ANNs Going Beyond Time Series Forecasting: An Urban Network Perspective

Jane Frances Pajo, George Kousiouris, Dimosthenis Kyriazis,
Roberto Bruschi, *Senior Member, IEEE,* and Franco Davoli, *Life Senior Member, IEEE*

***Abstract*—5G is expected to bring about disruptive industrial-societal transformation by enabling a broad catalogue of (radically new, highly heterogeneous) applications and services. This scenario has called for Zero-touch network and Service Management (ZSM). With the recent advancements in Artificial Intelligence (AI), key ZSM capabilities such as the runtime prediction of user demands can be facilitated by data-driven and Machine Learning (ML) methods. In this respect, the paper proposes a runtime prediction approach that transforms time series forecasting into a simpler multivariate regression problem with Artificial Neural Networks (ANNs), structurally optimized with a genetic algorithm (GA) metaheuristic. Leveraging on a novel set of input features that capture seasonality and calendar effects, the proposed approach removes the prediction accuracy's dependence on the temporal succession of input data and the forecast horizon, which is typical in time series forecasting. Evaluation results based on real telecommunications data show that the GA-optimized ANN regressor has better prediction performance compared to 1-day and 1-hour ahead forecasts obtained with state-of-the-art Multi-seasonal Time Series (MSTS) and Long Short-Term Memory (LSTM) forecasting models by an average of ~59% and ~86%, respectively. Furthermore, despite its longer training times compared to the baseline models, the proposed ANN regressor relaxes the monitoring requirements in 5G dynamic management systems by allowing less frequent re-training offline.**

***Index Terms*—Artificial Neural Networks, Genetic Algorithm, Network Dynamics, Runtime Prediction, Time Series, ZSM.**

I. Introduction

OVER the years, learning complex system dynamics has maintained significant interest in the research scene, as well as in various industrial domains, for its notable potential in autonomic management and control. At a networking perspective, the runtime prediction of user demands at various management levels (e.g., infrastructure-, service- and application-level) and coverage granularities (e.g., cell-, city- and nationwide-level) would not only facilitate the dimensioning of the network/service/application, but also enable a wide range of management and control mechanisms such as the dynamic (and proactive) provisioning of the underlying resources. Indeed, this is a fundamental component towards enabling Zero-touch network and Service Management (ZSM) [1], as well as the disruptive industrial-societal transformation brought forth by 5G [2].

User demand dynamics are usually recorded as temporal data, which have been widely investigated through time series analysis and forecasting, as well as regression analysis, among others. Moreover, it is worth noting that they naturally have layered *seasonality* (e.g., daily, weekly, monthly, etc.), and may exhibit the so-called *calendar effect* (e.g., weekends, holidays, sales, etc.), as well [3].

With the recent advancements in Artificial Intelligence (AI), data-driven and Machine Learning (ML) methods have opened new directions towards predictive analytics. Among others, bio-inspired Artificial Neural Networks (ANNs) have gained particular interests for their ability to model noisy and nonlinear systems, by learning from examples. In fact, numerous works in the literature (e.g., [4]-[7]) exploit past temporal data as input to extend ANNs' high prediction accuracies to time series forecasting. The training times of ANN-based time series predictors are, however, substantially longer than typical regressors, and such models usually have a limited usable prediction horizon [5]. Furthermore, they depend on the temporal succession of data and, hence, call for re-training as the series are updated with new observations.

With this in mind, the main contribution of this paper is a novel input feature set and framework for transforming time series forecasting into a simpler multivariate regression problem, using ANNs (that are structurally optimized with a Genetic Algorithm (GA) metaheuristic, as in [4]) in the context of network activity runtime prediction. In particular, by leveraging a set of seasonal and calendar features, the dependence of the modeling and forecasting steps on the temporal succession in the data is removed. Hence, the proposed approach is capable of predicting any future value based only on the fed inputs, as well as capture possible (ir)regularities in the dynamics brought forth by seasons and special events.

Manuscript received *date*; revised *date*. This work was partially supported by the H2020 Innovation Actions SPIDER and 5G PPP 5G-INDUCE, funded by the European Commission under Grants 833685 and 101016941, respectively.

J. F. Pajo, R. Bruschi and F. Davoli are with the Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, and with the National Laboratory of Smart and Secure Networks (S2N) of the Italian National Consortium for Telecommunications (CNIT), Genoa, Italy (e-mail: jane.pajo@tnt-lab.unige.it, roberto.bruschi@unige.it, franco.davoli@unige.it).

G. Kousiouris is with the Department of Informatics and Telematics, Harokopio University of Athens, Greece (e-mail: gkousiou@hua.gr).

D. Kyriazis is with the Department of Digital Systems, University of Piraeus, Greece (email: dimos@unipi.gr).

The remainder of this paper is organized as follows. Firstly, Section II sets the technological scene for time series runtime prediction. Section III then describes the proposed GA-optimized ANN modeling of the network activity in an urban area. Section IV provides an overview of the dataset considered in this work. Performance evaluation results are presented in Section V, and finally, conclusions are drawn in Section VI.

II. Relevance, Challenges and Requirements of Time Series Runtime Prediction

Time series runtime prediction in networking is becoming more crucial than ever before with the onset of the ZSM initiative. Understanding the dynamics of user demands and of the corresponding network resources' usage is foreseen to be the key in driving smarter dimensioning and provisioning of next-generation networks, services and applications. Usually expressed as temporal data, such dynamics can be explored using state-of-the-art time series analysis and forecasting techniques, as well as regression analysis, among others.

Box-Jenkins' Autoregressive Integrated Moving Average (ARIMA) model and *Holt-Winters* exponential smoothing are two of the earliest approaches for capturing various patterns within time series; their main difference lies in the way past series values impact the forecast – specifically, for the former they are weighted uniformly, while for the latter the weights follow an exponential function. Numerous works in the literature build on these two towards multi-seasonal time series (MSTS) forecasting. For instance, seasonal parameters can be used with ARIMA (denoted as SARIMA) to capture seasonality, as well as adding exogenous inputs (denoted as SARIMAX) for multivariate time series forecasting.

On the other hand, the presence of noise and nonlinearity has motivated the use of more advanced solutions based on ANNs. The authors in [5] considered GA-optimized Nonlinear Autoregressive with Exogenous inputs (NARX) ANNs for service demand prediction, with the past series values as the driving variables. Recurrent ANNs (RNNs) have also gained popularity for time series forecasting – in view of this, recent works such as [6] consider different RNN models (i.e., Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)) for forecasting series values like the next day traffic dynamics in datacenters. Furthermore, for multivariate time series forecasting, the authors in [7] proposed ensembling RNNs and convolutional ANNs (CNNs) along with an autoregressive model in order to capture long- and short-term patterns, while addressing the so-called *scale insensitivity* problem of ANNs.

A common limitation of the aforementioned approaches is the dependence on the temporal succession of data samples used in building the models, as well as during prediction for ANN-based models, since they use a sliding window of past series values as input. Keeping a model up-to-date and ensuring a valid forecast horizon entail re-training as new observations become available. For autoregressive models, the receding accuracies further in the forecast horizon and inability to capture nonlinearities are the main concerns. On the other hand, for the ANN-based models, the time cost of (re-)training, structural optimization, as well as the various vulnerabilities in the training process (e.g., non-representative training sets, temporal succession of time series data, etc.) remain open issues – which a number of published works in the literature have engaged in addressing. Among others, [8] used *hyperparameter* search strategies; [9] used *k*-fold cross validation to select the best ANN structure among the folds; [10] explored profile-based tuning of LSTM hyperparameters; and [4][5] used a GA metaheuristic to find a suboptimal combination of the ANN structural parameters. However, as previously anticipated, the usage of GA-optimized ANNs as time series predictors with past series values as input takes much more time (i.e., up to over 20 hours) to train than typical regression ANNs, and usually has a limited usable prediction horizon (e.g., 10-20 steps ahead) [5].

In this work, GA-optimized ANNs are used with a novel set of generic input features that capture both seasonality and calendar effects. The goal is to transform time series forecasting into a simpler multivariate regression problem, keeping ANN's high prediction performance while removing the dependence on the temporal succession of data samples and forecast horizon. This relaxes not only re-training requirements but also monitoring ones and, hence, enhances 5G management systems' fault tolerance against intermittent failures of monitoring subsystems. Furthermore, with a smaller number of input features, the training times of GA-optimized models are expected to be substantially reduced, as well.

III. Modeling the Network Activity with ANNs

This work explores transforming time series forecasting into a simple regression problem by defining some seasonal and calendar features as ANN inputs, which results in predictions that are indifferent to the forecast horizon, and only depend on the fed inputs.

A. Input Features

The idea is to not only capture the multi-seasonal effects with the approach, but also the calendar effects that bring possible irregularities in the dynamics – may it be a surge or a drop in the network activity levels. Hence, aside from the date/time information (decomposed as ***DayOfTheWeek*** and ***HourOfTheDay***), we additionally define and evaluate the following features: (a) ***PeriodType***, (b) ***isHoliday*** and (c) ***DaysToNextHoliday***, as inputs being mapped with the target variable, ***NetworkActivity***, exploring their ability to encode any calendar effects in the dataset. The ***PeriodType*** variable admits values from the set {0,0.5,1}, which are associated to *normal days*, *holiday period* and *sales*, respectively. Although the different kinds of holidays are not differentiated in this work, ***PeriodType*** is used to somehow add weight on holiday periods (e.g., weekends around holidays and year-end holidays), as well as special sales periods (e.g., Black Friday through Cyber Monday). The binary variable ***isHoliday*** flags the actual holidays, while the ***DaysToNextHoliday*** variable is a countdown to the next holiday; the latter can be useful to capture any trends as holidays or holiday periods are approaching.

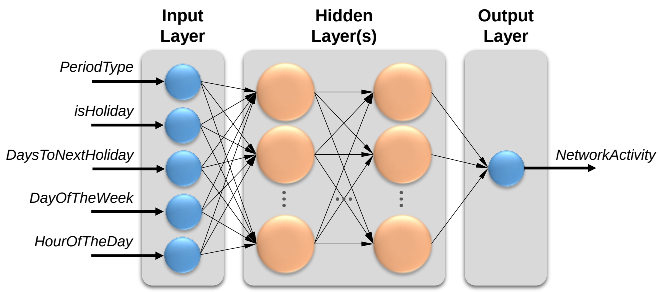

Fig. 1. ANN architecture.

It follows that the training dataset used in this work is structured with 6 columns – 5 for the aforementioned input features and 1 for the target.

B. GA-based ANN Structure Optimization

The operation of ANNs basically derives from the structural parameters used – specifically, the *number of layers*, the *neurons per layer* and their corresponding *transfer functions*. Note that the term 'transfer function' adopted in this work is based on the GNU Octave ANN library; on the other hand, the widely used TensorFlow-Keras library in Python adopts the more customary term 'activation function'.

While the number of neurons in the input and output layers are defined by the dataset structure (i.e., in our case, there are 5 neurons in the input layer and 1 neuron in the output layer), everything else is tunable. Nonetheless, ANN structural optimization still remains an open issue, which is usually addressed using search strategies [8], *k*-fold cross-validation [9] and evolutionary algorithms [4][5], among others.

Fig. 1 illustrates the ANN architecture of the proposed runtime predictor. To obtain its optimal structure, this work adopts the numerical method in [4] that automates the optimization process using a GA metaheuristic, which in turn recursively evaluates various combinations of structural parameters during the training and intermediate validation processes. Starting from an initial random population of ANN structures, ANNs are created, trained and dynamically updated based on the structural parameters generated by the GA in each succeeding generation. It is worth noting that in order to quickly converge to a suboptimal solution, the GA adopts the bio-inspired *elitism*, *crossover* and *mutation* operators in creating the population for subsequent generations, resulting in a recursive evolutionary process that is driven by the Mean Squared Error (MSE) on the training set.

GA is specifically considered for an automated approach in ANN structure optimization with no human intervention, such that it can be included in next-generation network/service management approaches. It also adapts to different data patterns that may need different logics and, therefore, different ANN base characteristics (such as types of transfer functions). Lastly, GA results in faster convergence than a random selection iteration.

C. Model Training and Validation

Feed-forward back-propagation ANNs are considered and trained with the *Levenberg-Marquardt* algorithm [11] according to the MSE training goal. The best ANNs (i.e., models resulting in MSE values that satisfy the set goal) of each generation are then saved for an intermediate validation phase at the completion of the GA. This step is aimed at enhancing the models' *generalizability*, and uses a subset of the training set, which includes samples that are not used in the actual training (i.e., 30% of the training set, as in [4]). The Mean Absolute Percentage Error (MAPE) between the actual and predicted ***NetworkActivity*** values in this subset is used to assess the generalizability of each model in the best ANNs' pool, such that the final model is selected as the one with the least MAPE value (denoted as "Best MAPE"). This model is validated against the final test set (i.e., not used during training or best model selection) and can then be used for runtime prediction.

In this work, we further explore a *k*-fold cross validation approach to evaluate inter-fold performance (e.g., [9]), as well as to investigate *federated* learning. Since the proposed approach removes the temporal succession dependencies among data samples, the training and test sets for each fold are obtained through stratified sampling of the data according to the ***DayOfTheWeek*** and ***HourOfTheDay***. This results in varying values of ***PeriodType***, ***isHoliday***, ***DaysToNextHoliday*** and ***HourOfTheDay*** for a given calendar day. The performance of the resulting ANNs for each fold will then be evaluated using an independent test set and analyzed to understand how to drive the final predictor – whether to adopt the best-performing model (e.g., the one with the least MAPE), or to integrate the *k* models (e.g., the final prediction as the mean of the individual predictions).

Fig. 2 illustrates the conceptual framework of the proposed runtime prediction approach, leveraging GA-optimized ANN regressors in a *k*-fold cross validation methodology.

IV. Dataset

Starting with the multi-source open dataset of Telecom Italia's *Big Data Challenge*, the telecommunications data over Milan's urban area [12] are used as bases in this work. In more detail, the *Milano Grid* area includes around 33 cities/towns in Italy's Lombardy region, over which a two-month worth of mobile network activity data has been collected. Although the Milano Grid is originally subdivided into 10,000 235m × 235m square areas, we consider the aggregate values on the entire grid in order to have a glimpse on the urban network perspective, resulting in a single time series and hence, location attributes were not necessary in this work.

A detailed description of the dataset and the data acquisition process can be found in [13]. Putting it briefly, the telecommunications data are derived from Call Detail Records (CDRs) of Telecom Italia's mobile network, which are generated for logging events related to the users' SMS, calls and Internet activities. Based on the CDRs, the dataset includes values that are proportional to the network interactions (rather than the actual load itself); these can be regarded akin to service requests, and hence, considered as such hereinafter. Although the original data samples are in 10-minute intervals, the dataset has been pre-processed to consider the hourly averages of the network interactions in this work.

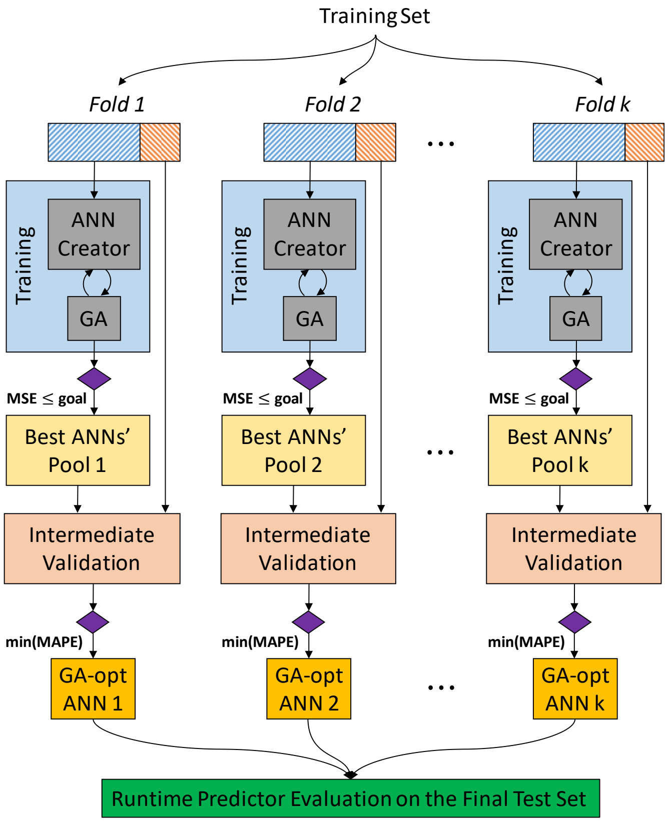

Fig. 2. Conceptual framework of the proposed runtime prediction approach.

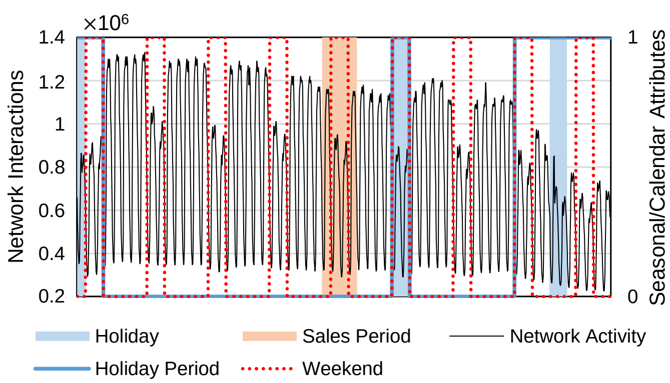

Fig. 3. Urban level network activity dynamics (November - December 2013) in terms of the average hourly network interactions, along with some seasonal and calendar attributes.

For simplicity, but without loss of generality, we suppose that the different types of network interactions (i.e., *SMS-in, SMS-out, Call-in, Call-out* and *Internet traffic*) have the same weight, considering the current shift towards IP-based messaging and calling services (e.g., WhatsApp, Facebook's Messenger, etc.). Summing up their values across both network interaction types and square areas, we obtain an aggregate time series for the Milano Grid's network activity. Fig. 3 illustrates the two-month worth of urban level network activity dynamics derived from Telecom Italia's dataset, along with some seasonal and calendar attributes. A decreasing trend can also be observed towards the year-end holidays, which can be attributed to people going out of town for vacation; this can be potentially captured by the ***DaysToNextHoliday***.

It is worth noting that the values in the dataset (dated 2013) may need to be scaled accordingly to account for the growth in volume from 2013 to 2020. For instance, scaling the Internet activity by a factor of 17, based on the growth in mobile data traffic produced by smartphones in Western Europe [14], and possibly, the network interactions related to the SMS and calls by other factors. However, since the focus of this work is to capture the inherent seasonality and calendar effects in the dataset (i.e., aspects that would not be affected by scaling), the values are kept unscaled.

V. Performance Evaluation

In this section, the evaluation details and outcomes of the proposed network activity runtime predictor based on GA-optimized ANNs are presented.

An ANN model with 5-fold cross-validation is generated for the entire Milano Grid area. The GNU Octave implementation in [4] is used, in which the GA explores ANN structures with 3-5 layers, 1-30 neurons/layer (excluding the inputs and outputs), and 1-3 types of transfer functions (i.e., *logsig*, *tansig* and *purelin*). The algorithm runs for 30 generations, each one considering a population of 20 ANN structures that is updated according to the elitism, crossover and mutation operators (with the default Octave parameters) for the subsequent generation. Each ANN structure is then trained for 150 epochs, with a maximum training time of 500s and a training goal of 10^{-7} for the MSE.

The proposed **ann-5features** (ANN with 5 input features – ***PeriodType***, ***isHoliday***, ***DaysToNextHoliday***, ***DayOfTheWeek*** and ***HourOfTheDay***) is evaluated with models based on the MSTS forecasting method available in R distributions and LSTM implementation of the TensorFlow-Keras bundle library available in Python. Particularly, two MSTS-based models **msts-*x*** and two LSTM-based models **lstm-*x*** are considered for comparison, where *x* indicates the forecast horizon in hours (i.e., **msts-1step** and **lstm-1step** for 1-hour ahead forecasts, **msts-24steps** and **lstm-24steps** for 1-day ahead forecasts, respectively).

It is worth noting that MSTS-based models naturally yield receding accuracies moving further a given forecast horizon, which only worsen in case of unexpected events such as a sudden surge/drop in the network activity levels. Moreover, while the MSTS and LSTM 1-step ahead forecasting are the better comparisons in terms of accuracy, the multiple-steps ahead models are also fair comparisons in terms of the enabled use cases of the proposed ANN model, with its ability to generate predictions for any given hour/day/week in the future, as defined by the input provided.

For the ANN models, the training and test sets for each fold are obtained through stratified sampling with respect to the ***DayOfTheWeek*** and ***HourOfTheDay***, as previously anticipated. Starting with the Milano Grid training set, the GA-optimized ANN regressors are then obtained by following the

Table I: GA-optimized ANNs.

Fold	Structure *transfer function (# of neurons) per layer*	Best MAPE
1	*logsig (5) - tansig (3) - logsig (2) - purelin (1)*	**5.02%**
2	*logsig (5) - tansig (3) - tansig (3) - purelin (1)*	**4.93%**
3	*tansig (5) - tansig (2) - tansig (2) - tansig (1)*	**6.38%**
4	*logsig (5) - logsig (3) - tansig (3) - logsig (3) - purelin (1)*	**5.21%**
5	*tansig (5) - tansig (3) - tansig (2) - tansig (3) - tansig (1)*	**6.49%**

framework in Fig. 2 and over 500 models are saved to the best ANNs' pool in each fold. Examples of GA-optimized structures generated for the proposed **ann-5features** are reported in Table I, along with their respective best MAPE values obtained in the intermediate validation, which are around 5.6% on average. It can also be observed that for the considered training dataset the GA-optimized ANN structures consist of 2~3 hidden layers, with 2~3 neurons/layer. While these structures seem straightforward to try by hand, when compared to today's common approach of manual parameter setting with the TensorFlow-Keras Python library for a similar range of 3- through 5-layer densely connected ANNs with 1, 5, or 30 neurons/layer and the widely-used *relu activation function* for regressors, the **ann-5features** performs way better. Particularly, the best validation set predictions for the manual approach are obtained with the 4- and 5-layer ANNs with 30 neurons/layer in the hidden layers, resulting in an average MAPE of ~36% for the 5 folds. These results further motivate the adoption of GA-based ANN structure optimization.

For the MSTS-based models, each fold corresponds to a running 4-week window, with 3-week worth of training data and the remaining 1 week for the test. The R `forecast` package's `msts` and `HoltWinters` functions are used to model the multi-seasonality (i.e., daily and weekly) in the time series data. The resulting models are in turn used as input to its `forecast` function to predict the values for a given horizon.

As regards the LSTM-based models, the performance was noted to be unstable and highly dependent on the stochastic nature of the initialization and optimization, as well as the number of time steps used as input and the number of neurons in the LSTM layer, among others. Setting the number of input time steps to 12 and the neurons in the LSTM layer to 20 is the first configuration that gave workable results; more time steps and neurons resulted in divergence (i.e., undefined loss value) during training, such that the model was not learning. The training/validation/test MAPE values obtained for the given training and test sets range from ~4% through ~100%, and although each experiment is repeated 100 times, a stationary average performance may still not be guaranteed. Furthermore, 5-fold cross-validation as well as varying training and test batch sizes have been also explored but the initial tests did not result in any improvements; hence, they are not included in the paper as they only present unnecessary computing overheads.

The independent test MAPE values for the five models evaluated in this study – namely, **ann-5features**, **msts-1step**, **msts-24steps**, **lstm-1step** and **lstm-24steps**, are shown in Fig. 4, in which results obtained with the ANN regressor and the MSTS models are also broken down into the 5 folds, while those with LSTM are simply the average of the 100 independent runs. It can be observed that, on average, the ANN regressor has better accuracies than the MSTS- and LSTM-based models – even with the best-case scenario of 1-step ahead forecasting. Besides the ability of predicting any value in the future enabled by the selected features of the **ann-5features**, it has improved the prediction accuracies by an average of ~66%, ~51%, ~88% and ~83% with **msts-24steps**, **msts-1step**, **lstm-24steps** and **lstm-1step**, respectively. It is important to note that the best run among the 100 independent runs of **lstm-24steps** and **lstm-1step** had independent test MAPE values of ~13% and ~4%; however, due to LSTM's performance instability even for the same values of hyperparameters and the same dataset, the average of the MAPE values obtained in the repeated experiments are way higher.

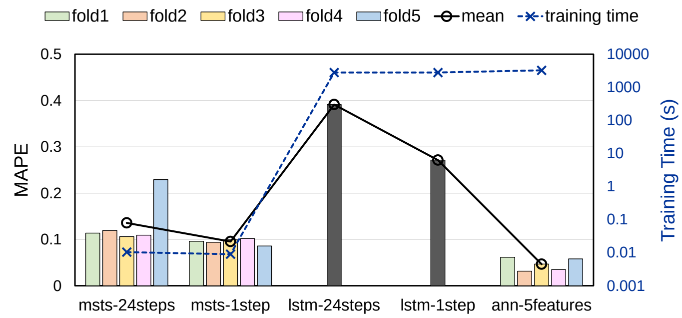

Fig. 4. MAPE values obtained in each fold of **msts-24steps**, **msts-1step** and **ann-5features** for the hourly demand prediction.

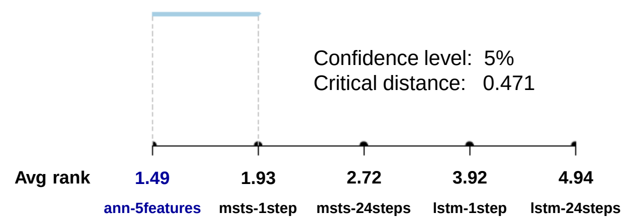

Fig. 5. Nemenyi test results for the models' independent test MAPE values.

Additionally, the Nemenyi test [15] in R has been used to further evaluate the similarities in performance of the five models, with the average MAPE value obtained for each sample in the models' 1-week worth of independent test sets being the input. It first computes for the *average rank* of the models, and then based on a *confidence level* parameter, the *critical distance* that defines the similarity among the average ranks is obtained. Fig. 5 shows the results of the Nemenyi test for the default confidence level of 5%, indicating a critical distance of 0.471. It is interesting to note that while the ranking of the models in both Fig. 4 and Fig. 5 coincide (i.e., **ann-5features** > **msts-1step** > **msts-24steps** > **lstm-1step** > **lstm-24steps**), the Nemenyi test finds both the **ann-5features** and **msts-1step** to have similar performances in terms of accuracy, since the difference between their average ranks is below the critical distance.

Fig. 4 also shows a comparison of the models' training times, where **msts-24steps** and **msts-1step** indicate durations of ~10ms, **lstm-24steps** and **lstm-1step** indicate ~47 minutes, while **ann-5features** indicate ~55 minutes. Since MSTS-based models are dependent on the temporal succession of the samples, fewer samples can be used to train the models for the given data in order to execute the 5 folds (i.e., 540 samples or 3-week worth of data, as previously anticipated); without cross validation, maximizing the number of training samples as in the ANN-based approaches results in durations of ~11ms. Moreover, online re-training is necessary as new samples are observed in order to keep valid models and maintain the temporal succession. This also entails high reliance on a dynamic management system's monitoring components. The time for receiving the new observation values, updating models and generating forecasts could be critical, especially in 5G and beyond environments. Therefore, further costs can be expected for MSTS-based models due to these concerns.

Without the 5-fold cross-validation, the training samples used for LSTM-based models can be maximized at 1284 samples or 7.6-week worth of data – just a little bit smaller than those used in the **ann-5features** because of the temporal succession dependencies. By removing such dependencies, a total of 1296 samples or 7.7-week worth of data can be used to train the **ann-5features**. Although LSTM-based models have training times lower by ~8 minutes compared to the latter, the limitations that come with re-training and performance instability pose critical concerns. Moreover, at least over 500 ANN structures have been evaluated in each fold of the **ann-5features** to obtain a generalized model, and the predictors' validity are also not constrained by the forecast horizon; re-training can also be done less frequently offline.

To summarize, an indirect benefit of the generic features used for the proposed ANN regressor is the lack of need for having the window of past series values at each time step, as typically needed in the other approaches. This enables the relaxation of monitoring requirements (such as time criticality for receiving these values) that a dynamic management system poses on the related monitoring components. On this note, monitoring requirements on 5G networks are already increased [2], therefore relaxing them on some parts of the system would be beneficial. This also signifies that the **ann-5features** is a good trade-off between accuracy, training time, as well as re-training and monitoring costs.

VI. Conclusion

The 5G scenario has called for AI-powered ZSM solutions to support next-generation applications and services. With this in mind, a novel input feature set and framework for transforming time series forecasting into a simpler multivariate regression problem using GA-optimized ANNs is proposed in the context of urban area network activity runtime prediction. Particularly, new input features are defined to capture the seasonality and calendar effects of the network activity time series, which enables the approach to remove the prediction accuracy's dependence on the temporal succession of input data and the forecast horizon.

Based on Telecom Italia's Milano Grid dataset, aggregated at the urban area level, evaluation results show that the GA-optimized **ann-5features** regressor has generally better prediction performance compared to the 1-day and 1-hour ahead forecasts obtained with the MSTS- and LSTM-based models by an average of ~59% and ~86%, respectively. Furthermore, despite its longer training times compared to the other models, the **ann-5features**' validity is not constrained by the forecast horizon and the temporal succession of data samples. Re-training can then be done less frequently offline, hence relaxing the monitoring requirements in the system.

References

[1] "Zero-touch network and Service Management (ZSM); Reference Architecture," ETSI GS ZSM 002 V1.1.1, Aug. 2019. Accessed on: Oct. 9, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/002/01.01.01_60/gs_ZSM002v010101p.pdf.

[2] "5G Systems: Enabling the Transformation of Industry and Society," Ericsson White Paper, UEN 284 23-3251 rev B, Jan. 2017. Accessed on: Apr. 9, 2020. [Online]. Available: https://www.ericsson.com/49daeb/assets/local/reports-papers/white-papers/wp-5g-systems.pdf.

[3] J. M. Vilar, R. Cao and G. Aneiros, "Forecasting Next-day Electricity Demand and Price Using Nonparametric Functional Methods," in *Int. J. Electr. Power Energy Syst.*, vol. 39, no. 1, 2012, pp. 48-55.

[4] G. Kousiouris *et al*., "Parametric Design and Performance Analysis of a Decoupled Service-Oriented Prediction Framework Based on Embedded Numerical Software," in *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 511-524, Oct.-Dec. 2013.

[5] G. Kousiouris, G. Vafiadis and T. Varvarigou, "A Front-end, Hadoop-based Data Management Service for Efficient Federated Clouds," in Proc. 3rd IEEE Int. Conf. Cloud Comput. Technol. Sci., Athens, Greece, 2011, pp. 511-516.

[6] U. Paul, J. Liu, S. Troia, O. Falowo and G. Maier, "Traffic-profile and Machine Learning based Regional Data Center Design and Operation for 5G Network," in *J. Commun. Netw.*, vol. 21, no. 6, pp. 569-583, Dec. 2019.

[7] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks," in Proc. 41st Int. ACM SIGIR Conf. Res. & Develop. Inf. Retrieval, Ann Arbor, MI, USA, 2018, pp. 95-104.

[8] T. Subramanya and R. Riggio, "Machine Learning-Driven Scaling and Placement of Virtual Network Functions at the Network Edges," in Proc. 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 2019, pp. 414-422.

[9] X. Zhang, T. Xue and H. Eugene Stanley, "Comparison of Econometric Models and Artificial Neural Networks Algorithms for the Prediction of Baltic Dry Index," in *IEEE Access*, vol. 7, pp. 1647-1657, 2019.

[10] C. G. Carreño Romano, N. Clivio and C. Righetti, "Sizing Techniques applied to Network Capacity Planning," in Proc. 2018 IEEE Biennial Congr. Argentina (ARGENCON), San Miguel de Tucumán, Argentina, 2018, pp. 1-8.

[11] J. J. Moré, "The Levenberg–Marquardt Algorithm: Implementation and Theory," in *Numerical Anal.*, vol. 630, G. A.Watson, Ed. Berlin, Heidelberg: Springer, 1978, pp. 105–116.

[12] Telecom Italia, "Telecommunications - SMS, Call, Internet - MI," Harvard Dataverse, May 2015. Accessed on: Aug. 1, 2018. [Online]. Available: https://doi.org/10.7910/DVN/EGZHFV.

[13] G. Barlacchi et al., "A Multi-source Dataset of Urban Life in the City of Milan and the Province of Trentino," in *Scientific Data,* vol. 2, Oct. 2015, Art. no. 150055.

[14] "Ericsson Mobility Visualizer," June 2020, Accessed on: Oct. 6, 2020. [Online]. Available: https://www.ericsson.com/en/mobility-report/mobility-visualizer.

[15] "Nonparametric multiple comparisons (Nemenyi test)," Accessed on: Feb. 4, 2021. [Online]. Available: https://rdrr.io/cran/tsutils/man/nemenyi.html.

Jane Frances Pajo received the B.Sc. degree (*cum laude*) in electronics and communications engineering from the Mindanao State University-IIT, Philippines, in 2010, while the M.Sc. (*cum laude*) and Ph.D. (*Europaeus*) degrees in Telecommunications Engineering were obtained from the University of Genoa, Italy, in 2015 and 2019, respectively. She received the IEEE-ABB PhD Thesis Award 2019 on New Challenges for Energy and Industry. She currently holds a post-doctoral research fellowship with the TNT Laboratory at the DITEN department of the University of Genoa. She also works with the CNIT S2N National Laboratory, Genoa. Her research interests include applications of network softwarization technologies and artificial intelligence for network/service management and orchestration.

George Kousiouris is an Assistant Professor at the Department of Informatics and Telematics of the Harokopio University of Athens, Greece. He received his Dipl. Eng. in Electrical and Computer Engineering from the University of Patras, Greece in 2005 and his Ph.D. in Cloud Computing at the Telecommunications Laboratory of the Dept. of Electrical and Computer Engineering of the National Technical University of Athens in 2012. He has participated in numerous EU funded projects such as H2020 BigDataStack, H2020 CloudPerfect (as lead architect and technical coordinator), H2020 SLALOM, FP7 COSMOS (as lead architect and technical coordinator), FP7 ARTIST, FP7 IRMOS, etc. His interests are mainly Cloud services evaluation and benchmarking, Service Level Agreements, IoT platforms, computational intelligence, performance and description modelling and service-oriented architectures.

Dimosthenis Kyriazis is an Assistant Professor at the Dept. of Digital Systems of the University of Piraeus, Greece. He received his diploma from the school of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in 2001 and his M.Sc. degree in "Techno-economics" in 2004. Since 2007, he holds a Ph.D. in the area of Service Oriented Architectures with a focus on quality aspects and workflow management. His expertise lies with service-based, distributed and heterogeneous systems, software and service engineering. Before joining University of Piraeus, he was a Senior Research Engineer at the Institute of Communication and Computer Systems (ICCS) of NTUA, having participated and coordinated several EU and National funded projects (e.g. ORBIT, LeanBigData, CoherentPaaS, ARTIST, COSMOS, VISION Cloud, IRMOS, etc.) focusing his research on issues related to quality of service provisioning, fault tolerance, workflow management, performance modelling, deployment and management of virtualized infrastructures and platforms.

Roberto Bruschi (M'09–SM'13) is Associate Professor of Telecommunication Networks at the DITEN department of the University of Genoa, Italy. Roberto has been the Project Coordinator of the RIA INPUT project, funded by the H2020 Programme. He took part in the activities of many national and European projects (e.g., MIUR GreenNet as Principal Investigator, H2020 MATILDA, H2020 SPIDER, FP7 IP ECONET, etc.). In the ECONET project, he served as Coordination Project Manager, and in 5G-PPP MATILDA was leading the R&D activities on the 5G Network Platform. Roberto has co-authored over 130 papers in international journals, book chapters and international conference proceedings, and he was the recipient of two Best Paper Awards (at IEEE ICC 2009 and at the 3rd IEEE Int. Workshop on Green Communications), one Runner-Up Best Paper (at IoT Word Forum 2015), and one Best Demo (at IFIP/IEEE IM 2017) Awards.

Franco Davoli (M'90–SM'99–LSM'19) is Professor Emeritus at the DITEN department of the University of Genoa, Italy. His current research interests are in dynamic resource allocation in multiservice networks and in the Future Internet, wireless mobile and satellite networks, multimedia communications and services, and in flexible, programmable and energy-efficient networking. He has co-authored over 380 scientific publications in international journals, book chapters and conference proceedings. In 2004 and 2011 he was Visiting Erskine Fellow at the University of Canterbury, Christchurch, New Zealand. He has been Principal Investigator in a large number of projects and has served in several positions in CNIT. He is currently the Head of the CNIT S2N National Laboratory, Genoa, and coordinator of the H2020 5G PPP 5G-INDUCE project.