Analysis of ML Algorithms to Support Elastic Service Chaining in eHealth Vertical Applications

Sandino Jardim (Corresponding author)*†, Felipe S. Dantas Silva†‡, Augusto Neto†, Harold Bustos§
Roger Immich†, Ramon Fontes†
*Federal University of Mato Grosso (UFMT), Barra do Garças - MT, Brazil
†Federal University of Rio Grande do Norte (UFRN), Natal - RN, Brazil
‡LaTARC Research Lab (IFRN), Natal - RN, Brazil
§State University of Rio Grande do Norte (UERN), Mossoró - RN, Brazil
sandino@ufmt.br, felipe.dantas@ifrn.edu.br, augusto@dimap.ufrn.br, haroldivan@uern.br
roger@imd.ufrn.br, ramon.fontes@imd.ufrn.br.

***Abstract*—The efficient design of SFC-enabled eHealth applications requires an accurate provision of the underlying infrastructure. This provision requires both computing and networking resources to meet stringent QoS requirements under any conditions of service demand. Cloud providers often offer automatic elasticity strategies based on monitoring specific metrics that lead to a waste of resources, time/energy consumption, and the problem of starvation with competing services. Our findings provide evidence that proactive-based elasticity overcomes these issues, when assisted by Machine Learning (ML) methods for predicting Internet traffic load. An optimal autoscaling algorithm depends on high precision and fast predictions to provide accurate results. Thus, this paper assesses ML algorithms to support SFC-enabled eHealth vertical applications. The experimental results suggest that the evaluated models achieved similar accuracy metrics, with an MLP architecture delivering the best performance in terms of time training and average prediction time.**

***Index Terms*—Service Chaining, eHealth, Machine Learning, proactive autoscaling**

I. Introduction

The healthcare landscape has changed significantly with the emergence of an electronic Health (eHealth) computing paradigm. In eHealth, non-digital healthcare platforms harness Internet-based electronic systems driven by the crucial support of the Internet of Things (IoT), Cloud/Edge Computing, Big Data, and other enabling technologies [1]. In such an ecosystem, many eHealth applications empower medical teams in different ways, ranging from real-time biometrics to intelligent data processing. This can result in precise diagnostics and therapeutic conduct, along with advanced decisions on treatment/supervision of patients. This is made possible through the integration of Artificial Intelligence (AI) techniques. In AI-enabled eHealth applications, intelligent systems can track eHealth activities, analyze and learn health data in real-time, and predict the future occurrence of diseases by allowing the early adoption of new treatments. However, the mission-critical eHealth services impose several stringent computing (processing, memory, and storage) and networking (bandwidth, limited delay/loss, and others) requirements that must be met by the infrastructure at runtime.

Infrastructures that adopt Service Function Chaining (SFC) [2] provide a means of tackling the aforementioned stringent requirements. In a general way, SFC allows the desired integration between the different service-forming components to be achieved, by enabling chains of services to be provided that combine virtualized functions and legacy systems. Together, all these actors operate in an integrated way to provide the desired eHealth service application. However, the efficient design of SFC-supported eHealth applications requires the accurate provision of the underlying infrastructure of both computing and networking resources. Resource provisioning is of paramount importance to meet the requirements of eHealth applications during the entire lifecycle to avoid performance degradation, especially in critical procedures (e.g., telesurgery or even during long/short-term interruptions).

Regardless of the architecture adopted, cloud-like services are continually being assessed in light of Quality of Service (QoS) requirements. These requirements are driven by different Key Performance Indicators (KPIs), which are established by Service Level Agreements (SLA) (e.g., efficiency, availability, and reliability) [3]. In the context of the eHealth vertical application, it is essential to provide services to patients, doctors, and researchers that meet strict QoS KPIs under any conditions of service demand [4]. For this reason, elastic management of virtualized infrastructures is a key factor, since it can automatically adapt (autoscaling) current patterns of energy to meet new changes in service demands.

Cloud providers offer reactive elasticity techniques that monitor resource utilization data and service performance KPIs to trigger autoscaling to reach minimum or maximum thresholds. The resource reservation system usually follows a staggered approach, which means scaling-up current values two-fold, to meet the new projected demand. If the first cycle is not enough, the autoscaling process repeats until it reaches the required amount [5]. This leads to an unassertive resource computing-staggered distribution scheme, where the response time of the reactive algorithm increases exponentially when applied to an infrastructure that approaches resource-depletion. In the eHealth scenario, this reactive stagger-based autoscaling is unsuitable, since the same service's successive autoscaling cycles are time/energy-consuming, can jeopardize QoS and impair user satisfaction, as well as resulting in starvation [6].

An optimal autoscaling algorithm depends on the ability to compute, in the first processing cycle, the final amount of resources that need to be scaled-up. The level of effectiveness and performance associated with the calculation task depends on the assertiveness of the autoscaling algorithm [7]. Such high-precision means the ability to meet projected service resource demands. A proactive-based elasticity approach paves the way to overcome the issues of reactive autoscaling approaches. One of the main goals of this work is to exploit the ability to predict future Internet traffic loads, so that accomplishing anticipated elasticity functions, and enable assertive autoscaling taking the predicted patterns as input. The time series analysis stands as the most traditional prediction method, since it can detect variation patterns of repetitive service demands.

Our findings from the bibliographical research analysis provide evidence of the prominence of Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN) algorithms for Internet traffic prediction in different use cases. Our hypothesis is that feed-forward artificial neural network architectures (e.g., MLP) will be feasible to meet the goals of SFC-enabled eHealth. The reason is on its ability to provide satisfactory outcomes in predicting traffic associated with an optimized training time caused by low computational complexity [8]. Thus, we expect that this promising precision and agility can be used to face the challenges of this research in tackling assertive and proactive autoscaling for SFC-based eHealth scenarios.

To the best of our knowledge, the literature lacks evidence ML-based solutions that support proactive autoscaling decisions in SFC-based eHealth services and use cases. In light of this, this paper seeks to fill the gaps referred to above by making the following research contributions: (i) providing a comprehensive analysis of the most appropriate algorithms tailored to SFC-based eHealth mission-critical services and use cases; and (ii) evaluating the performance of prominent ML-Based techniques for predicting time-series data to support elastic SFC-enabled eHealth services.

This paper is structured as follows. Section II outlines the most significant works in ML-based resource prediction. Section III describes the neural network architectures utilized in our evaluations. Section IV describes the methodology and setup of the experiments. Section V examines the results and discusses of the experiments. Section VI summarizes the findings of our research and provides some recommendations for future research.

II. Related Work

In cloud/edge-native infrastructures, the support of the resource elasticity capability of the virtualization paradigm is a key factor in provisioning both the computing and networking parameters of targeting containers and virtual machine instances in an automatic fraction. Our state-of-the-art analysis found that the currently adopted elasticity solutions design mostly follow a reactive decision-making approach to meet the optimal rate. For instance, the notable Kubernetes[1] tool can be exploited for autoscaling active containers and virtual machines. The Kubernetes approach proceeds by reconfiguring the running parameters (e.g., CPU cycles or incoming bandwidth) by doubling current patterns. If it is not enough, Kubernetes repeats the same workload until it reaches the desired demand, if it is possible.

Our previous work, designated as elaSticity in cLOudneTwork Slices (SLOTS) [5], introduces a statistically-based autoscaling solution to overcome the complexity of the computing approach that Kubernetes imposes. Although the experimental evaluation of SLOTS reveals impressive improvements over Kubernetes, it also deploys a reactive approach, which means it will be possible to detect a virtual application faces quality degradation caused by resource depletion.

There are several different schemes for Internet traffic prediction in the literature. However, the approaches can be grouped in two distinct ways: (i) statistically-based methods and (ii) machine learning-aided schemes. The integrated autoregressive moving average (ARIMA) is the most common statistical method. It can capture short-range dependencies, but not long-range ones, which means it has a poor performance when used for this purpose.

Some variations of these statistical methods, such as the Fractional ARIMA (FARIMA), can enable both dependencies to be described. They can also provide similar results to the artificial neural network schemes [9]. However, they are characterized by being subject to severe limitations in estimating the fractional differentiation parameter d, which is responsible for establishing the degree of differentiation necessary to form a second-order stationary time series.

Models of Artificial Neural Networks (ANN) have been widely used to design machine learning-based schemes and also specifically for Internet traffic prediction [9, 10]. For example, [8] compared the performance of four different ANN architectures (i.e., Sparse Autoencoder – SAE, RNN, and two variations of the MLP) for predicting network traffic. The authors suggested that MLP and RNN are the most appropriate methods for predicting network traffic because of their rapid data training capabilities.

LSTM is widely adopted as alternative to simple recurrent units to form a holistic recurrent neural network and learn complicated information within sequential data. In [11], an LSTM neural network model is designed to predict network traffic that shows non-linear trends and contains uncertain random factors, which may prevent the flow model with linear characteristics from being predicted accurately. Authors of [12] evaluated the performance of various RNNs, including a stacked LSTM within real-world data, to identify the optimal network parameters and network structure to achieve optimized predictions.

The use of CNNs to forecast short-term changes in the amount of traffic crossing a data center network is put forward in [13]. The proposed scheme outperformed ARIMA by an

[1] http://kubernetes.io/

increasingly significant margin as the forecasting granularity is above the 16-second resolution. In [14], the authors predicted network throughput to improve the adaptive streaming of the algorithm's performance using a dataset from which the network throughput could be extracted over different timescales.

Despite this range of works that employ ML-based Internet prediction, the literature lacks evidences about works that devote analyzing ML algorithms to support proactive autoscaling decisions in SFC-based eHealth use cases. Therefore, we provide a comprehensive analysis of the most appropriate algorithms tailored to this scenario. Aside from that, in this paper we study the performance of the main ML techniques founded in the literature to predict time series (e.g., MLP, LSTM, and CNN). The study's central premise lay in the characteristic traffic data that SFC-enabled eHealth use case yield, seeking to support its chained functions' proactive elasticity. We describe these architectures in more detail in Section III.

III. Predicting Models

We propose the use of four models derived from MLP, LSTM and CNN architectures to carry out demand prediction based on Artificial Neural Networks (ANN). ANNs are processing systems separated into strongly connected nodes known as artificial neurons. Each neuron has a synaptic weight, and is responsible for storing acquired knowledge. They are arranged into layers and capable of working in parallel to process and store data and knowledge and infer new data through learning processes. The learning process occurs during the network training, where the synaptic weights are modified until they reach the desired level of learning.

A. Multilayer Perceptron – MLP

Multilayer Perceptrons (MLP) are feed-forward artificial neural networks in which all the neurons in the same layer are connected to all the neurons of the next layer, but not in the same layer. The training algorithm used for MLP is backpropagation, which is a supervised learning algorithm, where the MLP learns the desired output from various data entries. However, backpropagation suffers the problem of the magnitude of a partial derivative, which makes it either too large or too small. This causes many fluctuations in the learning process by slowing the convergence time or making the network stuck in its local minimum. To avoid this problem, we will use Rprop, which has a dynamic learning rate, and updates the learning rate of every neural connection, by reducing the error for each neuron separately.

B. Long Short-Term Memory – LSTM

A Long Short-Term Memory (LSTM) neural network is a variant of a Recurrent Neural Network (RNN). RNN employs a method different from the traditional feed-forward neural network by introducing the recurrent structure in the network. It also establishes the neural network's connection to itself, where neurons store information from the previous period in the neural network and influence the current stage and output.

We also employed another architecture that combines two LSTM layers which will be referred to as Stacked. Multiple hidden LSTM layers can be stacked on top of another, and this is referred to as a stacked LSTM model. As an LSTM layer requires a three-dimensional input and LSTMs by default produce a two-dimensional output, we addressed this problem by having the LSTM output a value for each time step in the input data and the hidden output state for each input time step. This allows us to have 3D output from the hidden LSTM layer as input for the next layer.

C. CNN-LSTM Model

We also employ a Convolutional Neural Network (CNN) combined with a LSTM layer as in [15]. CNN can provide models for data with a meaningful topology efficiently, which are widely used for image recognition, but can be adapted for time series prediction. CNN is a specialized type of ANN featuring convolutional layers. These types of layers use convolutional filters, which are linear functions applied to the input data in a sliding-window fashion.

In our implementation, the CNN-LSTM architecture consists of one convolutional layer of 64 filters, followed by a pooling layer, an LSTM layer with 50 neurons and an output layer of one neuron. A flattening layer is used between the pooling layer and the LSTM layer to reduce the feature maps to a single one-dimensional vector. The CNN does not view the data as having time steps. Instead, it is treated as a sequence over which convolutional operations for reading can be performed, such as a one-dimensional image.

IV. Methodology

This work assumes that the SFC-enabled eHealth service will be subject to workloads that must go through the entire chain to be served [16] at random. However, they allow variations to be observed on demand throughout the day form the perspective of time windows of different sizes.

This means that, a dataset recommended for this context must start from individual requests that occur at any time. Different time windows can be derived from them that enable proactive decisions to be made for different time scales, from the representative statistical value for each window as input, to the decision of the mechanism of elasticity. In light of this, the evaluation here will be guided by a dataset of the lowest granularity that allows different time windows to be derived (as performed by [14]).

The traffic data has been extracted from point of presence of an Internet Service Provider (ISP) from Italy [17], during the period of 57 months. The dataset is available for download at the research dissemination page[2]. In particular, the dataset consists of HTTP requests from the provider's customers in a specific category of web pages. Thus, we believe that the variation in the number of requests captured, will represent the variation in the traffic demand for the eHealth use case of a chained patient data consultation service.

[2]https://mplanestore.polito.it:5001/sharing/b73FXl4KC

A. Data preparation

The raw data of the chosen dataset are originally arranged in the form of timestamps that contain the information of the day and time when each HTTP request has been made . In an attempt to transform the data series into a time series that represents variation in demand over time, we extracted the number of requests per second from the original data.

Once in possession of this time series, we divided it into 5 minute windows in order to configure the dataset in a format in which the prediction model could take action by carrying out the demand forecasting for the next five minutes. For this reason, as a statistical value to describe each window interval, we chose the average interval (mid-range) between the minimum and maximum per second requests that were made within the time interval. Figure 1 illustrates a time interval that corresponds to four days, as already configured in five-minute windows.

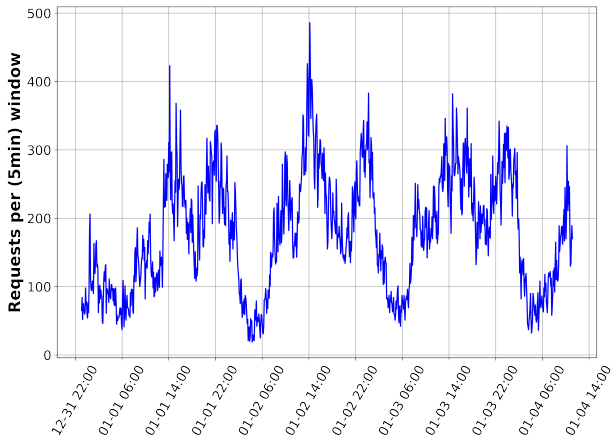

Fig. 1. Number of requests per window, where the number of requests is given by the midrange value in each window

B. Evaluating Performance of Models

To quantitatively evaluate the predictions of our models, we used the following metrics of the Mean Absolute Error (MAE), the Mean Squared Error (MSE), and the Median Absolute Error (MedAE). MAE, MSE and MedAE are respectively defined by the equations (1), (2) and (3):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y_1} - y_i| \tag{1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y_1} - y_i)^2 \tag{2}$$

$$MedAE(y, \hat{y}) = median(|y - \hat{y_1}|, ..., |y - \hat{y_n}|) \tag{3}$$

where n is the number of predictions, while $\hat{y}$ and y mean the amount predicted, together with its actual corresponding value. These are consolidated metrics utilized in forecasting problems which are used to measure a) the average of the forecast error values (Eq. 1), b) the average of the squared forecast error values (Eq. 2), which has the effect of putting more weight on large errors and c) the median of the absolute errors (Eq. 3), which is a measure of statistical dispersion that is more resilient to outliers than the standard deviation.

Additionally, we measured the performance of each forecasting model in terms of time for training and average prediction time. Both metrics combined with the performance prediction metrics are aimed at helping to choose the model that is most useful for proactive autoscaling, i.e., which offers faster and more accurate predictions.

C. Experimental Setup

The neural networks were implemented using the Python Keras[3] library, together with TensorFlow[4]. All the experiments were performed on a VM set with 64GB RAM, a 20-core Intel Xeon E5 2.0 GHz CPU, and running Ubuntu 18.04 LTS.

The quadratic mean error was used as the loss function that was employed during the training phase, which is needed to estimate model loss and optimize the iteration process. The number of training epochs was set to 10 as the results showed no improvement for higher values. The dataset consisted of 208.437 samples and normalized. Half of the dataset was used for training and the other half for validation. During the validation phase, when the model was already trained, we used three input samples at a time to compare the predictor output with the expected output and thus evaluate the accuracy of the models. Figure 2 depicts the obtained LSTM prediction for a period of one week.

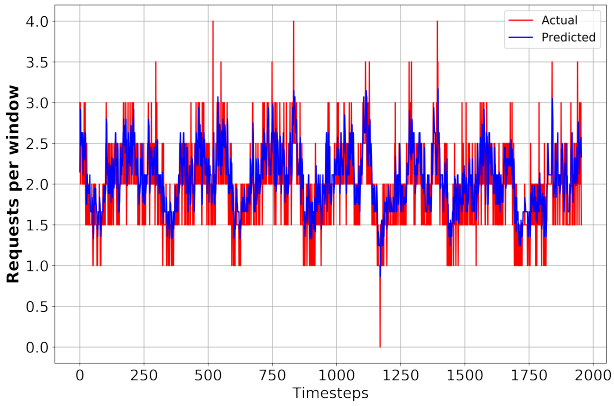

Fig. 2. LSTM prediction for a one week period

V. Results and Discussion

The results of the four models examined in this work were very close to each other, as illustrated in Figures 3, 4, and 5. In the case of MAE, the results were around 3.5×10^{-1}, with 3×10^{-1} for MedAE, and 2×10^{-1} for MSE.

The neural network architecture and model parameters, as the number of neurons and epochs were set empirically, the chosen metrics were unable to distinguish the model from the others in terms of accuracy, without displaying a significant

[3]https://keras.io

[4]https://www.tensorflow.org/

difference of values in the prediction of the same dataset employed. Nevertheless, the results for any of the evaluated models can be considered satisfactory both in terms of the average error from the actual measured value as in terms of their ability to reflect the demand variations. Thus, it indicates that an eHealth autoscaling mechanism supplied by one of these models will be able to adapt its resources with a five-minute anticipation (as the time window utilized in the experiments) and according to predicted values with a mean absolute error close to zero when handling with traffic with the same features.

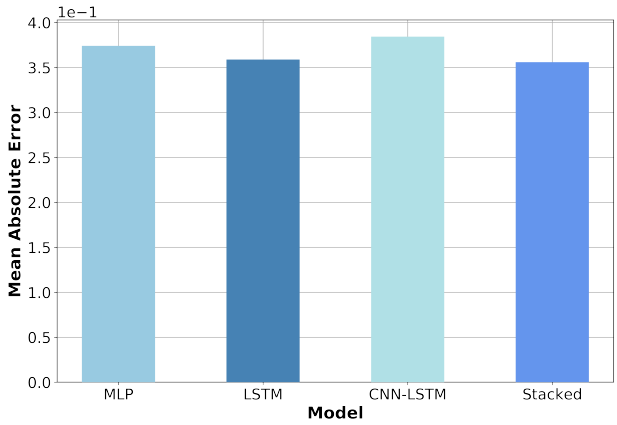

Fig. 3. Mean Absolute Error (MAE)

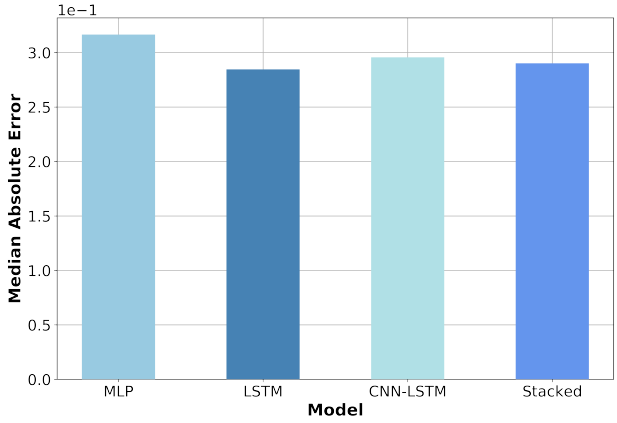

Fig. 4. Median Absolute Error (MedAE)

Figure 6 illustrates the results obtained from the training time that each algorithm took in the same dataset. The results showed the MLP model with training time approximately four times faster than the LSTM and CNN models, and approximately seven times faster than the *Stacked LSTM* model. This can be explained by the greater simplicity of the MLP architecture compared with the others, with one layer less than CNN and *Stacked*, and with more simplified neurons than LSTM units.

Once again, our empirical parameterization of setup of the architectures does not ensure that one model has an advantage over another in therms of accuracy. However, the results confirm the expectations about the MLP model and its time efficiency and establish it as a potential model for the autoscaling of eHealth applications. This is because the automatic decisions will be made more quickly as it takes less time to start making predictions owing to the fact that the training phase is faster.

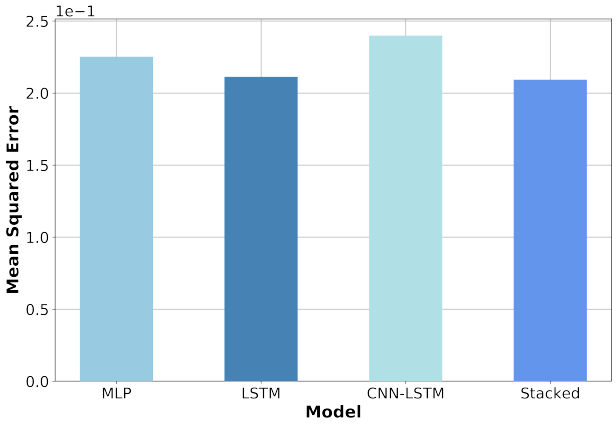

Fig. 5. Mean Squared Error (MSE)

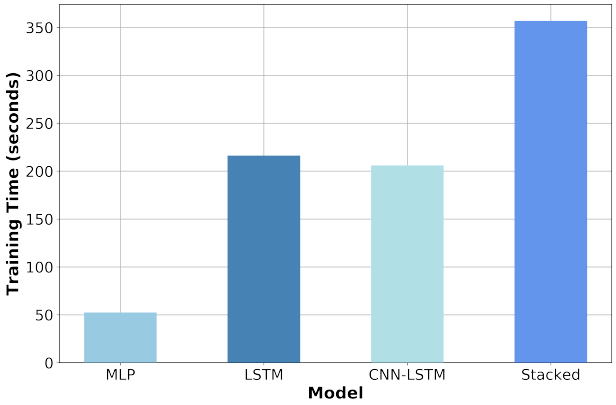

Fig. 6. Training time

Figure 7 illustrates the obtained results with regard to the time needed to obtain a new prediction value from the last three captured demand values. We intend to measure the speed with which the model can obtain new predictions in order to determine its impact on a eHealth real-time system. The results showed that once trained, the two-layer models and the LSTM model achieved results of approximately 4×10^{-3} seconds and the MLP again had an advantage, with a value below 1×10^{-3} seconds. Although all the models presented an average prediction time in millisecond fraction of seconds, MLP again emerged as a potential predictor for the eHealth domain, where it could lead to even more anticipated autoscaling decisions.

VI. Conclusion

In this paper, we evaluated ML algorithms aimed at supporting the elasticity of eHealth applications offered through service function chaining, proactively. Four different neural network models (MLP, LSTM, Stacked LSTM and CNN-LSTM) were utilized with a dataset representative of an SFC-based eHealth workload, which require to be resized

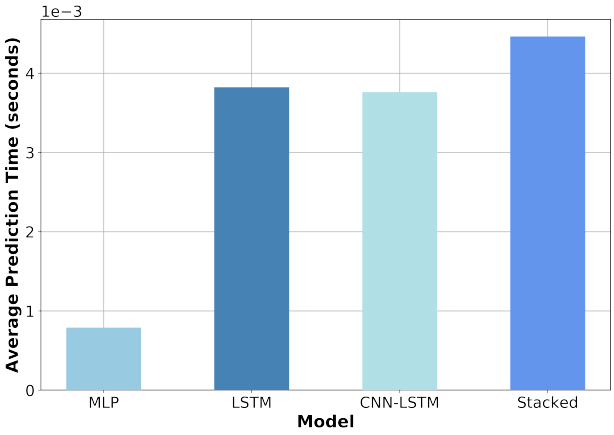

Fig. 7. Prediction time

in accordance with predicted demands. For this reason, the accuracy of the predictor is of great importance to drive proactive autoscaling decisions.

The four models had accuracy metrics that were close to each other. This enables the decision to be made about the best model as being the one that requires the least time for training and obtaining prediction results, and which is most suitable to SFC-based eHealth real-time systems. The MLP architecture achieved the best results for time training and average prediction time (time to obtain a prediction with the model already trained). This can be attributed to its simpler architecture which consists of more simplified neurons, and it achieved satisfactory results compared with the other models. The results can be improved for each of the models in future work through the application of optimization algorithms based on heuristics that would allow an optimal adjustment of the parameters of each one. Another recommended future work should concentrate on a couple of these models on a proactive autoscaling mechanism to evaluate the assertiveness of decisions in terms of the resources reserved to meet the demands of a chained eHealth service.

Acknowledgment

This research was partially supported by the H2020 4th EU-BR Collaborative Call, under the grant agreement no. 777067 (NECOS – Novel Enablers for Cloud Slicing), funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC. Additional thanks to the REGINA-Lab (UFRN) and the LORDI Lab (UERN) for the supporting technology.

References

[1] A. Neto et al. Predicting epileptic seizures: Case studies harnessing machine learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[2] J. Halpern and C. Pignataro. Service Function Chaining (SFC) Architecture. RFC 7665, October 2015.

[3] R. Pasquini et al. Inferring cloud-network slice's requirements from non-structured service description. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5. IEEE, 2020.

[4] S. Mbengue et al. Internet of medical things: Remote diagnosis and monitoring application for diabetics. In *International Wireless Communications and Mobile Computing*, pages 583–588, 2020. doi: 10.1109/IWCMC48107.2020.9148130.

[5] A. Medeiros et al. Enabling elasticity control functions for cloud-network slice-defined domains. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE, 2020.

[6] A. Medeiros et al. End-to-end elasticity control of cloud-network slices. *Internet Technology Letters*, 2(4):e106, 2019.

[7] H. Nikolas et al. Elasticity in cloud computing: What it is, and what it is not. In *10th International Conference on Autonomic Computing*, pages 23–27, San Jose, CA, June 2013. ISBN 978-1-931971-02-7.

[8] T. Prado Oliveira et al. Computer network traffic prediction: a comparison between traditional and deep learning neural networks. *International Journal of Big Data Intelligence*, 3(1):28–37, 2016.

[9] K. Christos and D. Sophia. Comparing forecasting approaches for internet traffic. *Expert Systems with Applications*, 42(21):8172–8183, 2015.

[10] Paulo Cortez et al. Multi-scale Internet traffic forecasting using neural networks and time series methods. *Expert Systems*, dec 2010. ISSN 02664720.

[11] S. Wang et al. A network traffic prediction method based on lstm. *ZTE Communications*, 17(2):19–25, 2019.

[12] R. Vinayakumar et al. Applying deep learning approaches for network traffic prediction. In *International Conference on Advances in Computing, Communications and Informatics*, pages 2353–2358. IEEE, 2017.

[13] M. Alberto et al. Forecasting short-term data center network traffic load with convolutional neural networks. *PloS one*, 13(2), 2018.

[14] B. Arkadiusz. Traffic prediction methods for quality improvement of adaptive video. *Multimedia Systems*, 24 (5):531–547, 2018.

[15] I. Livieris E et al. A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, 32(23):17351–17360, 2020.

[16] L. Ting-Mei et al. An e-healthcare sensor network load-balancing scheme using sdn-sfc. In *IEEE 19th International Conference on e-Health Networking, Applications and Services*, pages 1–4. IEEE, 2017.

[17] A. Morichetta et al. Characterizing web pornography consumption from passive measurements. In *International Conference on Passive and Active Network Measurement*, pages 304–316. Springer, 2019.