using Direct Forecasting when using linear regression models. Similar comments can be made for the case of NUERMSE, (Figure 6), where the surprisingly good performance of methods such as TSD-MLR or DF-NN is highlighted. It can be seen that, especially for high number of input samples, such methods obtain performances which are even better than baseline methods. That means that it is possible to forecast high-frequency traffic starting from low-frequency time series better than what is possible to do starting from the corresponding high-frequency series.

E. Uplink traffic forecast

We also study the case of uplink traffic. This is particularly interesting for C-RAN operations, as many works proposed a centralized processing of HARQ and FEC schemes, which are characterized by tight time requirements [17], [18]. Being able to forecast accurately uplink traffic is therefore crucial to understand how many resources will be needed in the cloud-RAN for such type of processing. Figure 7 shows the NRMSE performance obtained for the different proposed methods. In general we can observe that the performances

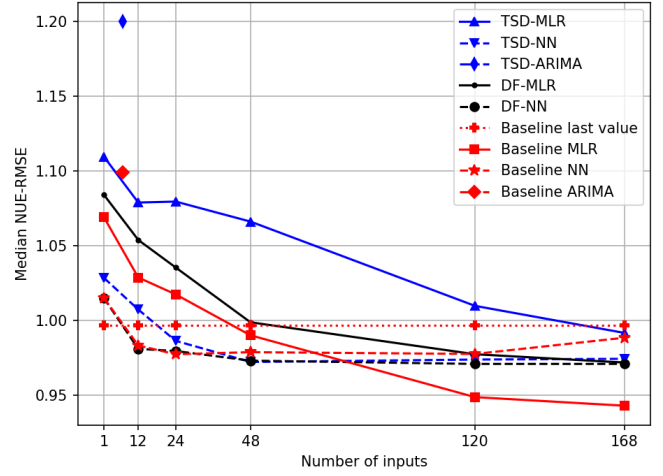


Fig. 8: NUE-RMSE at different number of input samples - deseasonalized Uplink data traffic.

are worse compared to the downlink case: this is due to the high variability of uplink data traces, whose behavior is much more unpredictable than downlink traffic. We can observe that the last value baseline method isn't able to perform accurate forecast (NRMSE greater than 1), while ARIMA and machine-learning based baselines perform just better than the average prediction method. Similarly to the downlink case, we can observe that: i) Neural Networks improve their performances as the number of input samples increases, while this is not true for Linear Regression and ii) surprisingly, TSD and DF methods in combination with Neural Networks outperform the baseline approaches, while TSD-ARIMA is outperformed by ARIMA baseline model. This means that also in the uplink case using a low-frequency traffic time series can be used to forecast higher-frequency traffic samples, even though with performance which are in general worse compared to the downlink scenario.

VI. CONCLUDING REMARKS

In this paper we have tackled the problem of forecasting a high frequency time series of mobile traffic (sampled every 5-minutes) starting from a lower-frequency time series of hourly aggregated traffic samples. We show through experiments that forecasting traffic sampled at different frequencies is possible and, especially for those cases where Neural Networks are used, the results even outperform forecasts obtained starting directly from high-frequency time series. This surprising result can be used as a starting point for several scenarios connected with cognitive networking and C-RAN optimizations (e.g., resource provisioning, storage optimization). Future works will better analyze the performance of other structures of Neural Networks when applied to the forecasting problem. In particular, we plan to add as input to the Neural Networks not only traffic samples but also other sources of data (e.g., number of users connected to the network or number of active sessions). Also, a fine-tuning of the Neural Network hyper-

parameters may be performed, in order to increase the overall performance. Recent Neural Network architectures tailored to time series forecasting such as LSTM or RNN may also be tested.

ACKNOWLEDGMENT

This research has been partially supported by the H2020-MSCA-ITN-2016 framework under grant agreement number 722788 (SPOTLIGHT).

REFERENCES

- [1] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (c-ran): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, 2015.
- [2] A. Jindal, V. Podolskiy, and M. Gerndt, "Multilayered cloud applications autoscaling performance estimation," in *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)*. IEEE, 2017, pp. 24–31.
- [3] X. Zhou, Z. Zhao, R. Li, Y. Zhou, and H. Zhang, "The predictability of cellular networks traffic," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2012, pp. 973–978.
- [4] D. Zhou, Z. Yan, Y. Fu, and Z. Yao, "A survey on network data collection," *Journal of Network and Computer Applications*, vol. 116, pp. 9–23, 2018.
- [5] G. C. Chow, A.-I. Lin *et al.*, *Best linear unbiased interpolation, distribution, and extrapolation of time series by related series*. Princeton University, 1971.
- [6] F. T. Denton, "Adjustment of monthly or quarterly series to annual totals: an approach based on quadratic minimization," *Journal of the American Statistical Association*, vol. 66, no. 333, pp. 99–102, 1971.
- [7] J. Casals, M. Jerez, and S. Sotoca, "Modelling and forecasting time series sampled at different frequencies," *Journal of Forecasting*, vol. 28, no. 4, pp. 316–342, 2009.
- [8] H. Wheeler, V. Isham, D. Cox, R. Chandler, A. Kakou, P. Northrop, L. Oh, C. Onof, and I. Rodriguez-Iturbe, "Spatial-temporal rainfall fields: modelling and statistical aspects," *Hydrology and Earth System Sciences Discussions*, vol. 4, no. 4, pp. 581–601, 2000.
- [9] R. R. Andrawis, A. F. Atiya, and H. El-Shishiny, "Combination of long term and short term forecasts, with application to tourism demand forecasting," *International Journal of Forecasting*, vol. 27, no. 3, pp. 870–886, 2011.
- [10] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1790–1821, 2017.
- [11] S. Ntalampiras and M. Fiore, "Forecasting mobile service demands for anticipatory mec," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2018, pp. 14–19.
- [12] A. Y. Nikraves, S. A. Ajila, C.-H. Lung, and W. Ding, "Mobile network traffic prediction using mlp, mlpwd, and svm," in *2016 IEEE International Congress on Big Data (BigData Congress)*. IEEE, 2016, pp. 402–409.
- [13] M. F. Iqbal, M. Zahid, D. Habib, and L. K. John, "Efficient prediction of network traffic for real-time applications," *Journal of Computer Networks and Communications*, vol. 2019, 2019.
- [14] A. Dalgkisis, M. Louta, and G. T. Karetsos, "Traffic forecasting in cellular networks using the lstm rnn," in *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*. ACM, 2018, pp. 28–33.
- [15] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2018, pp. 231–240.
- [16] A. Okic, A. E. Redondi, I. Galimberti, F. Foglia, L. Venturini, and I. e Bioingegneria, "Analyzing different mobile applications in time and space: a city-wide scenario," 2019.
- [17] N. Nikaein, "Processing radio access network functions in the cloud: Critical issues and modeling," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*. ACM, 2015, pp. 36–43.
- [18] P. Rost and A. Prasad, "Opportunistic hybrid argenabler of centralized-ran over nonideal backhaul," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 481–484, 2014.