This is the authors' accepted version of the article. The final version published by IEEE is L. Lo Schiavo, G. García, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Perez, "The TES Framework: Joint Statistical Modeling and Machine Learning for Network KPI Forecasting," in IEEE Transactions on Network and Service Management, vol. 23, pp. 350-364, 2026, doi: 10.1109/TNSM.2025.3628788.

The TES framework: Joint Statistical Modeling and Machine Learning for Network KPI Forecasting

Leonardo Lo Schiavo, Genoveva Garcia, Marco Gramaglia, Marco Fiore, Albert Banchs, Xavier Costa-Perez

***Abstract*—The vision of intelligent networks capable of automatically configuring crucial parameters for tasks such as resource provisioning, anomaly detection or load balancing largely hinges upon efficient AI-based algorithms. Time series forecasting is a fundamental building block for network-oriented AI and current trends lean towards the systematic adoption of models based on deep learning approaches. In this paper, we pave the way for a different strategy for the design of predictors for mobile network environments, and we propose the Thresholded Exponential Smoothing (TES) framework, a hybrid Statistical Modeling and Deep Learning tool that allows for improving the performance of network Key Performance Indicator (KPI) forecasting. We adapt our framework to two state-of-the-art deep learning tools for time series forecasting, based on Recurrent Neural Networks and Transformer architectures. We experiment with TES by showcasing its superior support for three practical network management use cases, *i.e.*, (i) anticipatory allocation of network resources, (ii) mobile traffic anomaly prediction, and (iii) mobile traffic load balancing. Our results, derived from traffic measurements collected in operational mobile networks, demonstrate that the TES framework can yield substantial performance gains over current state-of-the-art predictors in the applications considered.**

***Index Terms*—Forecasting, prediction, mobile traffic, network KPI, network management, neural networks, statistical modeling.**

I. Introduction

EVERY new generation of mobile networks invariably raises the bar for the performance, reliability, and security of cellular communication systems. Adhering to such a trend, 6G systems are expected to support diverse classes of services and do so with near-zero latency, apparent infinite capacity, and 100% availability, making *de-facto* the communication infrastructure fully transparent to applications [1] and turning 6G networks into general-purpose platforms providing smart connectivity to a plethora of very heterogeneous terminals.

While today's mobile communication infrastructures are already extremely tangled architectures that entail significant challenges in terms of equipment management, traffic engineering, and capacity allocation [2], 6G systems will introduce several layers of substantial additional complexity [3], [4]. Indeed, meeting the ambitious 6G performance targets will require instant orchestration of physical resources and Virtual Network Functions (VNFs) across different network domains, in concert with user demands and multi-tenancy requirements that rapidly shift in time.

Machine Learning (ML) and Artificial Intelligence (AI) are largely regarded as fundamental enablers to realize such a vision. Integrating AI/ML solutions, supported by a native network architecture [5], will pave the road towards the efficient support of various use cases that dramatically enhance the performance of next-generation systems. Data-driven models have been repeatedly shown to offer enhanced quality for key network management tasks such as anomaly detection [6], traffic classification [7], resource orchestration [8], radio access operation [9], and energy saving [10], just to name a few.

In many of those tasks, anticipatory decision-making is a very desirable –if not mandatory– feature, making prediction an essential building block to AI/ML-driven network management [11]. In this context, a plethora of works have proposed ever more accurate forecasting models [12], [13] and recent works have also shown how predictors can be tailored to the downstream network management task by steering their output [8], [14].

In this work, we focus on forecasting network Key Performance Indicators (KPIs), such as traffic demands or user throughput, as one of the cornerstones of future zero-touch network management [15]. While traffic prediction was carried out via statistical models until the first decade of the century [16], [17], AI/ML solutions have nowadays taken a clear lead and dominate the literature [18]–[21]. We depart from the common practice to propose pure AI/ML-based models and explore hybrid approaches where traditional statistical modeling is combined with deep learning [22]. This hybrid strategy is known to yield resilience to noisy data and wide excursions in time series of financial data or weather fluctuations [23], and we show that it can also help achieve higher prediction accuracy in the presence of real-world mobile traffic data, thus benefiting network management tasks that build upon such forecasts.

By pioneering the adoption of hybrid predictors using statistical modeling and ML in the context of anticipatory network management, our work yields the following main contributions.

- We introduce for the first time a hybrid model combining exponential smoothing (ES) with different deep learning models based on Recurrent Neural Network (RNN) and Transformer architectures, and demonstrate how it improves quality of the downstream anticipatory network management tasks, with improvements in the 4%-26%, 20%-40% and 6%-16% range depending on the scenario.

L. Lo Schiavo, G. Garcia, M. Gramaglia, and A. Banchs are with University Carlos III de Madrid. Corresponding email: lloschia@pa.uc3m.es, genoveva@pa.uc3m.es, mgramagl@it.uc3m.es, banchs@it.uc3m.es.

M. Fiore and A. Banchs are with IMDEA Networks Institute. Corresponding email: marco.fiore@imdea.org, albert.banchs@imdea.org.

X. Costa-Perez is with NEC Laboratories Europe GmbH, I2CAT, and ICREA. Corresponding e-mail: xavier.costa@neclab.eu.

- We update the operation of the state-of-the-art ES-RNN architecture to cope with unique features of mobile traffic dynamics; 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 network management functions.
- We apply the same methodology to the Transformer neural network architecture, understanding the benefits and trade-offs of the different approaches.
- We apply the proposed models to three practical zero-touch management use cases, *i.e.*, (i) capacity allocation, (ii) anomaly detection, and (iii) load balancing, for which we train the models with appropriate loss functions.
- We evaluate the performance of our solutions against recent works in the literature, demonstrating in all use cases above its superior performance with respect to state-of-the-art Deep Neural Network (DNN) architectures.

This provides a substantial update with respect to our previous work in [24], as we discuss the usage of Transformer models in this context (to our knowledge, the first attempt to integrate statistical modeling with this architecture), improve the threshold selection of the TES framework through an automatic and generalizable Reinforcement Learning (RL) algorithm, and add the load balancing use case to further showcase the adaptability of our solution.

The paper is structured as follows: we detail the context of the zero-touch network management and related work in time series forecasting for networks in Section II. We discuss the application of hybrid strategies that combine ML and statistical modeling for forecasting in Section III and present our proposed TES framework in Section IV. Finally, we analyze a set of relevant use cases in Section V and their performance evaluation in Section VI, before concluding in Section VII.

II. Related Work

Relying on a precise time-series forecasting algorithm is a fundamental building block for many autonomous network management and operation solutions [25]. Indeed, the quality of the prediction plays a role in the overall performance of the autonomous network management algorithm: with a more precise forecast, the decision taken can guarantee a better outcome. In the following, we revise the state-of-the-art solutions for autonomous and zero-touch network management, with a focus on anticipatory networking. Finally, we explore the works in the field of joint statistical modeling and ML, which is the solution we adopt in this paper to improve the forecasting quality of pure DNN models.

Network Intelligence for zero-touch management. Handling the escalating complexity of Beyond 5G (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 [26]. 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, AI models relying on 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 [27].

Forecasting for anticipatory networking. Many NI solutions build upon anticipatory networking principles and aim at proactively optimizing network configurations with respect to upcoming traffic conditions rather than to the current state [28]. 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 future projections of diverse metrics, including computational resources [29], capacity requirements [14], or sheer traffic volumes [12], possibly separated by mobile service [13].

Similarly to what happens for other aspects of NI design, DNN models have lately been established as the prevailing approach to developing the predictors that will support proactive decisions by NI solutions. In the past few years, a fairly large body of works has explored varied DNN architectures, which target diverse forecasting objectives, and are typically proven to yield improved accuracy over legacy statistical models.

Joint statistical modeling and DNN. While current state-of-the-art predictors in the networking domain invariably rely on deep learning, very recent results from the ML community suggest that hybrid engines that integrate statistical modeling and DNN can, in fact, substantially outperform pure DNN approaches in time series forecasting tasks. The very first model of this kind comfortably won the renowned M4 Competition, a challenge for data scientists to develop ever more accurate time series predictors [23]. It did so by beating a variety of statistical and ML benchmarks, as well as 48 competitor solutions, across 100,000 experiments.

The aforementioned engine combines a classical ES statistical model with a RNN architecture, hence it is named ES-RNN [22]. It is a true hybrid predictor since the parameters of the ES model are optimized concurrently with the RNN weights using 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 ML methods: unlike simple combination [30] or ensemble [31] strategies used to date, this technique takes full advantage of the strengths of statistical and ML methods, while mitigating their respective limitations.

III. Hybrid network KPI prediction with machine learning and statistical modeling

The hybrid prediction approach proposed in this paper builds upon the innovative design principles first introduced by the recent ES-RNN engine [22], which is presented in Section III-A. The considered ES-RNN predictor has limitations

when confronted with real-world mobile traffic dynamics, as discussed in Section III-B. Our proposed hybrid methodology enhances the structure proposed in [22] and solves such issues by enhancing the original engine with an automatically learned threshold parameter, as detailed in Section IV-B.

A. ES-RNN and joint SGD optimization

ES-RNN is a truly hybrid forecasting model for time series that mixes statistical modeling, *i.e.*, ES, and ML, *i.e.*, RNN. We consider the GPU implementation of ES-RNN [32] as the basis for our study: this variant presents a first pre-processing layer for adaptive and local normalization of input time series using 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 [33], hence we opted for a Holt linear non-seasonal ES formula [34], which is the recommended expression for time series with daily periodicity [22]. 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 using 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 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 for jointly optimizing the parameters of the statistical model and 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 ML input so that the neural network can concentrate on predicting actual trends, without overfitting on spurious patterns [22]. Thus, this normalization helps forecast time series with severe fluctuations, like those observed in mobile networks. This is not the case with traditional global normalization of all values to the same $[0,1]$ interval, which does not yield input smoothing and makes it hard for the RNN to learn to predict small values.

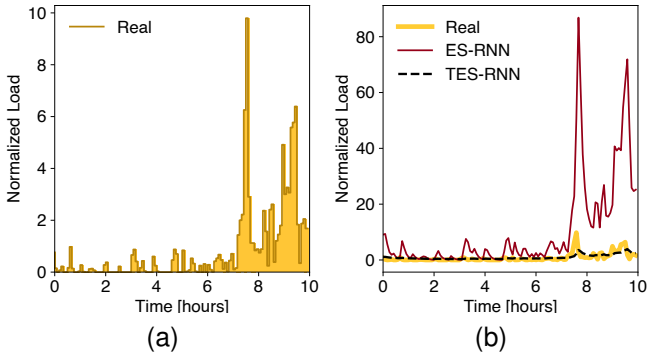

Fig. 1: Example of problematic prediction of real-world mobile traffic by ES-RNN. (a) Instagram demand at one BS. (b) The same demand is compared to the prediction generated by ES-RNN trained with a Mean Squared Error (MSE) loss function, using an input window of size $I = 6$ and an output window of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

B. Limitations of hybrid predictors with network KPIs

The ES-RNN model is intended to operate on a time series with strictly positive values of comparable magnitude. However, this assumption is often violated in the mobile networking context, where KPIs observed at the radio access and edge network elements are 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 [35] and data [33] 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, 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 for 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.

IV. The TES framework

Motivated by the analysis performed in Section III, we propose a hybrid methodology for the forecasting of time series for network management purposes. The methodology is composed of (i) a module that introduces the statistical

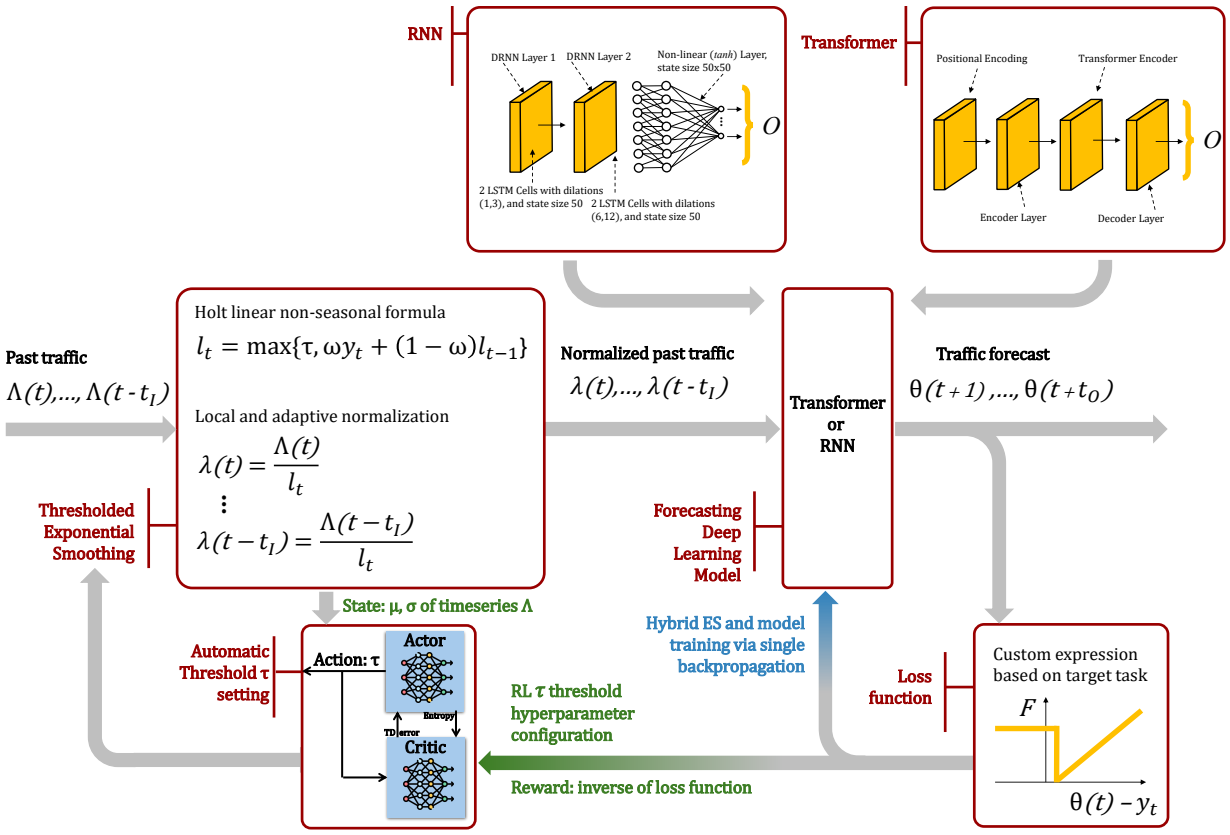

Fig. 2: The architecture. The traffic Λ observed over the past I time steps is input to the TES component for a local and adaptive normalization of non-negligible traffic above τ. The resulting traffic λ is fed to the model of choice, which outputs a forecast θ of traffic within a horizon of O time steps. During training, the loss computed from θ is used to learn the threshold hyperparameter τ of the TES block in an AutoML style via a Reinforcement Learning algorithm.

modeling part through ES, and (ii) a dynamic thresholding (T) module that counters the problem discussed in Section III-B, building hence our TES framework. The overall framework is depicted in Figure 2.

A. Neural network predictors

One of the goals of our work is to demonstrate the wide applicability of the TES approach, independently of the model used for the actual forecasting of the time series. We consider hence two models for time forecasting: the Recurrent Neural Network already used in [36] and a Transformer architecture [37], whose capabilities in time series forecasting have been recently studied.

1) RNN: RNNs are a class of neural networks designed for processing sequential data, making them particularly well-suited for time series forecasting. Unlike traditional feedforward neural networks, RNNs possess an internal state that allows them to retain information from previous inputs, enabling the modeling of temporal dependencies. This ability to maintain a memory of past observations makes RNNs effective in capturing patterns and trends over time, a very good feature for accurate time series prediction.

2) Transformer: Initially introduced for natural language processing tasks, the transformer architecture [38], [39] has been used for time series forecasting [40]–[42] due to its ability to handle long-range dependencies and parallelize computation. Unlike RNNs, Transformers do not rely on sequential data processing; instead, they employ self-attention mechanisms to weigh the importance of different time steps, capturing complex patterns and relationships within the data. This architecture consists of encoder and decoder layers, which enable the model to focus on relevant parts of the input sequence dynamically. The parallel processing capability significantly accelerates training and inference times, making Transformers suitable for large-scale time series datasets.

We corroborated this fact in a set of training experiments that we executed for the experimental evaluation detailed in Section VI, where we compare the time elapsed for training and the retained loss on the same training data, for the RNN and Transformer architectures. Pure transformer-based solutions could be trained generally one order of magnitude faster (less than 1s vs. tens of seconds) than RNN solutions. The advantages in training time that the Transformer architecture has are beneficial for the automatic thresholding feature that we propose in this work. See Table IV for more details.

B. Improved statistical modeling

As introduced in Section III-B, deep learning models suffer from noisy data with zeros. For these reasons, we introduce

TABLE I: Training time over 20 epochs and normalized MSE of forecasting models predicting Facebook time series.

Model	Training time (min)	Normalized MSE
RNN	2.85	0.023
Transformer	0.17	0.018
Autoformer	2.05	0.008
Informer	2.83	0.007
N-Beats	1.61	0.126
D-Linear	1.07	0.140

a thresholded version of the two models discussed below, following the framework in Figure 2.

1) TES-RNN: 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.

2) ES-Transformer and TES-Transformer: To improve the performance of the original Transformer model, we adopt equation 1 to introduce the ES-Transformer model with an unbounded adaptive normalization of the inputs. However, similar shortcomings observed for ES-RNN are also observed for ES-Transformer. Therefore, we introduce a Thresholded ES-Transformer (TES-Transformer) model to address those limitations. TES-Transformer adopts a conditional two-stage normalization scheme: in the first stage, a normalization coefficient l_t is computed as in equation 1 to normalize the values in the input window [t-t_I, t]. Then, a threshold τ is used to scale the maximum value of the normalized input window to get a second-stage normalization coefficient l_t^{max} as

$$l_t^{max} = \tau \cdot \max_{[t-t_I,t]} \lambda(t). \tag{3}$$

In the second conditional stage, if l_t^{max} is bigger than a guard value δ, then l_t^{max} is used to further normalize the input window to avoid the training artifacts of the original Transformer and ES-Transformer models.

C. Effect of ES and TES normalization

The effect of the normalization discussed in III-A, attained by applying equations 1, 2 and/or 3 depending on the model, can be observed in Figure 3, which shows training inputs over two consecutive days. While the global normalization scales all the input values to the same [0, 1] interval, the other normalizations yield better input smoothing and ease the prediction task for small values, i.e., low overnight traffic values. The benefits of the latter normalization on the forecasting performance will be discussed in detail in Section VI.

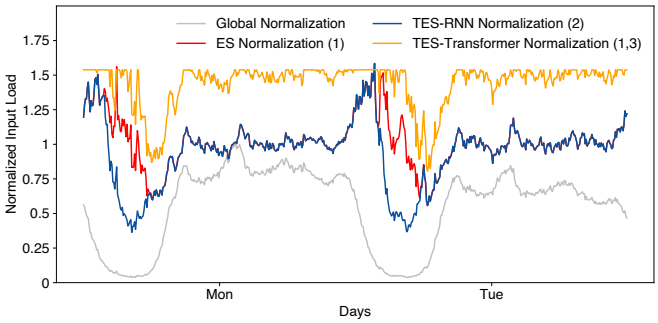

Fig. 3: Input values with different kinds of normalization.

TABLE II: Symmetric Mean Absolute Percentage Error (SMAPE) between stationary and non-stationary training.

Service	Stationary training	Non-Stationary training
Facebook	37.44%	36.42%
Instagram	67.27%	66.89%
Snapchat	69.73%	69.99%

D. Comparative analysis and impact of stationarity

To select the best internal design for the TES framework, we conducted a preliminary comparison of recent forecasting models based on Transformer and linear architectures, including Informer [37], Autoformer [43], pure Transformer [38], N-BEATS (Neural Basis Expansion Analysis for Time Series) [44], and D-Linear [45]. As shown in Table I, although Informer and Autoformer yielded a slightly lower normalized MSE, the Transformer offered a much shorter training time, a critical parameter given the complexity introduced by the TES framework for, e.g., finding the best τ. Importantly, TES compensates for the modest accuracy gap of the pure Transformer, improving the final performance as will be shown later in Section VI.

Another important aspect we take into account while designing the internal model of the TES framework is the impact of the stationarity of the input time series, which has been discussed in the literature [46]–[48] as an important metric for assessing the complexity of forecasting problems. Following these works in the literature, we evaluated the stationarity of the traffic time series using Augmented Dickey-Fuller (ADF) tests and statistical moments (mean, variance, skewness, and kurtosis). We found two different patterns: daily data showed non-stationarity, with varying statistics and ADF p-values above 0.05, while weekly aggregation revealed stationary behavior, as illustrated in Figure 4. To assess the robustness of our model, we trained the pure Transformer model over temporally ordered weekly data (stationary) and shuffled individual days (non-stationary). The performance analysis, summarized in Table II, yields comparable accuracy in both setups, showing that these models generalize well across time shifts even for the non-stationary case, showing how the model copes well with complex problems. The results of Table I and Table II are obtained using the time series of popular services that will be introduced in Section VI, where we train our models in larger stationary scenarios.

The results we presented in this section show that the TES

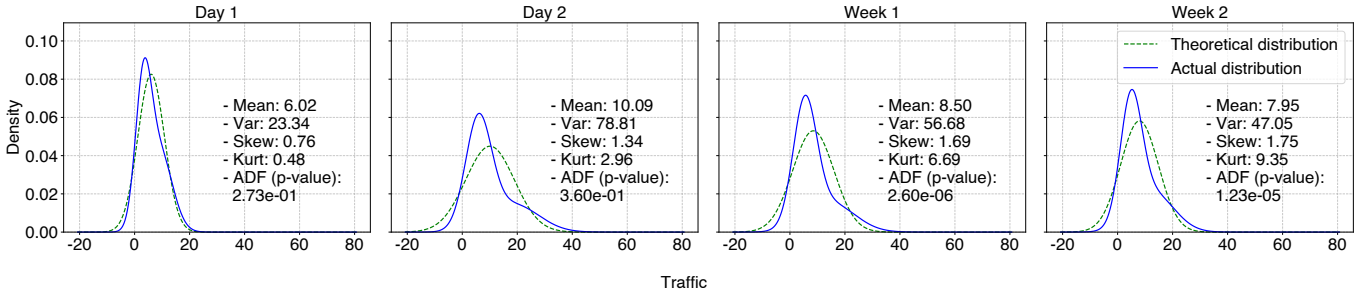

Fig. 4: Distribution of Instagram time series over 2 individual days (leftmost plots) and 2 consecutive weeks (rightmost plots).

framework is general in nature, without any noticeable bias on the kind of input data (e.g., stationary or not), and can be applied to forecasting models beyond Transformers.

E. Parametrization of the hyperparameter τ

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. Generally, 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 normalization but incur more often the issues related to discontinuous data. These problems, which are independent of the actual deep learning model used for forecasting, require an automatic algorithm for the correct setting of the τ.

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, τ also needs to be adjusted to the settings of the considered task. Notably, τ is a *hyperparameter* for the TES models, as it steers the overall system behavior. To ensure a smooth operation, it is highly desirable that the setting of τ does not require human intervention, but is fully automated and generalizable. The setup at hand calls for an *Automated Machine Learning* (or AutoML) approach, since our goal is to automate the design of complex neural network models [49].

For this task, while in an earlier version of the framework [36] we used a Golden-Section search algorithm based on convex loss functions [50], we now propose a more generalizable approach based on a RL algorithm that automatically selects the best τ value. We resort to a *soft actor-critic* deep RL algorithm to maximize an arbitrary reward function while exploring as randomly as possible the space of possible τ (action) values at training time through an entropy component. The critic neural network estimates the effect of selecting a given τ value for a state s, which captures the nature of the traffic time series and is represented by its mean μ and standard deviation σ. Such an effect is estimated using an instantaneous reward function, which is the additive inverse of the prediction loss obtained by forecasting the time series with state s using the selected τ. Leveraging the estimates

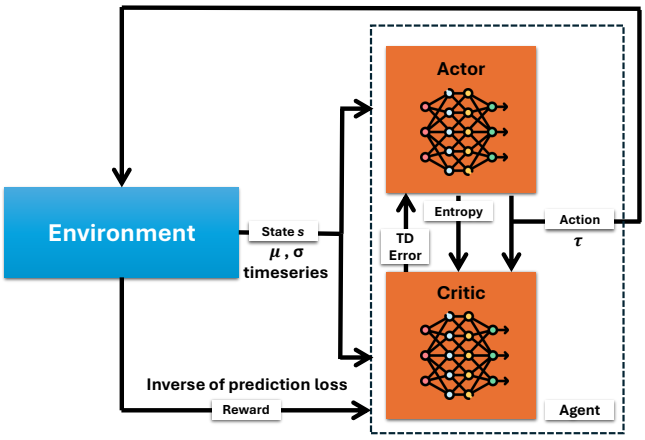

Fig. 5: Architecture of the *soft actor-critic* algorithm.

of the critic for a given state s, the actor outputs the best τ from a discrete action space with 20 possible values in the range [0.05, 1] with step 0.05. The architecture of the *soft actor-critic* algorithm is depicted in Figure 5.

V. Application Use Cases

As discussed in Section I, forecasting future network KPIs is a cornerstone task for many networking problems, including admission control [51], capacity allocation [14], handovers [52] and power management [53], among others, which involve different operation time scales [8]. The TES architecture described in Section IV-B 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 the TES framework can be used as the forecasting model, which also sets the ground for the experimental evaluation conducted in Section VI.

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) [51].

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, the prediction must stay 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 substantial economic penalties. Clearly, this must be avoided without allocating exceedingly large amounts of unnecessary resources, which also has a cost for the operator.

The problem of capacity forecasting has recently received attention, with the proposal of dedicated predictors [8], [14]. These models rely on a loss function that drives the learning process 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, to reflect the different costs they entail in the context of virtualized communication networks, as follows.

- A constant penalty β is associated with each negative error, which causes an 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

$$L(x) = \begin{cases} \alpha - \epsilon \cdot x & \text{if } x \leq 0 \\ \alpha - \frac{1}{\epsilon} x & \text{if } 0 < x \leq \epsilon\alpha \\ x - \epsilon\alpha & \text{if } x > \epsilon\alpha, \end{cases} \tag{4}$$

where steep slopes (implemented with a small positive ϵ) ensure differentiability over the whole x domain [14].

The parameter α serves as a knob to steer the operational point of the system towards higher expenses in deployed resources but reduced chances of SLA violations, or vice-versa. As a result, the loss function in (4) 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).

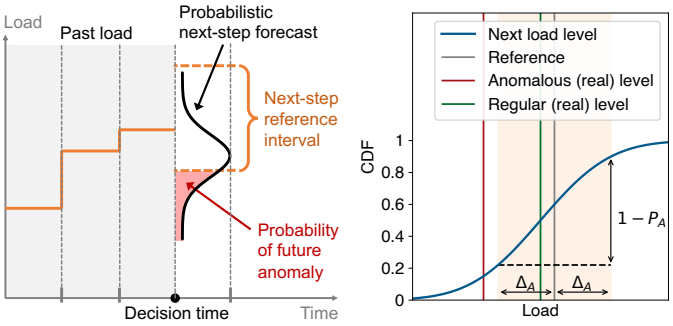

Fig. 6: Anomaly forecasting use case. (left) Representation of the anomaly forecasting problem. (right) Explanation of the operation of anomaly detection.

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 6a. 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 being 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 [54] technical specification under the "Abnormal behavior" analytics, which capture anomalies such as unexpected large rate flows generated by terminals.

We consider a simple yet practical implementation[1] of the approach above that is commonly adopted in many fields, also outside networking [55]. 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 modeling [56]: specifically, by activating dropout layers in the predictor during the inference phase and performing a Monte Carlo test, the neural network returns a set of values that have been shown to closely approximate the probabilistic result of a deep Gaussian process implemented with a Bayesian network.

[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.

The anomaly detection algorithm then operates on the empirical Probability Density Function (PDF) f_p of the predicted traffic values for each decision interval, as illustrated in Figure 6b. 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 Cumulative Distribution Function (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 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 MSE loss function is thus a sensible choice for this use case.

C. Use case III: anticipatory load balancing

Load balancing aims at equally splitting the load among different entities and is another networking functionality for which forecasting is critical. Indeed, anticipatory load balancing can be applied at different levels, including the following

- *Load balancing at edge clouds.* In edge cloud facilities, *e.g.*, for Cloud Radio Access Network (C-RAN), it is desirable to associate base stations with data centers in such a way that the future traffic load channeled to each data center is balanced, and no data center will suffer congestion and reduced performance.
- *Load balancing for network slicing.* Under slicing models, network entities must run dedicated, customized VNFs to serve the traffic associated with each slice. Operators then have to map slices to network nodes, ensuring that the upcoming VNF load is evenly shared across the latter, to optimize the global system performance.
- *Load balancing at base stations.* In the presence of increasingly dense network deployments, users are offered an increased choice of candidate base stations for association. Forecasting allows operators to make informed decisions on new user associations and equalize the charge across base stations to ensure service continuity.

Independent of the problem variant, anticipatory load balancing requires a prediction of future traffic that is as accurate as possible. In this case, whether the prediction falls above or below the actual load is irrelevant: any deviation of the prediction from the actual demand causes an imbalance in the resulting load that only depends on the error magnitude, hence a negative error is not more harmful than a positive one.

For the purpose of evaluation, we focus on the third problem above, *i.e.*, load balancing at base stations. This type of task is run in modern networks, for instance, by the Policy Control Function (PCF) through the User Equipment (UE) Route Selection Policy (RSP) [57]. The PCF assigns an incoming UE to a Protocol Data Unit (PDU) session or network slice, once the UE is activated. The availability of an accurate prediction of future traffic at each base station allows the NI deployed at the PCF to drive the assignment in a way that the ensuing load is leveled across the radio access infrastructure.

As mentioned above, this type of load balancing requires a traditional mobile traffic forecasting model to operate in an anticipatory fashion. We thus rely on a conventional loss function that weights equally negative and positive errors, *i.e.*, the MSE. Based on the traffic load predicted with this loss function, a load balancer performs the corresponding mapping to equalize the (expected) load at the different entities. In the case of load balancing at base stations, forecasting is naturally performed on traffic loads at the base station level; then, the load balancing NI engine running at the PCF manages each association request by assigning the soliciting UE to the base station with the lowest forecasted load.

VI. Performance evaluation

We assess the performance of the proposed TES framework in the two use cases set out in Section V, 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 11 continuous weeks, by passive probes tapping at interfaces of the Gateway GPRS Support Node (GGSN) and Packet Data Network Gateway (PDNG) in configurable intervals ranging from 5 to 15 minutes.

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 to determine the service associated with each session. As a result, the time series we use in our evaluation describes the traffic generated by individual popular services.

All time series have the finest temporal granularity allowed by our dataset, which is 5 minutes. This temporal granularity is compatible with the requirements of the three target use cases, since (i) the reconfiguration periodicity of slice resources allowed by modern Virtual Infrastructure Managers (VIM) is in the order of minutes [58], and (ii) anticipating anomalies or (iii) balancing load 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 the use cases, and we consider an output window size $O = 1$ in all our experiments. Also, we use an input window size of $I = 6$ time steps to feed the model

[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.

TABLE III: Summary of experimental settings

Parameter	Value
Temporal granularity	5 minutes
Input window size (I)	6 time steps (30 minutes)
Output window size (O)	1 time step (5 minutes)
Data split (weeks)	Training: 8; Validation: 2; Test: 1

(corresponding to a history of 30 minutes), and we employ 8, 2, and 1 different weeks of traffic for training, validation, and testing, respectively. The guard value for TES-Transformer second-stage normalization is $\delta = 0.001$. All the experimental settings are summarized in Table III: training, validation, and test sets include both on-peak and off-peak behavior, following a night-day, weekday-weekend pattern [59].

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 V-A in a network core Cloud scenario, where a data center runs VNFs for the traffic generated in the whole target region by three traffic-intensive mobile applications, *i.e.*, Facebook, Instagram, and Snapchat. Each such service is assigned a dedicated network slice, and the NI responsible for capacity allocation at the data center must reserve in advance enough resources to accommodate the future demand of single slices.

To address this problem, we train the models used in TES with the appropriate loss function in (4) and compare our hybrid solution against the following four relevant benchmarks:

- INFOCOM19 [14] is the predictor designed by the study that first introduced the problem of capacity forecasting and proposed the loss function in (4). 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 [22] 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 (4).
- 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 [22]. This benchmark is useful for understanding how statistical modeling favors prediction accuracy. We also train this benchmark with the loss function in (4).
- Transformer uses the model first introduced in [60], which is also the basis of our ES-Transformer and TES-Transformer models. For this baseline model, we use the loss function in (4) as well.

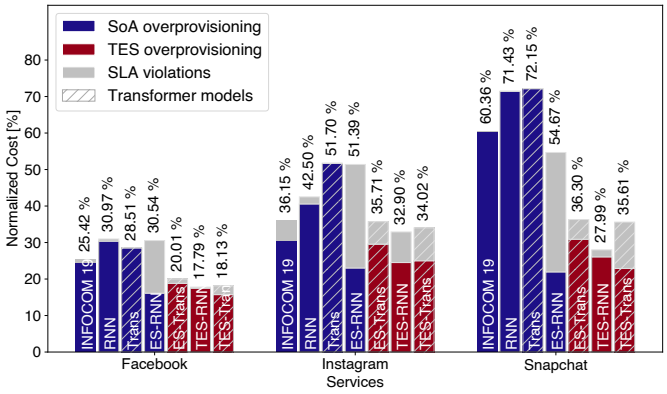

Fig. 7: Additional capacity allocation cost caused by state-of-the-art models (INFOCOM19, RNN, Transformer, and ES-RNN, in blue), and the proposed in this paper (ES-Transformer and TES models, in red) prediction errors. Results refer to three slices assigned to specific services at a network core data center, with parameter $\alpha = 3$.

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 7. 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. We group results into state-of-the-art approaches (blue bars in Figure 7) and our three proposals: ES-Transformer, TES-RNN, and TES-Transformer.

The key observation is that our approaches consistently outperform the benchmarks, with gains over the second-best solution that range between 4% and 26%. The different TES approaches yield distinctive results. In general, the ES approach, with or without the τ selection, improves the performance of the state-of-the-art algorithms. For instance, TES-RNN steadily guarantees very low SLA violation probabilities, which is a 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 but incurs much more frequent SLA violations. Transformer, in both the ES and TES configurations, improves the performance of the pure Transformer counterpart for all the services, although TES-Transformer incurs a higher SLA violation rate due to its aggressive forecasting. INFOCOM19, RNN, and Transformer, when compared to TES, induce a substantially higher overprovisioning that often helps limit SLA violations, at the expense of an overall higher expenditure.

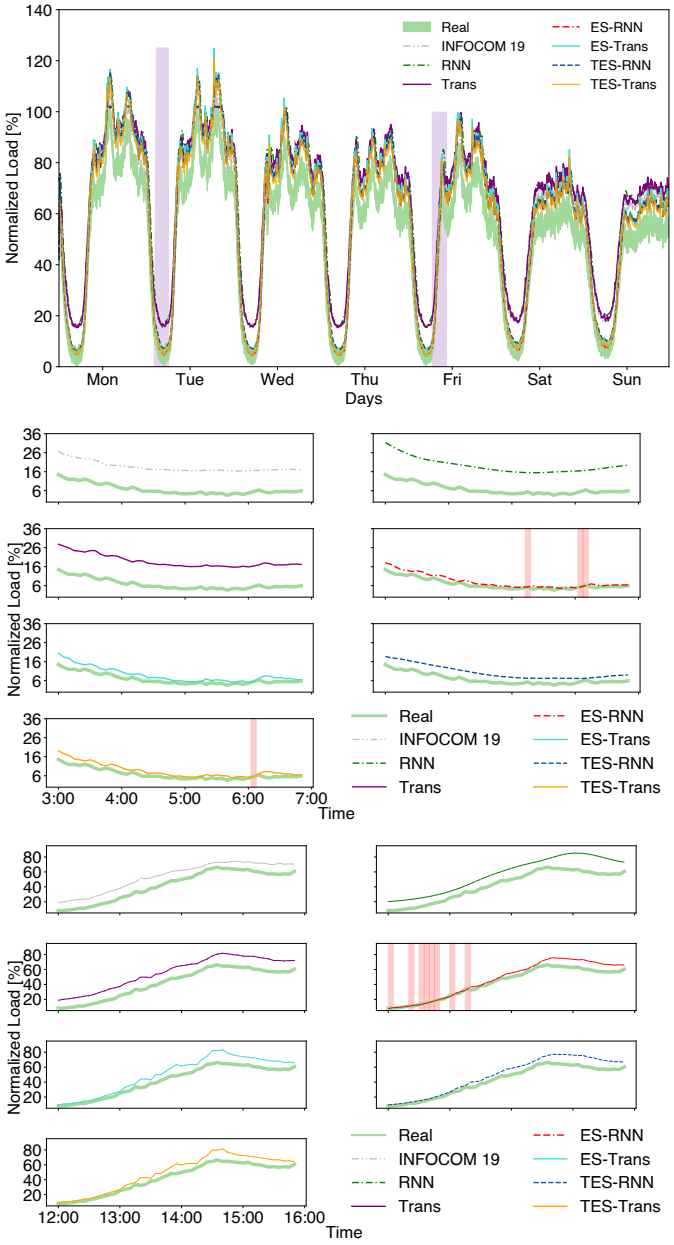

Fig. 8: Time series of the real traffic generated by the Facebook slice and of the relative capacity predictions of the different forecasting models. (top) Weekly time series, with highlighted time intervals for close-in analysis. (middle) Zoomed view of the 3:00-7:00 interval of Tuesday. (bottom) Zoomed view of the 12:00-16:00 interval of Friday. Whenever present, SLA violation periods are marked as red-shaded areas in the plots.

2) In-depth analysis of one prediction instance: To gain an additional understanding of the behaviors of the forecasting models presented above, we detail a representative case of capacity prediction in Figure 8. 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

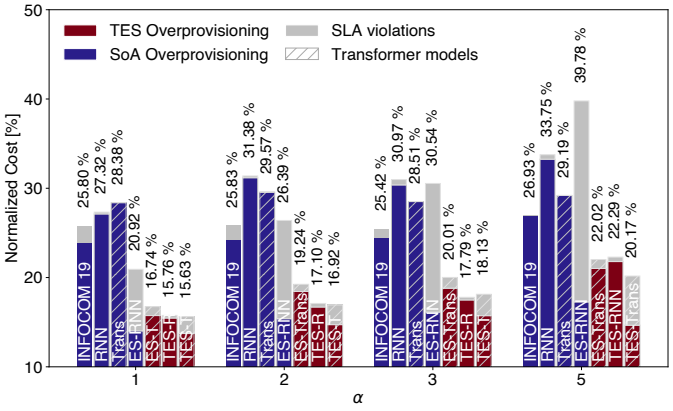

Fig. 9: Additional capacity allocation and SLA costs of INFOCOM19, RNN, Transformer, ES-RNN, ES-Transformer, and TES models versus α and for the Facebook slice.

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 and ES-RNN help dimension 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 (4), 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 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, ES-Transformer and TES models forecast a smoother capacity curve that stays above minor fluctuations, and hence yield a resource provisioning similar to ES-RNN but while avoiding numerous 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 (4). By varying the parameter, the operator shall be able to steer the capacity forecast to favor one source of cost over the other, as explained in Section V-A.

Figure 9 illustrates the capability of each model to enforce the desired control above by trading off overprovisioning cost with SLA violations cost. The plot shows, for the case of the Facebook slice, the normalized cost determined by each predictor, as α sweeps values from 1 (relatively low SLA violation cost) to 5 (high SLA violation cost). We observe that TES-Transformer and TES-RNN yield the best performance in all settings. Also, TES-RNN 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 TES-RNN. ES-RNN is instead unable to modulate the SLA violation cost, which in fact surprisingly grows with α.

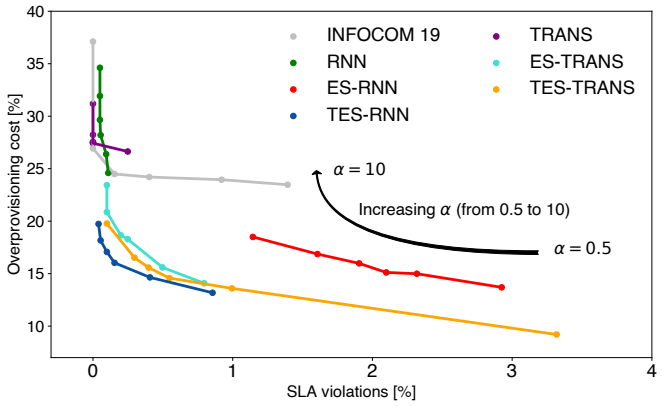

Fig. 10: Limits in terms of SLA violations and overprovisioning costs that can be attained by ES models, TES models, and the chosen benchmarks for the Facebook slice.

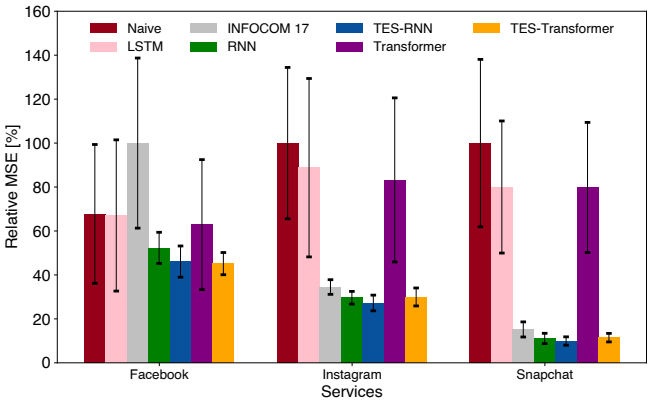

Fig. 11: Prediction accuracy in terms of MSE for Naive, LSTM, INFOCOM17, RNN, Transformer, and TES forecast.

The reason for the counter-intuitive ES-RNN performance can be explained by the breakdown of the two cost sources, in Figure 10. The plot illustrates, for each case in Figure 9: (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 across all models, and higher values of α always entail fewer violations, as one would expect. However, while TES models, 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 9.

More generally, Figure 10 gives a clear view of the operating points of each forecasting method. ES-Transformer and TES models offer the best options to the operator, as their configurations simultaneously provide fewer SLA violations and lower overprovisioning costs than the state-of-the-art 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, the proposed models bring the best of the other models: for any possible operating point of INFOCOM19, ES-RNN, or RNN, we can choose an α that improves cost on both dimensions.

B. Forecasting for anomaly detection

The second use case we consider is that of anomalous load detection at base stations introduced in Section V-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 that 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 three different services, *i.e.*, Facebook, Instagram, and Snapchat. For each slice, we consider different base stations and assess the performance of the anomaly detection algorithm discussed in Section V-B that relies on forecasting models of the slice traffic at each such base station.

To support the anomaly detection decision, the proposed models (ES-Transformer, TES-Transformer, and TES-RNN) are trained with an MSE loss function, according to the discussion in Section V-B. With such a loss function, our models operate as traditional mobile traffic forecasting models; this steers our choice of benchmark to the following models.

- Long Short-Term Memory (LSTM) is a simple NN-based model made of two fully-interconnected layers that is not specifically designed for mobile network traffic forecasting at base station level.
- INFOCOM17 [12] 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 the 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 a very long time horizons [61], or forecasting for the radio access [62].
- ES-RNN, RNN, and Transformer, as discussed in Section VI-A. In this case, the models are trained with an MSE loss function.

1) Mobile traffic prediction accuracy: We start our assessment by comparing the sheer accuracy of the proposed models against the benchmarks in the task of point forecasting mobile traffic at base stations. We also consider for comparison a

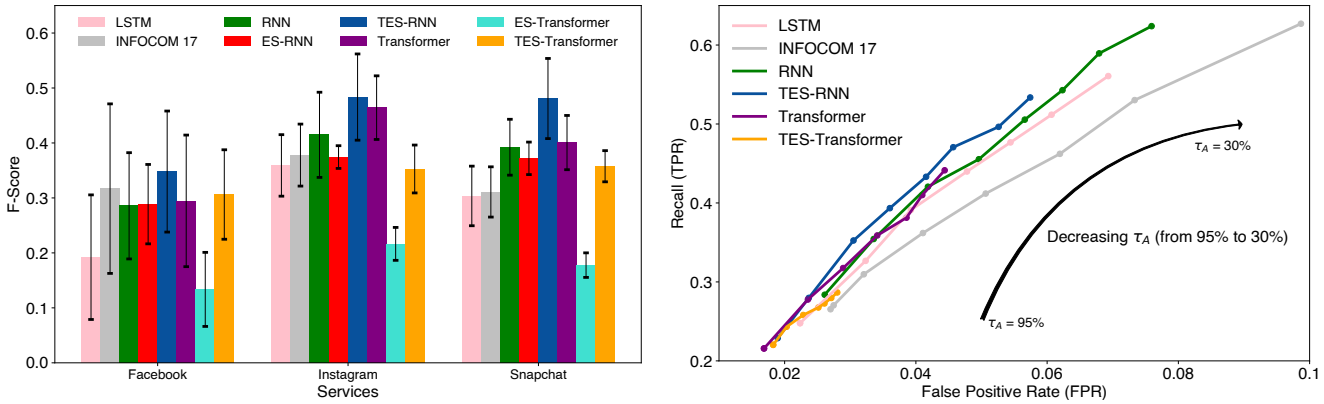

Fig. 12: Comparing LSTM, INFOCOM17, RNN, Transformer, ES, and TES models. (left) F1 Scores obtained with the algorithm discussed in Subsection V-B, and (right) Receiver Operating Characteristic (ROC) Curve for the Facebook service, for different τ_A thresholds. Subplots do not show results for the Naive model due to the deterministic nature of its forecasts.

Naive model, which uses the current value of the timeseries to predict the value of the following timestep. Figure 11 shows the results for all models and services averaged over five different base stations. We remark that results for the ES models are not shown because, as extensively explained in Section IV-B, training such models with an MSE loss function in the presence of mobile data traffic yields exceedingly high overestimation (see, *e.g.*, Figure 1). Independent of the test configuration, TES-RNN yields the most accurate prediction: the average MSE reduction across all base stations is in the 20%-40% range, peaking at 55% for the Facebook case when compared to INFOCOM17 and at 90% for the Snapchat service in comparison to the Naive model. TES-Transformer yields performance similar to that of TES-RNN.

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 User Plane Function (UPF) [54] to monitor, *e.g.*, the excessive usage of the network by a terminal, and has to generate an alarm if such an 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 V-B. Figure 12a shows the performance of all the benchmarks in terms of F1 Score, averaged over the five selected base stations, for the three selected application types. TES-RNN always achieves the best performance, while competitors provide unbalanced results, highly depending on the considered application. In particular, the average F-Score gain of TES-RNN across all services is in the 0.1-0.15 range, peaking at 0.31 for Snapchat when compared to ES-Transformer. The baseline ES-RNN and the proposed ES-Transformer and TES-Transformer almost always trigger 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 task, we also evaluate its effectiveness with variable τ_A values in Figure 12b, showing the ROC curve for the selected benchmarks and proposed models. ES models are not shown as their highly overestimated forecasts trigger a huge number of false positives, which explains the poor ES models performance in Figure 12a. TES-RNN always yields the best pairing between the Recall and the False Positive Rate (FPR), for all the considered τ_A in the range between 30% and 95%. TES-Transformer, instead, yields intermediate performance when compared to the other benchmarks.

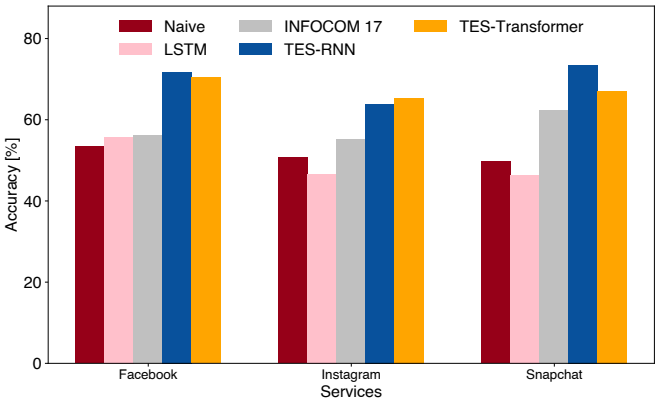

Fig. 13: Load balancing accuracy.

C. Forecasting for load balancing

The third use case we consider for the comparative evaluation is that of anticipatory load balancing at base stations introduced in Section V-C. As in the use case of Subsection VI-B, we also operate in a virtualized network environment running an end-to-end network slicing model, where strong quality of service guarantees are met by dedicating resources to different slices already in the radio access. In such settings, the load-balancing NI operates at the granularity of individual services, and we consider three different slices, each accommodating the three services of Facebook, Instagram, and Snapchat. For each service, we consider different base stations and evaluate the performance of a load balancer that leverages forecasting models of the traffic at the slice for each such base station.

To support the load balancing decision, the proposed models (TES-RNN and TES-Transformer) are trained with a suitable

MSE loss function, according to the discussion in Section V-C. With such a loss function, also for this use case, we choose as benchmarks the model LSTM and INFOCOM17 [12], as well as the Naive model already presented in Subsection VI-B1. To investigate the performance of a load balancer, we consider the case of UE association performance. We emulate a UE initial attachment or handover scenario in a dense deployment where multiple base stations may offer similar radio channel quality. A prominent criterion for UE association is the expected load that each base station will experience in the following minutes; in a sliced scenario, this becomes the demand generated by a specific service at each candidate base station. To balance the load across base stations, the soliciting UE should be assigned to the base station with the lowest forecasted traffic for the requested slice. As discussed in Section V-C, such functionality is part of the 5G standard, and would run in the load balancing NI engine at the PCF. We consider for our tests a load balancer that, in the presence of a slice association request that can be possibly satisfied by two base stations, selects the one with the lowest anticipated load for the requested slice in the following 5 minutes. The decision is thus driven by the per-service traffic forecast performed by the evaluated models.

Figure 13 shows the accuracy of the load balancer, *i.e.*, the fraction of UE association requests that are correctly directed to the base station with the lowest load on a specific slice in the following timestep. Decisions based on TES-RNN and TES-Transformer predictions have an average accuracy close to 70%, which is a decent performance for the very high variance of traffic at 5-minute timescales. This is also 6%-16%, 15%-27%, and 13%-24% higher than the accuracy granted respectively by INFOCOM17, LSTM, and the Naive model.

D. Complexity

We conclude by analyzing the complexity of the proposed solutions, in terms of average training time over the same number of epochs and memory usage for experiments running on an NVIDIA A100 GPU. The results for the first metric are reported in Table IV: in the first use case, Transformer models required on average lower training times than the equivalent RNN models, while yielding similar performance as observed at the beginning of Section VI. In particular, the high-performing TES-Transformer model required an average of 2.45 minutes per capacity allocation experiment, which is in the order of the average training time of the less-performing RNN model (2.85 min) but faster than the equivalent TES-RNN model with a mean of 5.1 minutes per run. Similar considerations apply to the second and third use cases, where all the models are trained with the same MSE loss function. Overall, TES-RNN and TES-Transformer can be trained as fast as the respective less optimized models (*i.e.*, ES-RNN and RNN for the recurrent neural network architecture, and ES-Transformer and Transformer) while guaranteeing the advantages in the forecasting performance. This is obtained with no additional burden on the memory of the GPU in which the models run. The results reported in Table V show that all the considered models require around 1400MiB, which represents only 3.4% of the total memory of the A100 GPU.

TABLE IV: Training time of the benchmarks, in minutes.

Use Case	RNN	ES-RNN	TES-RNN	Trans	ES-Trans	TES-Trans
1	2.85	4.93	5.1	0.17	2.3	2.45
2-3	2.64	4.43	4.87	0.12	2.12	2.29

TABLE V: GPU memory usage of the benchmarks, in MiB.

Use Case	RNN	ES-RNN	TES-RNN	Trans	ES-Trans	TES-Trans
1	1422	1422	1422	1400	1400	1400
2-3	1422	1422	1422	1420	1420	1420

VII. Conclusion

In this paper, we explored the potential of statistical modeling for anticipatory traffic management using deep learning. By avoiding high average-to-peak ratios in the input training data ES allows a more efficient forecast of the traffic time series. In this paper, we propose the TES framework[3], that effectively improves the normalization of bursty time series through a factor τ and couple it to two different neural network architectures based on RNNs and Transformer architectures, to prove the solution flexibility. We benchmarked both TES-RNN and TES-Transformer on three relevant use cases for network management: capacity allocation, anomaly detection, and load balancing. In the three scenarios, our solutions achieve performance gains with respect to state-of-the-art benchmarks in the order of, respectively, 4%-26%, 20%-40%, and 6%-16% range depending on the scenario. This paper demonstrates the advantages of the TES framework in time series forecasting, showing how it outperforms state-of-the-art baselines. Our results prove that a hybrid approach can enhance even advanced AI/ML solutions, paving the way for future research that could extend the framework to models beyond RNNs or Transformers. While we highlight these benefits, the challenge of understanding the limited performance of forecasting techniques on network traffic remains a separate research effort beyond the scope of this work.

Acknowledgements

The ORIGAMI project has supported this work, which has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101139270. It is also co-funded by the European Union under Grant Agreement No. 101191936 (SUSTAIN-6G). The views and opinions expressed are solely those of the author(s) and do not necessarily reflect those of the SUSTAIN-6G consortium parties, the European Union, or the SNS JU (granting authority). Neither the European Union nor the granting authority can be held responsible for them. The work of IMDEA Networks was supported by the 6G-IRONWARE project funded under grant CNS2023-143870 by MICIU/AEI/10.13039/501100011033 and the EU NextGenerationEU/PRTR. Finally, the work of A. Banchs has been supported by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D 6G-CLARION project.

[3]The code of the TES framework is publicly accessible at the following link: https://doi.org/10.5281/zenodo.16045349

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] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 58–72. [Online]. Available: https://doi.org/10.1145/2934872.2934891

[3] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, X. S. Shen, and W. Zhuang, "AI-Native Network Slicing for 6G Networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 96–103, 2022.

[4] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, "SliceOps: Explainable MLOps for Streamlined Automation-Native 6G Networks," *IEEE Wireless Communications*, vol. 31, no. 5, pp. 224–230, 2024.

[5] L. E. Chatzieleftheriou *et al.*, "Network Intelligence in Action: the DAEMON Perspective," in *Proc. of European Conference on Networks and Communications & 6G Summit*, Antwerp, Belgium, Jun. 2024, pp. 1–6.

[6] M. Milani, D. Bega, M. Gramaglia, P. Serrano, and C. Mannweiler, "ATELIER: Service Tailored and Limited-Trust Network Analytics Using Cooperative Learning," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3315–3330, 2024.

[7] A. T.-J. Akem, M. Gucciardo, and M. Fiore, "Flowrest: Practical Flow-Level Inference in Programmable Switches with Random Forests," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[8] 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*, 2020, pp. 794–803.

[9] L. L. Schiavo, G. Garcia-Aviles, A. Garcia-Saavedra, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "CloudRIC: Open Radio Access Network (O-RAN) Virtualization with Shared Heterogeneous Computing," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 558–572. [Online]. Available: https://doi.org/10.1145/3636534.3649381

[10] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "EdgeBOL: automating energy-savings for mobile edge AI," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 397–410. [Online]. Available: https://doi.org/10.1145/3485983.3494849

[11] 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.

[12] 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, May 2017, pp. 1–9.

[13] 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, Jun. 2019, pp. 1–6.

[14] 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, Apr. 2019, pp. 280–288.

[15] E. Coronado, R. Behravesh, T. Subramanya, A. Fernàndez-Fernàndez, M. S. Siddiqui, X. Costa-Pérez, and R. Riggio, "Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2535–2578, 2022.

[16] Q. Y. Ding, X. F. Wang, X. Y. Zhang, and Z. Q. Sun, "Forecasting Traffic Volume with Space-Time ARIMA Model," in *Advanced Manufacturing Technology, ICAMMP 2010*, ser. Advanced Materials Research, vol. 156. Trans Tech Publications Ltd, 1 2011, pp. 979–983.

[17] D. Zhou, S. Chen, and S. Dong, "Network traffic prediction based on ARFIMA model," 2013. [Online]. Available: https://arxiv.org/abs/1302.6324

[18] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, 2019.

[19] L. Yu, M. Li, W. Jin, Y. Guo, Q. Wang, F. Yan, and P. Li, "STEP: A Spatio-Temporal Fine-Granular User Traffic Prediction System for Cellular Networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3453–3466, 2021.

[20] N. Zhao, A. Wu, Y. Pei, Y.-C. Liang, and D. Niyato, "Spatial-Temporal Aggregation Graph Convolution Network for Efficient Mobile Cellular Traffic Prediction," *IEEE Communications Letters*, vol. 26, no. 3, pp. 587–591, 2022.

[21] Y. Yao, B. Gu, Z. Su, and M. Guizani, "MVSTGN: A Multi-View Spatial-Temporal Graph Network for Cellular Traffic Prediction," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 2837–2849, 2023.

[22] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207019301153

[23] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207019301128

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

[25] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, and X. Costa-Pérez, "Ai-based autonomous control, management, and orchestration in 5g: From standards to algorithms," *IEEE Network*, vol. 34, no. 6, pp. 14–20, 2020.

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

[27] 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.

[28] 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.

[29] 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, Dec. 2018, pp. 353–365.

[30] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169207089900125

[31] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249

[32] 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.

[33] 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, Jun. 2020.

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

[35] 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)*, 2016, pp. 903–908.

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

[37] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11 106–11 115, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17325

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[39] R. E. Turner, "An Introduction to Transformers," 2024. [Online]. Available: https://arxiv.org/abs/2304.10557

[40] Y. Hu, Y. Zhou, J. Song, L. Xu, and X. Zhou, "Citywide Mobile Traffic Forecasting Using Spatial-Temporal Downsampling Transformer Neural Networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 152–165, 2023.

[41] Q. Liu, J. Li, and Z. Lu, "ST-Tran: Spatial-Temporal Transformer for Cellular Traffic Prediction," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3325–3329, 2021.

[42] B. Gu, J. Zhan, S. Gong, W. Liu, Z. Su, and M. Guizani, "A Spatial-Temporal Transformer Network for City-Level Cellular Traffic Analysis and Prediction," *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 9412–9423, 2023.

[43] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[44] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," 2020. [Online]. Available: https://arxiv.org/abs/1905.10437

[45] U. SA, *DLinear Model Documentation - Darts: Time Series Made Easy*, 2023. [Online]. Available: https://unit8co.github.io/darts/generated_api/darts.models.forecasting.dlinear.html

[46] Y. Ge, Y. Zhang, K. Shi, and H. Li, "A moment cross predictor for non-stationary mobile traffic forecasting," in *2024 IEEE/CIC International Conference on Communications in China (ICCC)*, 2024, pp. 2059–2064.

[47] M. Di Mauro, G. Galatro, F. Postiglione, W. Song, and A. Liotta, "Multivariate time series characterization and forecasting of voip traffic in real mobile networks," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 851–865, 2024.

[48] K. Wu, J. Lu, F. Lin, Y. Huang, C. Zhan, and L. Sun, "A realistic network traffic forecasting method based on vmd and lstm network," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2023, pp. 1–5.

[49] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120307516

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

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

[52] 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.

[53] 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.

[54] 3GPP TS 23.288 v17.4, "Architecture enhancements for 5G System (5GS) to support network data analytics services (Rel. 17)," May 2022.

[55] P. J. Rousseeuw and M. Hubert, "Anomaly detection by robust statistics," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 2, 2018. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1236

[56] 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, Jun. 2016, pp. 1050–1059.

[57] 3GPP TS 23.503 v16, " Policy and charging control framework for the 5G System (5GS); Stage 2," Mar. 2019.

[58] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[59] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, C. Ziemlicki, and Z. Smoreda, "Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 180–186. [Online]. Available: https://doi.org/10.1145/3143361.3143369

[60] O. Guhr, "Transformer Time Series Prediction," https://github.com/oliverguhr/transformer-time-series-prediction, 2024, proof of concept for a transformer-based time series prediction model.

[61] 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*, ser. Mobihoc '18. Association for Computing Machinery, 2018, p. 231–240. [Online]. Available: https://doi.org/10.1145/3209582.3209606

[62] 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.

Leonardo Lo Schiavo received his Ph.D. degree in Telematic Engineering in 2025 from Universidad Carlos III de Madrid, where he is currently working as a Postdoctoral Researcher. His research interests include virtualized Radio Access Networks, AI-driven network automation and traffic forecasting.

Genoveva García is a graduate in Telecommunication Technologies Engineering from Universidad Carlos III de Madrid (UC3M), where she is currently pursuing a Master's degree in Artificial Intelligence.

Prof. Marco Gramaglia got his Ph.D. in Telematics Engineering from Universidad Carlos III de Madrid (UC3M). Marco has contributed extensively to several research projects at both the European and national levels. His research interests include network automation, privacy, and AI-driven resource management. He co-authored more than 100 articles and, according to Google Scholar, his h-index is 35.

Marco Fiore is a Research Professor at IMDEA Networks Institute, where he leads the Networks Data Science group, and co-founder and CTO at Net AI. He received MSc degrees from University of Illinois at Chicago and Politecnico of Torino, a PhD degree from Politecnico di Torino, and a Habilitation à Diriger des Recherches from Université de Lyon. Marco has held tenured positions at Institut National des Sciences Appliquées de Lyon and National Research Council of Italy, and has been a visiting researcher at Rice University, Universitat Politècnica de Catalunya, and University College London. Marco's research is at the interface of mobile networks and data science, and has received funding from the European Commission and national agencies in Spain, France and Italy, as well as a number of recognitions that include two best paper awards at IEEE INFOCOM. Marco is a former Marie Curie fellow and Royal Society visiting research fellow, and a Senior Member of IEEE and ACM.

Albert Banchs (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees from the Polytechnic University of Catalonia (UPC-BarcelonaTech) in 1997 and 2002, respectively. He is currently a Full Professor with the University Carlos III of Madrid (UC3M), with double affiliation as the Director of the IMDEA Networks Institute. Before joining UC3M, he was with ICSI Berkeley in 1997, Telefonica I+D in 1998, and NEC Europe Ltd., from 1998 to 2003. He was an Academic Guest with ETHZ in 2012, a Visiting Professor with EPFL in 2015, 2013, and 2018, and a Fulbright Scholar with The University of Texas at Austin in 2019. He is the author over 150 publications in international conferences and journals and is the co-inventor of several patents.

Xavier Costa is ICREA Research Professor, Scientific Director at the i2cat Research Center and Head of 6G R&D at NEC Laboratories Europe. His team generates research results that are regularly published at top scientific venues, produces innovations that have received several awards for successful technology transfers, and participates in major European Commission R&D collaborative projects. He has held multiple leadership positions both in industry and research organizations, such as Deputy General Manager, Chief Researcher, Technology Board member, and Scientific Advisory Board Member. As a standards delegate, he contributed to multiple standardization bodies (e.g., IEEE 802.11, 802.16, WiFi Alliance, 3GPP, ...) and was recognized in several standards as a top contributor. He has served on the Organizing Committees of several conferences (including ACM MOBICOM, IEEE INFOCOM, WCNC, and Greencom), published papers of high impact, and holds about 100 granted patents. He has served as Editor at IEEE Transactions on Mobile Computing (TMC), IEEE Transactions on Communications (TCOM), and Elsevier Computer Communications journals (COMCOM). Xavier received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia (UPC) in Barcelona and was the recipient of a national award for his Ph.D. thesis.