L. Lo Schiavo, M. Fiore, M. Gramaglia, A. Banchs and X. Costa-Perez, "Forecasting for Network Management with Joint Statistical Modelling and Machine Learning," 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2022, pp. 60-69

DOI: 10.1109/WoWMoM54355.2022.00028

This is the authors' accepted version of the article. The final version published by IEEE is L. Lo Schiavo, M. Fiore, M. Gramaglia, A. Banchs and X. Costa-Perez, "Forecasting for Network Management with Joint Statistical Modelling and Machine Learning," 23rd IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2022, pp. TBD, doi: TBD.

Forecasting for Network Management with Joint Statistical Modelling and Machine Learning

Leonardo Lo Schiavo[⋆†], Marco Fiore[⋆], Marco Gramaglia[†], Albert Banchs[⋆†] and Xavier Costa-Perez[*]
[⋆]IMDEA Networks Institute, Spain
[†]Universidad Carlos III de Madrid, Spain
[*]i2cat, NEC Laboratories Europe and ICREA, Spain
Email: {leonardo.loschiavo, marco.fiore}@imdea.org, {mgramagl, banchs}@it.uc3m.es, xavier.costa@ieee.org

***Abstract*—Forecasting is a task of ever increasing importance for the operation of mobile networks, where it supports anticipatory decisions by network intelligence and enables emerging zero-touch service and network management models. While current trends in forecasting for anticipatory networking lean towards the systematic adoption of models that are purely based on deep learning approaches, we pave the way for a different strategy to the design of predictors for mobile network environments. Specifically, following recent advances in time series prediction, we consider a hybrid approach that blends statistical modelling and machine learning by means of a joint training process of the two methods. By tailoring this mixed forecasting engine to the specific requirements of network traffic demands, we develop a Thresholded Exponential Smoothing and Recurrent Neural Network (TES-RNN) model. We experiment with TES-RNN in two practical network management use cases, *i.e.*, (i) anticipatory allocation of network resources, and (ii) mobile traffic anomaly prediction. Results obtained with extensive traffic workloads collected in an operational mobile network show that TES-RNN can yield substantial performance gains over current state-of-the-art predictors in both applications considered.**

I. Introduction

The vision for Beyond 5G (B5G) systems sets an extraordinarily high bar for mobile networks, which are expected to become general-purpose platforms providing smart connectivity to a plethora of extremely heterogeneous terminals. As such, B5G shall support diverse classes of services, and do so with outstanding performance: near-zero latency, apparent infinite capacity, and 100% reliability and availability will make the communication infrastructure fully transparent to applications [1]. Meeting this ambitious goal requires further increasing the already substantial complexity of mobile network architectures, enabling the instant orchestration of physical resources and Virtual Network Functions (VNFs) across different network domains, in concertation with time-varying user demands and multi-tenancy requirements.

Network Intelligence for zero-touch management. Handling the escalating complexity of B5G networks with traditional human-in-the-loop approaches will not be possible anymore. Instead, it is expected that current management models will be replaced by zero-touch network and service management technologies, which fully automate the network operation and are presently being standardized [2]. As a result of this transition, the success of B5G will vastly depend on the quality of the Network Intelligence (NI) that will run at schedulers, controllers, and orchestrators across network domains, de-facto managing the zero-touch infrastructure.

Following a popular trend in many research and engineering domains, Artificial Intelligence (AI) models relying on deep neural network (DNN) architectures are regarded as a promising approach for the design of NI solutions. Indeed, AI models have proven remarkably effective at solving complex network operation tasks, and they thrive on the large amount of control and traffic data available within network architectures [3].

Forecasting for anticipatory networking. Many NI solutions abide by anticipatory networking principles, and aim at proactively optimizing network configurations with respect to upcoming traffic conditions rather than to the current state [4]. The prominence of anticipatory NI makes predicting future network states a fundamental task for the effective operation of B5G systems. Forecasting is in fact a manifold problem in networking environments, where different applications require accurate projections of diverse metrics, including computational resources [5], capacity requirements [6], or sheer traffic volumes [7], possibly separately across mobile services [8].

Similarly to what happens for other aspects of NI design, DNN models have lately established as the prevailing approach to develop the predictors that will support proactive decisions by NI solutions. In recent years, a fairly large body of works have explored varied DNN architectures for diverse forecasting objectives, improving the accuracy of legacy statistical models.

Joint statistical modelling and DNN. Owing to the success of DNN models for prediction tasks, current state-of-the-art predictors in the networking domain invariably rely on deep learning, as extensively discussed in Section II. However, very recent results from the machine learning community suggest that hybrid engines integrating statistical modelling and DNN can in fact substantially outperform pure DNN approaches in time series forecasting tasks [9]. The first model of this kind combines a classical Exponential Smoothing (ES) statistical model with a Recurrent Neural Network (RNN) architecture, hence is named ES-RNN [10]. It is a true hybrid predictor, since the parameters of the ES model are optimized concurrently with the RNN weights using a unified gradient descent. Thanks to this joint training, the ES-RNN model represents a leap forward with respect to previous attempts at mixing different statistical and/or machine learning methods: unlike simple combination [11] or ensemble [12] strategies used to date, this technique takes full advantage of the strengths of statistical and machine learning methods, while mitigating their respective limitations.

Contributions. In this paper, we pioneer the adoption in the context of anticipatory NI solutions of a hybrid statistical modelling and machine learning approach Our work yields the following four main contributions.

- We employ, for the first time, a joint ES and RNN training in the context of forecasting for network management.
- We update the operation of the ES-RNN architecture so as to cope with unique features of time series of mobile traffic demands; the result is an original Thresholded ES-RNN (TES-RNN) model, *i.e.*, a general-purpose network traffic forecasting technique that can be tailored to perform predictions for different NI functions.
- We apply the proposed TES-RNN model to two practical anticipatory NI use cases, *i.e.*, (i) anticipatory allocation of network resources, and (ii) prediction of anomalies in the mobile traffic, by training the model with appropriate loss functions for each application.
- We evaluate the performance of TES-RNN against multiple benchmarks that include the original ES-RNN model, a pure RNN architecture, and DNN-based dedicated predictors from the literature, demonstrating the superior performance of our model in both use cases above.

The rest of the paper is structured as follows. In Section II we provide a primer on recent forecasting models in mobile networks. The original ES-RNN engine, and its TES-RNN evolution are presented in Section III. The two target NI use cases are described in Section IV, and the associated comparative evaluation results are discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Forecasting in mobile networks

Standard Developing Organizations (SDOs) are currently delineating the road for the automated management of future-generation mobile network architectures. Dedicated modules are being defined to collect the data and infer the necessary knowledge to support NI functionalities. For instance, 3GPP has already standardized for that purpose a Management Data Analytics Function (MDAF) [13] in the Management and Orchestration (MANO) domain, and a Network Data Analytics Function (NWDAF) [14] in the control plane. Forecasting models implemented in orchestrators and controllers will feed on information made available by functions like MDAF and NWDAF to enable anticipatory zero-touch networking.

Forecasting models for plain traffic volumes. While SDOs are creating the place for prediction tools in NI-managed B5G mobile network architectures, the question of which forecasting techniques shall be implemented within those architectures remains open. There exists a vast literature about prediction in networks [15], where the vast majority of works focus on foretelling future demands expressed in terms of plain traffic volumes. A variety of approaches have been proposed to solve the problem of mobile traffic forecasting, spanning exponential smoothing [16], autoregressive modelling [17]–[19], information theory [20], or Markovian processes [21].

In recent years, deep learning models have rapidly established themselves as the conventional tool for forecasting mobile network traffic. A plethora of diverse DNN architectures have been proposed to date, which leverage, among others, Long Short-Term Memory (LSTM) [7], [8], [22], Stacked AutoEncoder (SAE) [7], and MultiLayer Perceptron (MLP) [23] layers. Convolutional layers have been also extensively tested, in their vanilla [8], [22], three-dimensional [23], or graph [24] versions. These solutions have been used to predict traffic in different settings, including over short [7] and long [23] time horizons, or on aggregates [7], [23] and on a per-application basis [8]. For a comprehensive review, we refer the reader to surveys on applications of deep learning in networking, for forecasting and beyond [25]. We highlight, however, that comparative evaluations carried in the studies above have repeatedly proved that DNN models improve the quality of the prediction with respect to other approaches in general, and to statistical models in particular. This conclusion is also confirmed by dedicated performance analyses juxtaposing heterogeneous models [26], [27].

Forecasting models for network functionalities. Several studies have proposed predictors tailored to specific tasks in anticipatory networking, rather than generic traffic forecasting. These works have targeted forecasting for resource allocation [6], reconfiguration [28] and overbooking [5] in sliced networks, reservation of Physical Resource Blocks (PRB) at the radio access [29], or assignment of baseband processing resources in Cloud Radio Access Network (C-RAN) [30]. Predictors have been also devised for physical layer indicators, such as bandwidth [31], transmission inactivity periods [32], or signal strength [33]. All such efforts are very recent, and exclusively rely on DNN architectures. Among others, they employ MLP [28], 3D convolutional [6], or regular [29] and multivariate [30] LSTM layers.

Positioning of our work. All forecasting models proposed so far in the context of anticipatory networking consider design strategies based on a single approach, most often relying on deep learning. Instead, we introduce the concept of a hybrid prediction for network management that integrates and jointly optimizes statistical modelling and machine learning models. The sizeable performance improvement that this design attains over state-of-the-art DNN architectures in two practical NI use cases indicates that hybrid approaches may set a new standard for forecasting in mobile network environments.

III. TES-RNN hybrid forecasting model

The TES-RNN model proposed in this paper builds upon the innovative design principles first introduced by the very recent ES-RNN engine [10], which is presented in Section III-A. The considered ES-RNN predictor has limitations when confronted to real-world mobile traffic dynamics, as discussed in Section III-B. Our proposed TES-RNN model solves such issues by enhancing the original engine with an automatically learned threshold parameter, as detailed in Section III-C.

A. ES-RNN and joint SGD optimization

ES-RNN is a truly hybrid forecasting model for time series that mixes statistical modelling, *i.e.*, Exponential Smoothing (ES), and machine learning, *i.e.*, Recurrent Neural Networks (RNN). We consider the GPU implementation of ES-RNN [34] as the basis for our study: this variant presents a first pre-processing layer for adaptive and local normalization of input time series by means of ES formulas, followed by a neural network architecture that processes the normalized data and provides forecasts over a customizable time horizon.

The original ES-RNN may adopt a variety of ES expressions, depending on the temporal features of the target data. In networking settings, 24-hour circadian rhythms are known to dominate the fluctuations of mobile data traffic [35], hence we opted for a Holt linear non-seasonal ES formula [36], which is the recommended expression for time series with daily periodicity [10]. At each time step t, the non-seasonal ES updates a normalization coefficient l_t (called level) as

$$l_t = \omega y_t + (1-\omega) l_{t-1}, \tag{1}$$

where $\omega \in [0,1]$ is the exponential smoothing parameter, and y_t represents the value of the input time series at time step t.

The level l_t is used for data normalization. At a given time step t, all values in the input window [t-t_I, t] of size I and in the output interval [t+1, t+t_O] of size O are divided by l_t. During training, the normalized input window is fed to the RNN, whose (normalized) forecast is compared with the normalized output window by means of a loss function. In testing, or when running the model in production systems, de-normalization is performed by multiplying the normalized values forecasted in the prediction horizon O by the level l_t.

The major novelty of the ES-RNN model is that the smoothing parameter ω is a treated as a system variable that is learned together with the weights of the subsequent RNN architecture. In other words, the stochastic gradient descent (SGD) process, normally used to fit the RNN weights, backpropagates in this case before the neural network input layer, and into the preceding ES model, where it updates ω. In this way, a single SGD allows jointly optimizing the parameters of the statistical model and of the neural network, adapting them all to the characteristics of the target time series.

The SGD optimization of ω operated by ES-RNN results in a level l_t that is dynamically adapted to the input data. In turn, this enables a so-called local and adaptive normalization, which (i) ensures that all portions of the time series are equally important to the ensuing neural network training process, and (ii) suitably smooths the machine learning input so that the neural network can concentrate on predicting actual trends, without overfitting on spurious patterns [10]. Thus, this normalization helps forecasting time series with severe fluctuations, like those observed in mobile networks. This is not the case with traditional global normalization of all values to a same $[0,1]$ interval, which does not yield input smoothing, and makes it hard for the RNN to learn to predict small values.

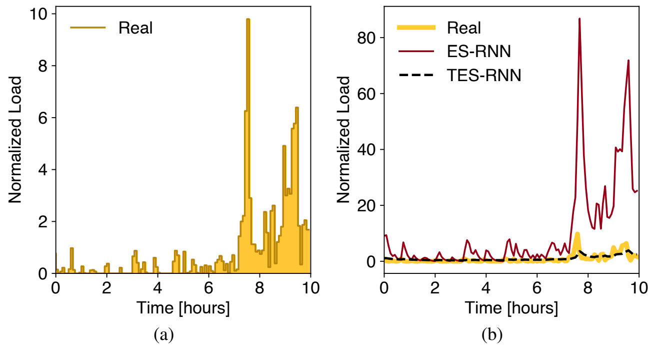


Fig. 1. Example of problematic prediction of real-world mobile traffic by ES-RNN. (a) Instagram demand at one BS. (b) The same demand compared to the prediction generated by ES-RNN trained with a MSE loss function, using an input window of size $I = 6$ and an output window of size $O = 1$.

B. Limitations of ES-RNN with network traffic

The ES-RNN model is intended to operate on time series with strictly positive values of comparable magnitude. However, this assumption is often violated in the mobile networking context, where traffic observed at the radio access and edge network elements is highly irregular and bursty, with continued inactivity periods that lead to a possibly significant presence of zero or near-zero values and severe underutilization of the network. This consideration holds for both voice [37] and data [35] traffic, especially when predictions target demands generated by individual users or at single base stations.

These characteristics of mobile traffic dynamics determine levels l_t computed with (1) that are at times equal to zero, or close to that value. In the case of zero-level values, the ES normalization is simply not possible, as it would involve a division by zero. In the case of values close to zero, value discontinuities between the input and output windows yield normalized outputs that are not numerically comparable with (and in fact much higher than) the values predicted by the neural network; the loss function returns then inflated costs that hinder the quality of the learning process. Figure 1 illustrates the latter problem in a practical scenario. Plot (a) portrays the real-world demand generated by Instagram at one base station during several hours: the inconsistent nature of the traffic, with a long period of very low or no activity, is evident. Plot (b) shows how, when a traffic peak occurs after such a sequence of low-traffic time steps, the network starts predicting amplified values largely above the real traffic demand.

C. TES-RNN and automated level thresholding

To address the shortcomings of the original ES-RNN, we introduce the Thresholded ES-RNN (TES-RNN) model. Our solution employs a threshold τ to bound the minimum value of l_t, which is then updated at each time step t as

$$l_t = \max\{\tau,\ \omega y_t + (1-\omega) l_{t-1}\}. \tag{2}$$

The enhancement in (2) is simple yet effective in solving the issues observed for ES-RNN. A representative example is provided in Figure 1b: TES-RNN does not suffer from inflated predictions and correctly anticipates the growing traffic.

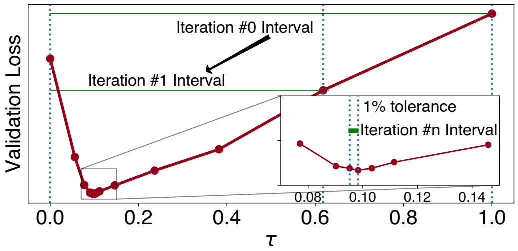


Fig. 2. Example of the sensitivity $F(\tau)$ of the TES-RNN validation loss to the threshold τ (here normalized by the traffic peak), in a practical forecasting task involving Snapchat traffic recorded at a network core datacenter. Dots denote the exploration steps of the Golden-Section search. The inset plot zooms onto the minimum loss, around which the final iterations of the algorithm occur.

In fact, the result in Figure 1b is not obvious to achieve. In particular, the threshold τ is challenging to configure, as it introduces an interesting trade-off. In general, a threshold closer to the traffic peak ensures higher robustness to the problem of time series discontinuities highlighted above. However, it also triggers a global normalization to level τ more often, raising the issue of model insensitivity to low values below the threshold that the local and adaptive normalization aims at solving. Conversely, thresholds closer to the smallest possible level tend to preserve the desirable properties of the fine-tuned ES-RNN normalization, but also incur more often into the issues related to discontinuous data.

There is no one-size-fits-all solution to the trade-off above, and the best value of τ depends on the nature of the traffic time series that is relevant to the target networking functionality. Therefore, also τ needs to be adjusted to the settings of the considered task. Notably, τ is an *hyperparameter* for the TES-RNN model, *i.e.*, an external variable that steers the overall system behaviour. To ensure a smooth operation of the model, it is highly desirable that the setting of τ does not require human intervention, but is fully automated. The setup at hand calls for an *Automated Machine Learning* (or AutoML) approach, since our goal is automatizing the design (in our case, the hyperparameter configuration) of complex neural network models [38].

We design an AutoML approach to learn a correct τ at training time, by leveraging on two considerations about the forecasting model. First, we note that τ is bounded between the value of 0 and the value of the traffic peak. Second, these two extreme values represent the worst case scenarios for the hyperparameter values, for opposite reasons: (i) if $\tau = 0$, the model degenerates into the original ES-RNN with an unbounded normalization, and suffers from the same issues discussed in Section III-B; (ii) when τ is set to the traffic peak, $l_t = \tau, \forall t$ in (2), and the exponential smoothing is simplified to a global normalization that, as explained in Section III-A, does not perform well with high-variance mobile traffic time series. These properties make it very likely that the function $F(\tau)$ that describes the behavior of the TES-RNN loss versus the threshold hyperparameter is convex: a representative example with real-world traffic is portrayed in the main plot of Figure 2.

The convexity of the problem makes a Golden-Section search algorithm [39] a suitable way to identify the τ minimizing the validation loss $F(\tau)$: the values $\tau = 0$ and $\tau = 1$ are used as the initial search interval boundaries, one of which is then updated at each iteration so as to narrow the interval around the minimum. As an illustrative example, we depict in Figure 2 the evolution of our proposed AutoML approach with a target tolerance of 1%. In this case, the loss is minimum when the normalized threshold τ is 0.1 (*i.e.*, around 10% of the traffic peak), and our approach correctly identifies such a value within 11 iterations.

The final structure of the TES-RNN model, including the threshold τ and its AutoML configuration, is depicted in Figure 3. For the predictor part, the TES-RNN model uses the same Dilated Recurrent Network (DRNN) employed by ES-RNN. The DRNN architecture is composed of Dilated LSTM-based layers [40], which, unlike vanilla LTSM, realize a RNN attention mechanism. Indeed, dilation allows a given LSTM layer to use as input the hidden state of layers associated to more than one time step in the past, and allows dynamically shifting importance from a particular single state to a group of past states. In our implementation, we employ a computationally efficient DRNN variant [34], and let the Dilated LSTM stack be followed by a non-linear layer. A final linear adapter performs size adaptation between the state of the last layer and the overall output layer. The AutoML component implements a Golden-Section search algorithm to find the best operational τ. At each step, the validation loss $F(\tau)$ is computed for two threshold values inside the search interval: the threshold yielding the higher validation loss is used to update either the left or the right extreme of the interval. The algorithm iterates until the length of the search interval falls below a target tolerance: then, the final τ is the mean of the last interval.

IV. Forecasting use cases

As discussed in Section I, forecasting future network states is a cornerstone task for many networking problems, including admission control [41], resource allocation [6], handovers [30] and power management [32], among others, which involve different operation time scales [28]. The TES-RNN architecture described in Section III is general-purpose and agnostic to the specific networking application: it can be trained to support different NI instances, *e.g.*, by combining it with a suitable loss function that allows optimizing the model for a given prediction task. Next, we present two practical anticipatory networking use cases where TES-RNN can be used as the forecasting model, which also set the ground for the experimental evaluation conducted in Section V.

A. *Use case I: capacity allocation for network slicing*

A first use case of interest for forecasting in anticipatory networking is that of capacity allocation, *i.e.*, reserving the resources needed to meet the upcoming demand for a given service. This functionality is especially relevant in network slicing settings, where (sets of) services run in different slices, and the operator needs to dedicate sufficient resources to each slice, in agreement with the load generated by the corresponding service(s) [41].

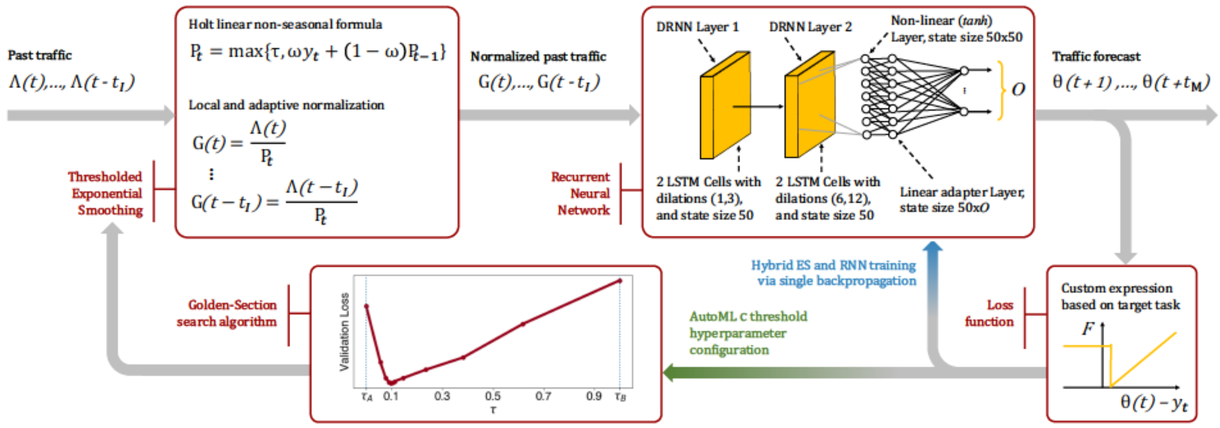


Fig. 3. TES-RNN architecture. The traffic Λ observed over the past I time steps is input to the TES component for a local and adaptive normalization of non-negligible traffic above τ. The resulting traffic λ is fed to the RNN component, which outputs a forecast θ of traffic within a horizon of O time steps. During training, the loss computed from θ is used to learn the threshold hyperparameter τ of the TES block in a AutoML style via a Golden-Section search.

The anticipatory NI in charge of capacity allocation to slices must rely on so-called capacity forecasting, *i.e.*, predicting the minimum capacity sufficient to accommodate the future slice traffic. We highlight that capacity forecasting is a fairly unique problem, where sheer accuracy is not the most relevant metric. Instead, it is critical that the prediction stays above the actual load with a very high probability, because underestimation determines the allocation of insufficient capacity to slices, hence service disruption on the user side. Underprovisioning also triggers violations of the Service Level Agreement (SLA) between the slice tenant and the network operator, which thus incurs into substantial economic penalties. Clearly, this must be avoided without allocating exceedingly large amounts of unnecessary resources, which also have a cost for the operator.

The problem of capacity forecasting has recently received attention, with the proposal of dedicated predictors [6], [28]. These models rely on a loss function that drives the learning process so as to capture the actual cost of incurring SLA violations against that of overprovisioning the slice capacity. Specifically, the function handles negative and positive errors differently, so as to reflect the different costs they entail in the context of virtualized communication networks, as follows.

- A constant penalty β is associated to each negative error, which causes a SLA violation during the predicted time interval. β can be customized to the desired behavior: for instance, higher values may be used when reliability is paramount (*e.g.*, for slices serving ultra-reliable low-latency communications, or URLLC), and lower penalties can be applied for slices with more relaxed requirements.
- A monotonically increasing cost is attributed to positive errors, which imply the allocation of excess resources. Therefore, the cost is proportional to the amount of (unnecessarily) provisioned capacity. Typically, the expenditure is assumed to grow linearly with the overprovisioned capacity, with a fixed rate γ of cost per surplus capacity.

The configuration of the two costs can be in fact controlled by a single parameter $\alpha = \beta/\gamma$, which represents the amount of overprovisioned capacity that the operator is willing to deploy to avoid committing an SLA violation. Formally, for a given prediction error x, the loss function that abides by the specifications above is expressed as

$$L(x) = \begin{cases} \alpha - \epsilon \cdot x & \text{if } x \leq 0 \\ \alpha - \frac{1}{\epsilon}x & \text{if } 0 < x \leq \epsilon\alpha \\ x - \epsilon\alpha & \text{if } x > \epsilon\alpha, \end{cases} \tag{3}$$

where steep slopes (implemented with a small positive ϵ) ensure differentiability over the whole x domain [6].

The parameter α serves as a knob to steer the operational point of the system towards higher expenses in deployed resource but reduced chances of SLA violations, or vice-versa. As a result, the loss function in (3) can be parametrized to the specifications of different network infrastructure locations (*e.g.*, reflecting the higher cost of deploying resources at the network edge than at the core), resource types (*e.g.*, capturing the fact that radio resources are sensibly more expensive than CPU resources), and SLA strategies (*e.g.*, expressing the higher fees for violations affecting slices of critical services).

B. Use case II: anomaly detection in mobile service traffic

The second use case we study is an anticipatory anomaly detection framework, where the NI must trigger an alarm when an abnormal future traffic load is expected for a specific mobile service. The anomaly detection problem is summarized in Figure 4a. The predictor module is in charge of producing a *probability distribution* of the traffic demand that the target service will generate in the next time slot. Such a probabilistic prediction is compared against a reference interval that encompasses the expected range of *normal* traffic values in the following time slot. Then, if the probability of the anticipated traffic to be outside the reference interval is beyond a threshold, an alarm is raised. This allows the associated NI to perform some preventive actions, such as those detailed in the 3GPP TS 23.288 [14] technical specification under the "Abnormal UE behavior" analytics, which capture anomalies such as unexpected large rate flows generated by terminals.

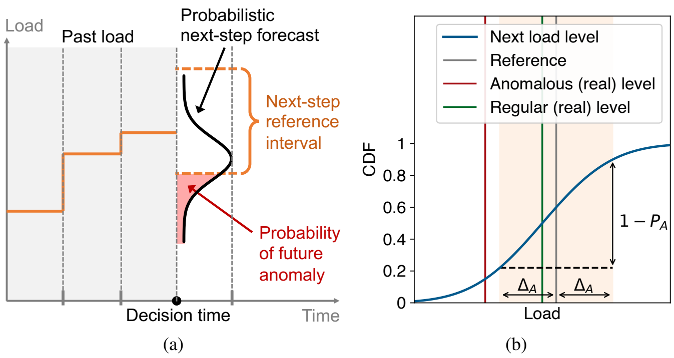


Fig. 4. Anomaly forecasting use case. (a) Representation of the anomaly forecasting problem. (b) Explanation of the operation of anomaly detection.

We consider a simple yet practical implementation[1] of the approach above that is commonly adopted in many fields also outside networking [42]. First, it is worth noting that the output of the forecasting algorithm shall not be a scalar but a probability distribution of the future traffic load. This type of output is implicit in certain types of models like Bayesian Neural Networks, which are however computationally expensive and not suited for resource-constrained network environments. In order to generate a probabilistic forecast with a generic neural network, we resort to recent findings in uncertainty modelling [43]: specifically, by activating dropout layers in the predictor during the inference phase and performing a Monte Carlo testing, the neural network returns a set of values that has been shown to closely approximate the probabilistic result of a deep Gaussian process implemented with a Bayesian network.

The anomaly detection algorithm then operates on the empirical PDF f_p of the predicted traffic values for each decision interval, as illustrated in Figure 4b. First, the upper and lower limits that mark the boundaries between regular and anomalous values are computed as $x_{l,h} = R \pm \Delta_A$, where R is a configurable reference value. From these two values, the probability of a future anomaly is empirically calculated as $P_A = 1 - (F_p(x_h) - F_p(x_l))$, where $F_p(x) = \sum_{k<x} f_p(k)$ is the CDF of the anticipated traffic. Finally, an alarm is triggered if $P_A > \tau_A$. The parameters R, Δ_A and τ_A control the sensitivity of the algorithm. In our experiments, we set the reference values for the estimated load in the next prediction step R as the average of the last three load values, $\Delta_A = 0.9 \cdot R$, and $\tau_A = 0.9$. In other words, we trigger an alert when the model forecasts a future traffic with a 90% probability to fall outside a range $\pm 90\%$ of the reference value.

The correctness of the anticipatory anomaly detection can be determined by checking whether the actual traffic falls into the $x_{l,h}$ interval or not, and computing precision and recall scores. Clearly, a higher accuracy in the probabilistic traffic forecast, denoted by a lower variance around a value closer to the true one, yields better performance: a Mean Square Error (MSE) loss function is thus a sensible choice for this use case.

[1]Our goal is not to propose a novel anomaly detection algorithm, but to compare the effectiveness of different forecasting models in supporting such a task. Therefore, we are not interested in developing a complex algorithm for anomaly detection, and using a baseline solution is sufficient for our purpose.

V. Performance Evaluation

We assess the performance of TES-RNN in the two use cases set out in Section IV, hinging on real-world mobile traffic measurement data collected in an operational network. Specifically, we consider mobile data traffic time series recorded at more than 400 4G/LTE base stations that provide coverage to millions of subscribers in a metropolitan area.[2] The data was collected in the production infrastructure of a major operator during 8 continuous weeks, by passive probes tapping at interfaces of the relevant Gateway GPRS Support Node (GGSN) and Packet Data Network Gateway (P-GW).

The measurement probes leverage Deep Packet Inspection (DPI) to extract protocol information from packets in the GPRS Tunneling Protocol user plane (GTP-U). Such information is then fed to proprietary classifiers developed by the operator in order to determine the service associated to each session. As a result, the time series we use in our evaluation describe the traffic generated by individual popular services.

All time series have a set temporal granularity of 5 minutes. This temporal granularity is compatible with the requirements of the two target use cases, since: (i) the reconfiguration periodicity of slice resources allowed by modern Virtual Infrastructure Managers (VIM) is in the order of minutes [44], and (ii) anticipating anomalies by several minutes is largely sufficient to plan and enact countermeasures. Therefore, a prediction of the traffic in the next 5-minute time step (*i.e.*, a point forecast of the time series) is aligned with both use cases, and we consider an output window size $O = 1$ in all our experiments. Also, and unless stated otherwise, we use an input window size of $I = 6$ time steps to feed the model (corresponding to a past history of 30 minutes), and we employ 8, 2 and 1 different weeks of traffic for training, validation and testing, respectively.

As a final remark, we highlight that our study observes high privacy and ethical standards: (i) the network operator conducted the data collection abiding by applicable regulations at national and international levels; (ii) the competent national privacy agency and the data protection officer of the operator authorized the data processing; and, (iii) the time series we accessed for the purpose of this work solely describe traffic aggregated at individual base stations over large sets of users, and do not contain personal subscriber information.

A. Forecasting for capacity allocation

We set the capacity allocation use case presented in Section IV-A in a network core Cloud scenario, where a datacenter runs VNFs for the traffic generated in the whole target region by four mobile applications, *i.e.*, Facebook, Instagram, Snapchat and YouTube. Each such service is assigned a dedicated network slice, and the NI responsible for capacity allocation at the datacenter must reserve in advance enough resources to accommodate the future demand of single slices.

[2]Due to confidentiality reasons, we cannot disclose the identity of the operator, the target geographical region, or the absolute volumes of traffic captured in the data. We thus either normalize the traffic values, or report them without the scaling factor that would reveal their order of magnitude.

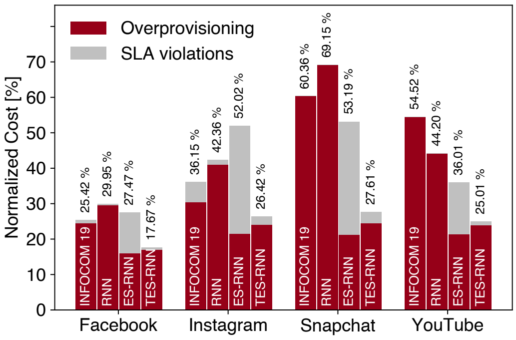


Fig. 5. Additional capacity allocation cost caused by INFOCOM19, RNN, ES-RNN, and TES-RNN prediction errors. Results refer to four slices assigned to specific services at a network core datacenter, with parameter $\alpha = 3$.

To address this problem, we train TES-RNN with the appropriate loss function in (3) and compare our hybrid solution against the following three relevant benchmarks:

- INFOCOM19 [6] is the predictor designed by the study that first introduced the problem of capacity forecasting and proposed the loss function in (3). It relies on a DNN architecture fed with a 3D tensor of the spatiotemporal mobile data traffic, and uses convolutional layers to capture geographical correlations in the demands. This is the state-of-the-art forecasting model for capacity allocation.
- ES-RNN [10] is the GPU implementation of the original ES-RNN approach presented in Section III-A. For the sake of fairness, ES-RNN is trained with the loss in (3).
- RNN uses the same RNN architecture of ES-RNN, but relies on a global normalization for the input data, thus without any of the optimizations proposed in this work and in [10]. This benchmark is useful to understand how statistical modelling favors the prediction accuracy. We train also this benchmark with the loss function in (3).

1) Overall capacity forecasting performance: We start by comparing the total costs incurred by the operator when supporting capacity allocation with the different forecasting models, in Figure 5. In order to make these values interpretable, all costs are normalized to the (unavoidable) cost of the minimum resources needed to accommodate the exact demand for each service. In other words, costs are expressed as the percent excess over a baseline given by an oracle that makes a perfect prediction. In each case, the figure also tells apart the fraction of the cost resulting from the two sources of penalty, *i.e.*, resource overprovisioning and SLA violations.

The key observation is that TES-RNN consistently outperforms the benchmarks, with gains over the second best solution that range between 8% and 25%. Moreover, our solution steadily guarantees very low SLA violation probabilities, which is a clearly desirable feature for the operator. And, it does so by causing an overprovisioning that is lower than or comparable to that produced by the other predictors. These are very encouraging findings, as one of the benchmarks is the state-of-the-art model designed for capacity forecasting.

Interestingly, ES-RNN yields an allocation of unnecessary resources close to that of TES-RNN, but incurs into much more frequent SLA violations. INFOCOM19 and RNN, when compared to TES-RNN, induce a substantial higher overprovisioning that often helps limiting SLA violations.

2) In-depth analysis of one prediction instance: To gain additional understanding on the behaviors of the forecasting models presented above, we detail a representative case of capacity prediction in Figure 6. The plots show the time series of the real traffic in the Facebook slice, as well as the corresponding capacity allocation foreseen by each predictor.

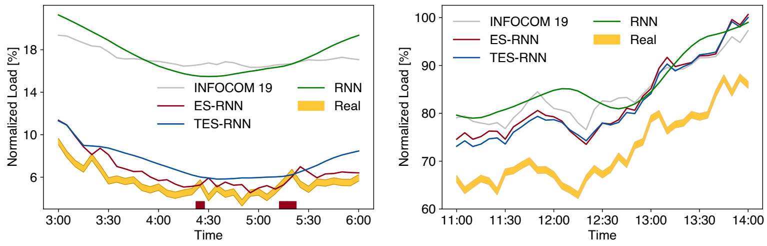
Plot (a) portrays the traffic dynamics over a full week, and underscores how all models follow well the long-timescale fluctuations of the demands, such as low overnight traffic or different activity peaks during daylight. Plots (b) and (c) present a close-in view of two specific 3-hour periods, which are evidenced by vertical shades in plot (a). The zoom magnifies how TES-RNN and ES-RNN help dimensioning a capacity that is closer to the real demand than that anticipated by INFOCOM19 and RNN, especially in low traffic conditions.

Plot (b) also exemplifies the reason for the poor performance of ES-RNN in terms of high SLA violations: when used in combination with the loss function in (3), the model has issues in anticipating small variances in the traffic fluctuations, which causes the capacity forecast to come too close to the future demand. The result is a frequent underprovisioning: for instance, ES-RNN assigns insufficient resources to the Facebook slice in multiple periods in the considered example, highlighted by the red intervals on the abscissa in the figure. Instead, TES-RNN forecasts a smoother capacity curve that stays above minor fluctuations, and hence yields a resource provisioning similar to ES-RNN but avoids SLA violations.

3) Control of SLA violations: The results presented before are for one specific value of the parameter α that controls the equilibrium of overprovisioning and SLA violation risk in the loss function in (3). By varying the parameter, the operator shall be able to steer the capacity forecast so as to favor one source of cost over the other, as explained in Section IV-A.

Figure 7 illustrates the capability of each model to enforce the desired control above. The plot shows, for the case of the Facebook slice, the normalized cost determined by each predictor, as α sweeps values one order of magnitude apart. We observe that TES-RNN yields again the best performance in all settings. More importantly, it keeps the overall cost low by progressively decreasing the occurrence of SLA violations as α grows, which is exactly the desired behavior. INFOCOM19 and RNN can also achieve this result, however at a cost in terms of overprovisioning that is almost twice that of our hybrid model. ES-RNN is instead unable to modulate the SLA violation cost, which in fact surprisingly grows with α.

The reason for the counter-intuitive ES-RNN performance can be explained by the breakdown of the two cost sources, in Figure 8. The plot illustrates, for each case in Figure 7: (i) on the ordinate, the overprovisioning, still expressed as the added cost over that of the optimal oracle; and, (ii) on the abscissa, the SLA violations, measured as the percentage of 5-minute time steps during which the allocated resources are insufficient to serve the slice demand. The trends are consistent



(a) (b) (c)

Fig. 6. Time series of the real traffic generated by the Facebook slice, and of the relative capacity predictions of INFOCOM19, RNN, ES-RNN and TES-RNN. (a) Weekly time series, with highlighted time intervals for close-in analysis. (b) Zoomed view of the 3:00-6:00 interval of Tuesday, with SLA violation periods of ES-RNN marked in red on the abscissa. (c) Zoomed view of the 11:00-14:00 interval of Wednesday. Figure best viewed in colors.

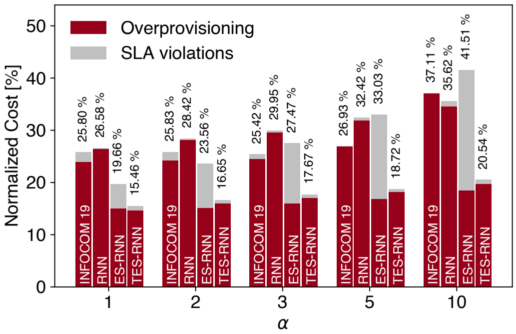


Fig. 7. Additional capacity allocation cost of INFOCOM19, RNN, ES-RNN, and TES-RNN prediction errors, versus α and for the Facebook slice.

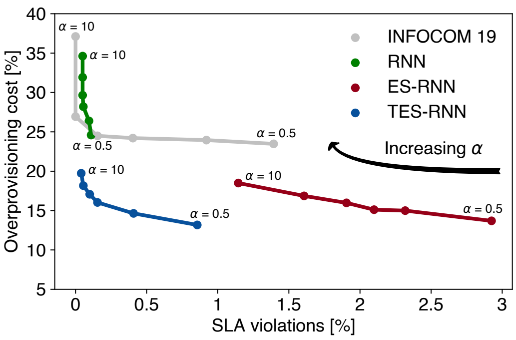


Fig. 8. Limits in terms of SLA violations and overprovisioning costs that can be attained by TES-RNN, and the chosen benchmarks for the Facebook slice.

across all models, and higher values of α always entail less violations, as one would expect. However, while TES-RNN, INFOCOM19 and RNN can rapidly bring underprovisioning cases down to zero when α surges, the ES-RNN model is much less sensitive to the parameter. Specifically, this predictor reduces SLA violations at a slower pace than the rate at which α increases: by looking at the extreme cases in the plot, ES-RNN lowers violations by just one third when α grows 20-fold. As α represents the cost of one SLA violation, the cut in the number of occurrences is insufficient to compensate for the higher penalty of each infraction, which explains the growing trend of the SLA violation cost under ES-RNN in Figure 7.

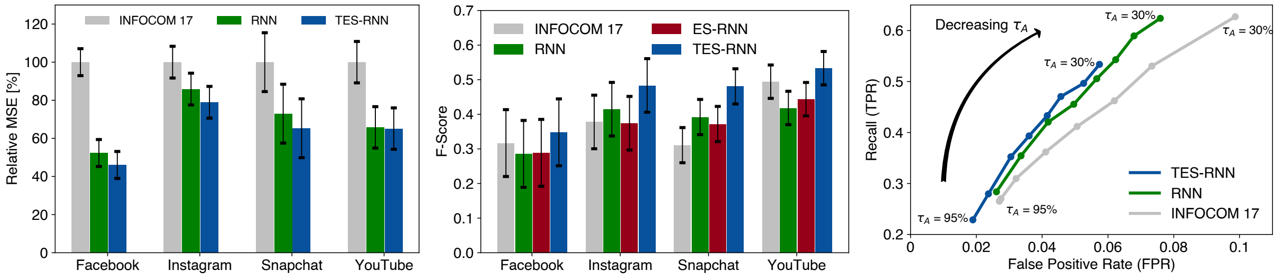
More generally, Figure 8 gives a clear view of the operating points of each forecasting method. TES-RNN offers by far the best options to the operator, as it allows choosing among configurations that simultaneously provide less SLA violations and lower overprovisioning costs than the benchmarks. ES-RNN allows staying at low overprovisioning levels as well, but SLA violation rates cannot be controlled even with very aggressive α settings, as discussed before. In contrast, both INFOCOM19 and RNN can limit SLA violation rates, but without a clear (and relatively high) bound on the minimum overprovisioning costs achievable. As a result, TES-RNN brings the best of the other models: for any possible operating point of INFOCOM19, ES-RNN or RNN, we can choose an α for TES-RNN that improves cost on both dimensions.

B. *Forecasting for anomaly detection*

The second use case we consider for the comparative evaluation of TES-RNN is that of anomalous load detection at base stations introduced in Section IV-B. We set this use case in a virtualized network environment running an end-to-end network slicing model, where proactive load anomaly detection is paramount for the timely identification of undesired situations which could be amended by, *e.g.*, new network configurations. In such settings, the anomalous load detection NI operates at the granularity of individual services. Specifically, we run experiments for slices that each accommodate one of four different services, *i.e.*, Facebook, Instagram, Snapchat and YouTube. For each slice, we consider different base stations and assess the performance of the anomaly detection algorithm discussed in Section IV-B that relies on forecasting models of the slice traffic at each such base station.

To support the anomaly detection decision, TES-RNN is trained with an MSE loss function, according to the discussion in Section IV-B. With such a loss function, TES-RNN operates as a traditional mobile traffic forecasting model; this steers our choice of benchmark to the following models.

- INFOCOM17 [7] is a popular forecasting technique that is explicitly designed to predict mobile network traffic at the level of individual base stations. It leverages a DNN architecture where both global and local SAE layers are used to learn spatial features in the data, followed by LSTM layers that capture temporal correlations. This benchmark represents the state of the art in point forecast at base station level, *i.e.*, the problem at hand; while other, more recent predictors of mobile traffic volume have been proposed in the literature, they target different objectives, such as forecasting over very long time horizons [23], or forecasting for the radio access as a whole [24].
- ES-RNN and RNN, as discussed in Section V-A. In this case, the models are trained with an MSE loss function.



(a) (b) (c)

Fig. 9. Benchmarking between INFOCOM17, RNN and TES-RNN forecast. (a) MSE values, (b) F1 Scores obtained with the algorithm discussed in Section IV-B, and (c) ROC Curve for the Facebook service, for different τ_A thresholds. We remark that subplots (a) and (c) do not show results for ES-RNN because, as extensively explained in Section III-C, training such a model with a MSE loss function in presence of mobile data traffic yields exceedingly high overestimation (see, *e.g.*, Figure 1) and a very high number of false positives. This results in the poor ES-RNN performance in subplot (b).

1) Mobile traffic prediction accuracy: We start our assessment by comparing the sheer accuracy of the TES-RNN against the benchmarks in the task of point forecasting mobile traffic at base stations. Figure 9a shows the results, for all models, services, averaged over five different base stations. Independently of the test configuration, TES-RNN yields a more accurate prediction than INFOCOM17. The average MSE reduction across all base stations is in the 15%-30% range, peaking at 50% for the Facebook case.

2) Anomaly detection performance: Having observed the superior accuracy of TES-RNN mobile traffic prediction, we investigate how this reflects on the actual performance of the anomaly detection. We emulate the situation in which the network analytics function of a mobile network gathers data from the UPF [14] to monitor, *e.g.*, the excessive usage of the network by a terminal, and has to generate an alarm if such event is anticipated to happen in the near future. We consider for our test a scenario where different base stations are monitored with the algorithm introduced in Section IV-B.

Figure 9b shows the performance of all the benchmarks in terms of F1 Score, averaged over the five selected base stations, for the four selected application types. TES-RNN always achieves the best performance, while competitors provide unbalanced results, highly depending on the considered application. In particular, the baseline ES-RNN almost always triggers the anomalous load alarm, as the predicted values are often well above the thresholds (and the real values).

To further corroborate the quality of TES-RNN in this kind of tasks, we also evaluate its effectiveness with variable τ_A values. More specifically, we show in Figure 9c the ROC curve for the selected benchmarks. TES-RNN always yields the best pairing between the Recall and the false positive rate, for all the considered τ_A in the range between 30% and 95%.

C. Complexity

We conclude by analyzing the complexity of the proposed solutions, in terms of the average training time for a single prediction. The results are reported in Table I and refer to experiments on a NVIDIA Tesla T4 GPU. In the first use case, TES-RNN required on average 16.93 minutes per capacity allocation experiment (combining training and testing), whereas ES-RNN needed 16.32 minutes. INFOCOM19 was the most taxing model, with a mean of 24.64 minutes per run. For the second use case, TES-RNN required on average 6.8 minutes for a single forecasting experiment, while INFOCOM17 needed on average 51.61 minutes to process the spatial dependency data needed by the model. Overall, TES-RNN can be trained as fast as less optimized models (*i.e.*, ES-RNN) and much faster than state-of-the-art solutions, and is only slower than models with lower performance (*i.e.*, RNN).

TABLE I
TRAINING TIME OF ALL CONSIDERED BENCHMARKS, IN MINUTES.

Use Case	TES-RNN	ES-RNN	RNN	INFOCOM19/17
1	16.93	16.32	4.56	24.64
2	6.8	6.5	2.09	51.61

VI. CONCLUSIONS

We proposed TES-RNN, a first-of-its-kind hybrid forecasting model for mobile networks. By enhancing a recently proposed method for joint optimization of statistical models and neural network architectures, TES-RNN offers unprecedented performance in supporting different anticipatory networking tasks. When confronted with two practical application use cases characterized by different real-world traffic volumes and dynamics, TES-RNN granted gains up to 25% over state-of-the-art predictors that were specifically designed for the target problem. As a result, we believe that TES-RNN paves the way for a new and improved generation of forecasting models that will contribute to the success of fully automated, zero-touch mobile network architectures.

The code of the TES-RNN model is publicly accessible at https://github.com/nds-group/wowmom22_tes-rnn.

ACKNOWLEDGEMENTS

This work is partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement no.101017109 DAEMON. This work is partially supported by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D 6G-CLARION-OR and AEON-ZERO. The authors would like to thank Dario Bega for his contribution to developing the forecasting use case I, and Slawek Smyl for his feedback on the baseline ES-RNN model.

REFERENCES

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6g networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[2] European Telecommunications Standards Institute (ETSI), "ZSM Scenarios and key requirements." ETSI ISG ZSM 001, Oct. 2018.

[3] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[4] N. Bui *et al.*, "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.

[5] J. X. Salvat, L. Zanzi, A. Garcia-Saavedra, V. Sciancalepore, and X. Costa-Pérez, "Overbooking Network Slices Through Yield-driven End-to-end Orchestration," in *Proc. of the 14th International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT)*, (Heraklion, Greece), pp. 353–365, Dec. 2018.

[6] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," in *Proc. of IEEE INFOCOM*, (Paris, France), pp. 280–288, Apr. 2019.

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

[8] C. Zhang, M. Fiore, and P. Patras, "Multi-Service Mobile Traffic Forecasting via Convolutional Long Short-Term Memories," in *Proc. of IEEE International Symposium on Measurements and Networking (IEEE M&N)*, (Catania, Italy), pp. 1–6, Jun. 2019.

[9] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54 – 74, 2020.

[10] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, no. 1, pp. 75 – 85, 2020.

[11] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, 1989.

[12] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[13] 3GPP TS 28.533 v16, "Management and Orchestration of Networks and Network Slicing; Management and Orchestration Architecture (Rel. 16)," June 2019.

[14] 3GPP TS 23.288 v16, "Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services (Rel. 16)," June 2019.

[15] M. Joshi and T. H. Hadi, "A review of network traffic analysis and prediction techniques," 2015.

[16] D. Tikunov and Toshikazu Nishimura, "Traffic prediction for mobile network using holt-winter's exponential smoothing," in *2007 15th International Conference on Software, Telecommunications and Computer Networks*, pp. 1–5, 2007.

[17] F. Xu *et al.*, "Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach," *IEEE Transactions on Services Computing*, vol. 9, pp. 796–805, Sep. 2016.

[18] 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)*, pp. 14–19, 2018.

[19] M. Zhang, H. Fu, Y. Li, and S. Chen, "Understanding urban dynamics from massive mobile traffic data," *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 266–278, 2019.

[20] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, "The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 234–240, 2014.

[21] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '11, (New York, NY, USA), p. 305–316, Association for Computing Machinery, 2011.

[22] C. Huang, C. Chiang, and Q. Li, "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.

[23] 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*, Mobihoc '18, p. 231–240, Association for Computing Machinery, 2018.

[24] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, 2019.

[25] B. Ma, W. Guo, and J. Zhang, "A survey of online data-driven proactive 5g network optimisation using machine learning," *IEEE Access*, vol. 8, pp. 35606–35637, 2020.

[26] J. Yin, W. Rao, M. Yuan, J. Zeng, K. Zhao, C. Zhang, J. Li, and Q. Zhao, "Experimental study of multivariate time series forecasting models," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, (New York, NY, USA), p. 2833–2839, Association for Computing Machinery, 2019.

[27] S. Sone, J. Lehtomäki, and Z. Khan, "Wireless traffic usage forecasting using real enterprise network data: Analysis and methods," 06 2020.

[28] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Aztec: Anticipatory capacity allocation for zero-touch network slicing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pp. 794–803, 2020.

[29] C. Gutterman, E. Grinshpun, S. Sharma, and G. Zussman, "Ran resource usage prediction for a 5g slice broker," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '19, (New York, NY, USA), p. 231–240, Association for Computing Machinery, 2019.

[30] L. Chen, T. Nguyen, D. Yang, M. Nogueira, C. Wang, and D. Zhang, "Data-Driven C-RAN Optimization Exploiting Traffic and Mobility Dynamics of Mobile Users," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[31] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, "LinkForecast: Cellular link bandwidth prediction in LTE networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1582–1594, 2018.

[32] P. Brand, J. Falk, J. Ah Sue, J. Brendel, R. Hasholzner, and J. Teich, "Adaptive predictive power management for mobile LTE devices," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2020.

[33] A. Kulkarni, A. Seetharam, A. Ramesh, and J. D. Herath, "DeepChannel: Wireless channel quality prediction using deep learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 443–456, 2020.

[34] A. Redd, K. Khin, and A. Marini, "Fast ES-RNN: A GPU Implementation of the ES-RNN Algorithm," *arXiv e-prints*, p. arXiv:1907.03329, Jul 2019.

[35] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and Z. Smoreda, "Identifying Common Periodicities in Mobile Service Demands with Spectral Analysis," in *IEEE MedComNet*, (Arona, Italy), June 2020.

[36] R. Hyndman, A. Koehler, J. Ord, and R. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008.

[37] B. Cici, E. Alimpertis, A. Ihler, and A. Markopoulou, "Cell-to-cell activity prediction for smart cities," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 903–908, 2016.

[38] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[39] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American mathematical society*, vol. 4, no. 3, pp. 502–506, 1953.

[40] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. J. Witbrock, M. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," *CoRR*, vol. abs/1710.02224, 2017.

[41] V. Sciancalepore *et al.*, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *IEEE INFOCOM 2017*, May 2017.

[42] P. J. Rousseeuw and M. Hubert, "Anomaly detection by robust statistics," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 2, 2018.

[43] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. of The 33rd International Conference on Machine Learning (ICML)*, (New York, NY, USA), pp. 1050–1059, Jun. 2016.

[44] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 518–532, Sep. 2016.