Online Approach for Proactive Scaling of Mobile Core Network Functions

Abrahão Ferreira[1], Kauan Tavares[1], Douglas Vidal[1], Silvia Lins[3], Aldebaro Klautau[1], Cristiano Bonato[2], Glauco Gonçalves[1]

[1]Universidade Federal do Pará (UFPA), Brasil

[2]Universidade do Vale do Rio dos Sinos, Brasil

[3]Ericsson Research, Brasil

{abrahao.ferreira, kauan.tavares, douglas.vidal}@itec.ufpa.br

{aldebaro.klautau, glaucogoncalves}@itec.ufpa.br

cbboth@unisinos.br silvia.lins@ericsson.com

Abstract. *New use cases for 6G networks introduce stringent requirements, like massive connectivity, which demand effective scaling of core functions. This is often achieved through scaling techniques that utilize machine learning models to proactively predict traffic demand. However, these models face challenges concerning concept drifts, i.e., changes in the statistical patterns previously learned by the model. To address this limitation, this study evaluates a method based on online learning for scheduling core functions. The method was tested using data from a mobile network, simulating scenarios with concept drift. The results demonstrate that the online model effectively adapts to changes, reducing prediction errors and maintaining stability in dynamic environments.*

1. Introduction

As mobile networks evolve toward 6G, they face increasingly dynamic and demanding use cases. Unlike previous generations, which relied on fixed architectures based on predefined user requirements, 6G introduces native AI-supported intelligence to adapt to changing applications and user demands [Lu et al. 2023]. These changes require flexible provisioning of core functions, with dynamic adjustment of instances to ensure service continuity and efficient resource usage [Kuranage et al. 2023].

Strategies for scaling functions in the current 5G core include reactive and proactive methods. In the reactive method, new instances are created or removed according to current demand. However, delay between decision making and action can cause underprovisioning, service degradation, and oscillations, that is, instances that are reconstructed soon after being removed [Akshlley et al. 2022]. The proactive method, in turn, employs predictive models, usually based on machine learning, to anticipate traffic variations [Hoi et al. 2021].

A common practice in proactive methods is to capture data from the network to train the models in an offline way. However, offline models tend to lose accuracy over time due to concept drift — that is, changes in the statistical properties of traffic that occur as new applications and network technologies emerge [Manias et al. 2023]. In this way,

offline learning models trained for 5G networks may become less effective as the network evolves towards 6G. In contrast, online learning models train with living data and stand out for their ability to continuously adapt to changes in data through the incremental learning process [Pérez-Sánchez et al. 2018]. Due to this characteristic, online models have recently been used to solve problems in non-stationary scenarios. An example is the work of [da Silva et al. 2024], which obtained promising results when investigating the use of online learning for edge computing environments.

This work aims to evaluate the feasibility of using online learning techniques for scaling the number of replicas of the *Access and Mobility Management Function* (AMF) function in a 5G core. In this regard, the AMF was chosen due to its critical role in user and session management, making it highly sensitive to traffic fluctuations. Although this study focuses on AMF, the proposed method is general and can be extended to other core network functions, as many follow similar architectural and scaling principles. As a contribution, the work compares the quality of traffic predictions of online and offline learning models in non-stationary scenarios. These evaluations follow a method proposed in this work, using data from a telecommunications operator [Barlacchi et al. 2015].

The remainder of this paper is structured in four sections. Section 2 presents the main works related to 5G core scalability. Section 3 details the methodology adopted in this work, describing the data used and the predictive models employed. Finally, Section 4 presents and analyzes the results obtained, followed by the conclusions and future perspectives in Section 5.

2. Related Work

Among the reactive approaches for 5G core scalability, a reactive solution is presented that minimizes the number of instances based on service classes [Chouman et al. 2023]. Although it produces optimal solutions, the proposal may suffer from provisioning delay, as it only reacts to the observed demand. Furthermore, the dependence on parameters to be adjusted compromises the efficiency and scalability of the proposal.

On the other hand, the use of Deep Neural Networks and *Long Short-Term Memory* (LSTM) models to predict request demand was investigated [Alawe et al. 2018b]. The task is approached as a classification problem, subdividing traffic into load ranges, where each range is associated with the number of AMF instances required to support it. In their evaluations, the models outperformed the reactive method, reducing the number of rejected requests and the overload on the instances. However, the use of classification models limits the scope of prediction by the maximum load observed during training, making the models inadequate in scenarios of increasing traffic.

Gated Recurrent Unit (GRU) and Wavenet models have been proposed to support the scalability of AMF functions in Kubernetes environments to predict CPU usage [Kuranage et al. 2023, Passas et al. 2022]. The works simulate the behavior of AMF through a *web* server, with HTTP requests following known traffic patterns [Barlacchi et al. 2015]. However, in non-stationary scenarios such as those found in telecommunications, the lack of adaptability to the evolution of network behavior compromises the feasibility of these methods in the long term.

The scalability of AMF instances from the perspective of concept drift is addressed in previous work [Alawe et al. 2018a]. Using a predictive offline model, the

authors overcome this challenge by retraining the model after a drift occurs. Although they address the concept drift problem, the solution has limitations, such as the delay between concept drift detection and model adaptation by retraining and the need to store a large volume of data for retraining. Due to these limitations, the solution becomes poorly suited to highly dynamic scenarios.

The analyzed works show that the proactive approach is preferred due to its advantages over the reactive approach. However, the main limitation of the existing works is the use of prediction models with low capacity to continuously adapt to changes in the network without long retraining. In this sense, this work fills these gaps by presenting an online scalability model that guarantees the efficiency of predictions, even in constantly evolving scenarios, as it can update itself as new data becomes available.

3. Evaluation method

The main contribution of this work is the comparative evaluation of the performance of an online learning method against an offline learning method for the task of proactively sizing the AMF function in the presence of concept deviations. In this way, the evaluation method seeks to obtain, at the same time, a controlled environment, which allows the introduction of concept deviations, and a relevant one, in which the statistical properties of traffic are similar to those found in reality and in which high-level operational indicators can be captured.

The first stage of the evaluation, described in Section 3.1, involves processing traffic data from a real mobile network, ensuring that the probability distributions learned by the models are similar to those found in practice while at the same time, due to the structure of the data, enabling the introduction of concept biases. The next stage (Section 3.2) consists of selecting, training, and evaluating the offline and online learning models for the task of predicting traffic under concept biases.

3.1. Dataset and Processing

The assessment uses telecommunications activity data from Telecom Italia's mobile network in and around Milan [Barlacchi et al. 2015], resulting from the collection of *Call Detail Records* (CDRs)[1] from November 1, 2013 to January 1, 2014[2]. These records are known in the mobile network literature, having been used in similar works [Alawe et al. 2018a, DeAlmeida et al. 2021]. It is worth noting that although the data set used does not have recent data, it remains appropriate for this study, which focuses on evaluating the adaptability of online learning to traffic pattern changes.

The records are organized by dividing the study region into a square grid of 100 x 100 cells, each with an ID and a position `(x, y)`.[3] Records are obtained at 10-minute intervals, containing cell ID, timestamp of the start of the measurement, SMS activity, phone calls, and internet. For security and privacy reasons, the CDR data were resized by a factor known only to the provider, which preserves the distribution of real traffic.

[1]CDRs records of user activity over time for billing and network management purposes.

[2]Hosted at `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV`

[3]The location of each cell is available at `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QJWLFU`

The Activity data (SMS, calls, and Internet) were summed into Total Activity, as seen in [Alawe et al. 2018b]. Likewise, it is established that 10% of the Total Activity corresponds to the Control Activity, i.e., the total amount of control requests sent to the 5G core every 10 minutes.

In order to evaluate the impact of concept drift, the Control Activity of each cell was used to divide the dataset into groups (clusters) with different patterns. clusters allow simulating abrupt concept drift events since it is possible to train a model with data from one cluster and test it with data from another cluster with a different statistical pattern. Thus, similarly to the work of [DeAlmeida et al. 2021], the *K-means* clustering algorithm was applied to separate the cells into different clusters, according to their Control Activity. Using the same dataset, the authors evaluated different values of K, determining the optimal number of 5 clusters. Figure 1 shows the result of grouping the 10,000 cells into 5 clusters, on the left, and the Cumulative Function of Control Activity in each cluster (in requests per second), on the right.

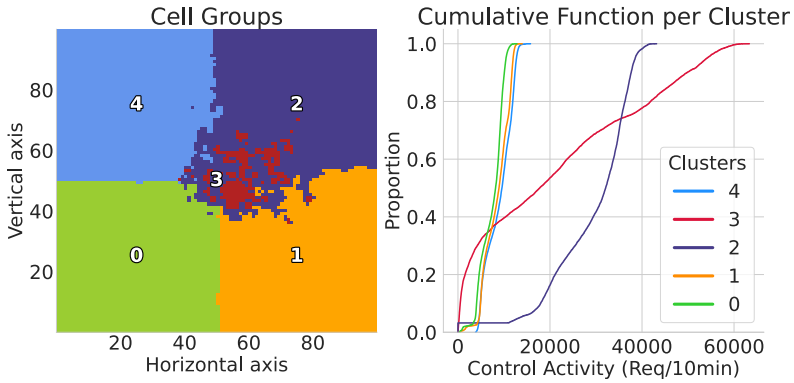

Figure 1. Left: Grid of cells subdivided into five clusters. Right: Cumulative function of Control Activity in each cluster.

In the grid of 10,000 cells, it is possible to observe that *Clusters* 2 and 3 cover mainly the central region of the city of Milan, and, therefore, their distributions present greater amplitude, being essentially different from each other and the others. *Clusters* 0, 1, and 4, in turn, present similar spatial coverage and relatively similar distribution, with a smaller amplitude. These results suggest relevant differences in the traffic pattern between clusters 2, 3 and the remaining three, while the latter present more remarkable similarities between themselves.

3.2. Control Activity Prediction Models

In order to compare the prediction of Control Activity, we used the LSTM model as a representative of the offline models. Commonly used for time series forecasting, LSTM was selected due to the results presented in [Alawe et al. 2018b], which, using the same data set, showed that the LSTM model was more accurate over other offline models.

For online learning, *Aggregated Mondrian Forests Regressor* (AMFR) was chosen because it outperformed other online regression methods in a similar scenario [da Silva et al. 2024]. AMFR is a supervised online learning model based on random forests that use the concept of Mondrian partitions to create adaptive decision trees

[Mourtada et al. 2021], which are updated as new data are received. The individual predictions of each tree are calculated as a weighted average of the predictions of its partitions, with a higher weight given to those that performed better. The final prediction is obtained by averaging the predictions of all the trees in the forest.

Each model is trained on the November's Control Activity traffic from one cluster and tested on the December's data from another cluster. Following this strategy, one can make 120 combinations of training/testing cluster pairs, but, due to space constraints, the results here are obtained by training the models on cluster 0 and testing them on all clusters (including cluster 0). Since cluster's 0 traffic distribution is similar to clusters 1 and 4 and distinct from clusters 2 and 3, we have three test cases with slight to no drift (0, 1, and 4), and two cases with severe drifts (2 and 3). The LSTM model was trained exclusively on the November data from cluster 0 and asked to predict the traffic of each cluster from December onwards. AMFR uses November data from cluster 0 as pre-training but continually adjusts to new data as it is observed.

In order to optimize the performance of each model, we performed hyperparameter selection using the November's Activity of cluster 0, through Bayesian Optimization [Bergstra et al. 2013]. The best LSTM model has an LSTM layer with 256 hidden units, followed by a dense layer with one output unit. The activation function used in the LSTM layer was the hyperbolic tangent, while the dense layer used linear activation. A learning rate of 0.00156 and a batch size of 256 were adopted. For AMFR, seventy estimators were selected, using aggregation in the output and a step of 6.3936. As for model inputs, AMFR uses only the most recent input to make predictions, while LSTM uses a window of 30 previous inputs to explore the autoregressive capacity better. The size of this window was defined based on an analysis of the autocorrelation found in the data.

The quality of the predictions is measured by the *Mean Absolute Error* (MAE), the *Root Mean Square Error* (RMSE), and the Residual Error (RE). The ER captures the difference between the real value of requests in the 10-minute interval and the value predicted by the model for the same interval (RE = $y_i - \hat{y}_i$) and makes it possible to analyze overestimation or underestimation; the MAE evaluates the magnitude of these deviations, providing a measure of the model's accuracy and is calculated as MAE $= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, where n is the total number of observations; RMSE is more sensitive to significant errors and is calculated by RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$.

4. Results

Table 1 presents the error metrics of the models for each cluster. It is possible to observe that AMFR was able to present better results for scenarios where concept drifts are less pronounced (clusters 0, 1, and 4). The LSTM model presents competitive results in these scenarios, confirming the suitability of offline models for stationary scenarios. Additionally, clusters 2 and 3 show how concept drift impacts the results of both algorithms, with the MAE of LSTM being 20% to 35% higher than those of AMFR; on the other hand, they also show how the online method can adapt, presenting significantly better metrics.

In order to show in detail how the distribution of the error of each method is impacted by the concept drift, Figure 2 shows the density curves of the residual error of each model in clusters 0 (no drift) and 3 (with drift). In cluster 0, both methods present similar

Table 1. Error measures of the AMFR and LSTM models for each cluster.

Metric	Models	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
MAE	**AMFR**	156.93	171.63	490.64	745.70	187.73
	LSTM	166.64	187.93	589.18	1009.91	190.04
RMSE	**AMFR**	308.24	307.74	1055.34	1430.31	315.13
	LSTM	361.48	339.15	1414.50	1573.37	310.55

error distributions, centered close to zero, but with LSTM having a significantly higher variance (134546.53 versus 95008.87 of AMFR), indicating that the offline model is more susceptible to extreme variations. In cluster 3, the concept drift makes the errors of both models larger, but, in this scenario, AMFR demonstrates superior performance, having a much higher error concentration at zero, whereas LSTM presents a marked left-skewed distribution (skewness -4.4), indicating a tendency towards underestimation, which could make undersizing decisions more frequent, leading to more rejected requests and increased AMF's response time. Considering that an AMF supports a maximum of 12000 requests per 10 minutes (from [Alawe et al. 2018b]), for example, an underestimate of 30000 requests per ten minutes would produce a deficit of about 3 AMF instances.

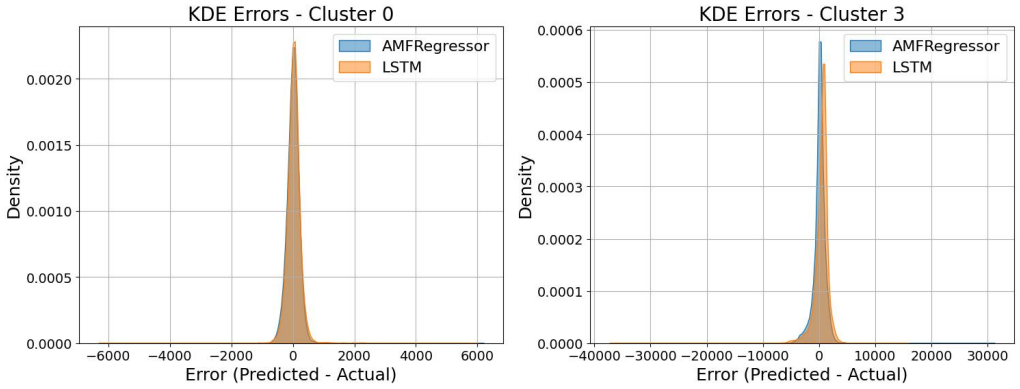

Figure 2. Density curves of the Residual Error of the models in clusters 0 and 3.

For better visualization of how the offline and online methods behave in the presence of concept drifts, the Figure 3 shows the average daily traffic and the forecasts for the whole test month for cluster 3. In general, one can see that the LSTM model (orange bars) tended to overestimate the actual activity (black bars), mainly at the end-of-year festivals, which are marked by reduced activity and can be seen as a new concept drift. Despite imposing a new challenge to the prediction methods, the AMFR model proved to be considerably more resilient to this seasonal effect, producing predictions much closer to the actual values and adapting more quickly to oscillations in network activity.

These characteristics can be better understood by comparing both predictions during a short oscillation. The inset graph on Figure 3 presents cluster 3's activity between December 21st 8:00 pm and 22nd 6:00 am. In this period, the activity (black crosses) softly reduces and suddenly drops to 0 in the last hours of December 21st, returning to grow after that. One can observe that in the drop period the LSTM model predictions (orange line) oscillate between overestimates and underestimates during four hours. In contrast, AMFR follows closely the actual activity, without prominent deviations.

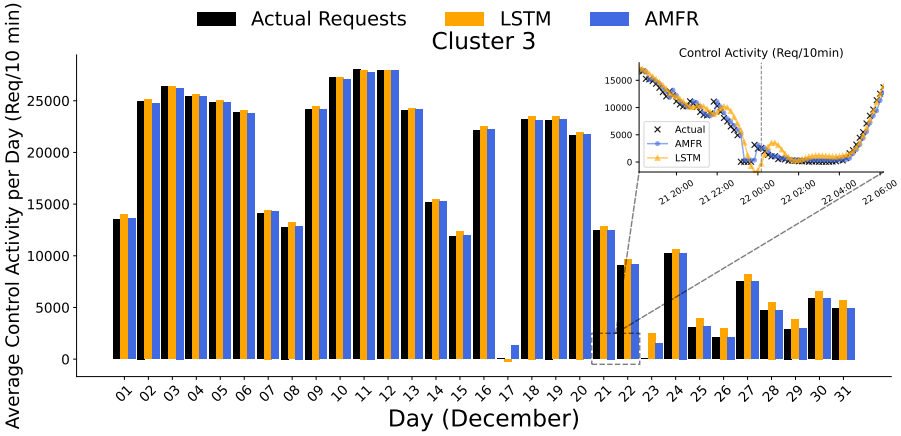

Figure 3. Average daily traffic forecasts for cluster 3 in December, emphasizing (inset graph) the nighttime fluctuations of lower activity for December 22.

5. Conclusion

This work evaluated the feasibility of employing online learning techniques for the proactive scalability of 5G core network functions in the presence of concept drifts. An analysis based on the different traffic patterns was conducted to evaluate the behavior of an online prediction model against an offline model in the face of these pattern changes.

The results show that the online approach stands out for its adaptability to concept drifts and presented fair results in low-variation scenarios. This analysis highlights the importance of adaptive approaches, such as AMFR, to deal with the variability and complexity of non-stationary environments, such as future 6G networks, since a model that offers inadequate predictions can compromise the experience of users, delaying or rejecting service requests, and, in more serious cases, make the mobile network unfeasible. Thus, this work shows that online models guarantee consistent predictions while avoiding the need for retraining, so familiar to offline models.

In future work, we plan to investigate other online learning models and experiment with scalability solutions in a simulation environment to evaluate their impacts on function provisioning.

Acknowledgements

This work was partly supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil, the National Council for Scientific and Technological Development, and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

Akshlley, K., Carvalho, M., and Lopes, R. (2022). Análise de desempenho de estratégias de autoscaling vertical e horizontal: um estudo de caso com o kubernetes. In *Anais Estendidos do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 201–208. SBC.

Alawe, I., Hadjadj-Aoul, Y., Ksentinit, A., Bertin, P., Viho, C., and Darche, D. (2018a). An efficient and lightweight load forecasting for proactive scaling in 5G mobile networks. In *IEEE Conference on Standards for Communications and Networking*.

Alawe, I., Ksentini, A., Hadjadj-Aoul, Y., and Bertin, P. (2018b). Improving traffic forecasting for 5G core network scalability: A machine learning approach. *IEEE Network*, 32(6):42–49.

Barlacchi, G., De Nadai, M., Larcher, R., Casella, A., Chitic, C., Torrisi, G., Antonelli, F., Vespignani, A., Pentland, A., and Lepri, B. (2015). A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2(1):1–15.

Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, number 1, pages 115–123, Atlanta, Georgia, USA. PMLR.

Chouman, A., Manias, D. M., and Shami, A. (2023). A reliable AMF scaling and load balancing framework for 5G core networks. In *2023 International Wireless Communications and Mobile Computing (IWCMC)*, pages 252–257.

da Silva, T., Batista, T., Delicato, F., and Pires, P. (2024). An online ensemble method for auto-scaling NFV-based applications in the edge. *Cluster Computing*, pages 1–25.

DeAlmeida, J. M., Pontes, C. F. T., DaSilva, L. A., Both, C. B., Gondim, J. J. C., Ralha, C. G., and Marotta, M. A. (2021). Abnormal behavior detection based on traffic pattern categorization in mobile networks. *IEEE Transactions on Network and Service Management*, 18(4):4213–4224.

Hoi, S. C., Sahoo, D., Lu, J., and Zhao, P. (2021). Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289.

Kuranage, M. P., Hanser, E., Nuaymi, L., Bouabdallah, A., Bertin, P., and Al-Dulaimi, A. (2023). Ai-assisted proactive scaling solution for CNFs deployed in kubernetes. In *2023 IEEE 12th International Conference on Cloud Networking*, pages 265–273.

Lu, L., Liu, C., Zhang, C., Hu, Z., Lin, S., Liu, Z., Zhang, M., Liu, X., and Chen, J. (2023). Architecture for self-evolution of 6g core network based on intelligent decision making. *Electronics*, 12(15).

Manias, D. M., Chouman, A., and Shami, A. (2023). Model drift in dynamic networks. *IEEE Communications Magazine*, 61(10):78–84.

Mourtada, J., Gaïffas, S., and Scornet, E. (2021). AMF: Aggregated mondrian forests for online learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3):505–533.

Passas, V., Makris, N., Wang, Y., Apostolaras, A., Mpatziakas, A., Drosou, A., Korakis, T., and Tzovaras, D. (2022). Artificial intelligence for network function autoscaling in a cloud-native 5G network. *Computers and Electrical Engineering*, 103:108327.

Pérez-Sánchez, B., Fontenla-Romero, O., and Guijarro-Berdiñas, B. (2018). A review of adaptive online learning for artificial neural networks. *Artificial Intelligence Review*, 49:281–299.