DeepRANKPI: Time Series KPIs Prediction in a Live Cellular Network with RNN

Mohammad Rasoul Tanhatalab
Machine learning researcher
Asigmo Data Science
Vienna, Austria
mohammad.tanhatalab@asigmo.com

Fatemeh Afghah
School of Informatics, Computing and Cyber Systems
Northern Arizona University
Flagstaff, AZ USA
fatemeh.afghah@nau.edu

Ehsan Vazifeh
Independent researcher
Vienna, Austria
em.vazifeh@gmail.com

*Abstract***— Predictive analytics is employed by telecommunications operators to gain valuable insights into the network performance and make data-driven decisions in order to optimize the quality of service for the users while saving the network resources. However, predicting the network key performance indicators (KPIs) is a challenging task in large scale cellular networks with complex spatio-temporal variations. Moreover, to optimize, expand, and modify the strategy of mobile networks, many parameters are changed or features are being implemented every day. One of the major challenges in network maintenance is to track the side-effects of such changes, in order to avoid any anomalies, by calculating the amount of degradation or improvement. Sometimes these changes coincide with other network seasonality behavior, hence, adding complications to the network. Inspired by the promising performance of recurrent neural network (RNN) in time-series prediction, we developed a KPI prediction model using an RNN algorithm in a mobile network with about 280000 cells of various technologies and frequencies. Model training is performed on three years of historic mobile network data. In this work, we compare the performance of several deep learning-based approaches to predict the network's KPIs in the future intervals.**

Keywords—Recurrent neural network, deep learning, KPI prediction, cellular networks, timeseries prediction.

I. INTRODUCTION

Artificial intelligence (AI)-based approaches have been being applied to the 5G and beyond. Especially, AI contributes to the different aspects of these next communications technologies such as intelligent radio (IR), self-organizing networks (SON), big data analytics [1,2], real-time intelligent edge [3], and control of the propagation environment by reconfigurable intelligent surfaces (RIS) [4]. Among them, SON would play a key role in improving the optimization, operations and maintenance (O&M), and administration activities in future networks [5]. Different types of machine and deep learning techniques have been employed in self-configuration, self-optimization, and self-healing of SONs to address various problems including backhaul, coverage and capacity, caching, and setting antenna parameters, mobility management, interreference control, optimizing the handover parameters, resource optimization, load balancing and call admission control [6, 7]. For instance in [5], local subnetworks in 5G and beyond may evolve individually to upgrade themselves. The local evolution may happen in a cluster or a few neighboring cells in order to flexibly apply cutting-edge developments on new waveforms, coding, and multi-access protocols in subnetworks without extensive time-consuming tests. In this case, each subnetwork should collect and analyze its local data, which may include the wireless environments, user requests, mobility patterns, and so on, and then exploit AI methods to upgrade itself locally and dynamically. Such methods make predictions and provide suggestions based on the results obtained by processing the datasets that are too complex and too large. In particular, in the context of telecommunication, the accurate time-series forecasting and reliable estimation of the prediction uncertainty are critical for anomaly detection, optimal resource allocation, budget planning, and other related tasks. Considering the nature of KPI data in terms of involving complex spatio-temporal variations such as seasonality, trend, and random variability, the existence of external factors and environmental changes as well as the large scale and high density of cellular networks, it is very challenging to predict the KPIs accurately. Thus, the traditional models such as univariate and multivariate statistical forecasting methods are not effective [8]. Some examples of these methods include (i) Holt Winter's exponential smoothing (HWES) and seasonal aautoregressive iintegrated moving-average (SARIMA) which are suitable for univariate time series with seasonal and/or trend components, (ii) simple exponential smoothing (SES) and autoregressive moving average (ARMA), which are suitable for univariate time series without seasonal and trend components, (iii) vector autoregression moving-average with exogenous regressors (VARMAX), which can be applied in multivariate time series without seasonal and trend components with exogenous variables*, (iv)* vector autoregression moving-average (VARMA) and vector autoregression (VAR) which have been used in multivariate time series without seasonal and trend components.

However, such methods cannot provide the required reliable prediction, in particular in a real-time manner [9]. For instance, KPI data exhibits complex behavior such as multiple seasonal patterns, non-integer seasonality, calendar effects, etc. Therefore, an accurate KPI prediction is even more challenging during these high variance segments such as holidays and sports events due to the dependency of occasional events forecasting on numerous external factors such as weather, city population growth, or marketing. Another challenge associated with the network prediction is the heterogeneity of data where different types of data in mobile networks such as counters or KPIs which show the summation of measurements like payload or traffic, or different sort of KPIs (e.g., rate, number of specific events) are collected. Therefore, a simple prediction system cannot fully provide the required precise prediction for KPIs and counters as a whole. Emerging machine learning and in particular, deep learning techniques offer a promising performance in time-series analysis and prediction. RNN as a class of artificial neural networks is a powerful tool for modeling sequence data such as time-series as it employs memory units in its architecture. In this paper, we propose an RNN approach for KPI prediction in a mobile operator to predict special seasonal events or trends in KPIs in order to appropriate actions such as expansion before facing the lack of resources. On the other hand, based on these KPI forecasting, the operators can decide on how much an area should be expanded to avoid congestion. As another example, when an optimization activity has been applied to the networks, the proposed KPI prediction model would be really useful in benchmark reports to observe if the optimization strategy has contributed to improving the network performance or resulted in any drawbacks and it should be rolled back. Applying this method to the management of wireless mobile networks has the potential to significantly increase revenue while simultaneously improving the user experience. The contributions of the proposed model are summarized as follows:

- This study has been performed on more than 10 years of actual KPIs data in the training step.
- Given the availability of long term KPI data, various time horizons for KPI prediction have been studied.
- The proposed RNN system architecture is dynamic, meaning that the RNN configuration is changed and optimized based on the type of KPI.
- The dataset is collected from a mobile operator that covers around 1.5 million km^2 and provides various cellular technologies including 2G, 3G, and LTE with about 50 million subscribers and around 280000 cells.

The rest of this paper is organized as follows. Section II reviews previous related work. Section III presents different types of data in a mobile network. Section IV discusses the proposed deep learning model. Section V represents the methodology by comparing the proposed algorithms' performance and analyzing their sensitivity to various system parameters. Section VI shows the results of RNN configuration for each KPI and presents some practical applications which use this system.

II.RELATED WORKS

Deep learning-based techniques have been utilized for time series prediction in various domains including weather forecast, economics, healthcare, telecommunication, and etc. Ben Taieb [12] has worked on different types of time-series data and developed a recurrent neural network system for three types of time-series data, namely univariate time-series, multivariate and multi-step time-series. It is worth mentioning that univariate type is a time-series with a single time-dependent variable. While, there exist more than one time-dependent variable in multivariate time-series. In addition, each variable depends not only on its previous values but also has some dependency on other variables. This dependency is used for forecasting the future values in a multi-step approach and the predictions are computed by the estimated model for each prediction horizon.

Machine learning methods have been recently utilized to optimize the performance of cellular networks. Authors in [10] discuss an application of several machine learning techniques such as exponential smoothing of time-series, Gaussian process regression, and random forests in the performance prediction of cellular networks. The utilized data set is only based on drive tests, where several classical machine learning techniques such as random forest and Gaussian process regression are applied. Bhorkar et al. [11] proposed DeepAuto as a deep learning-based approach for KPI prediction in cellular networks that uses LSTM networks horizontally to capture periodic, instantaneous, and seasonal patterns in KPI time-series. This structure further merges with feed-forward networks to learn the impact of network configurations and other external factors. In [13], the author compares the performance of different machine learning methods including auto regressive (AR), neural network (NN), and Gaussian process (GP) for traffic forecasting in cellular networks. In [15], four machine learning algorithms have been presented to predict the throughput in mobile networks. The paper includes the performance comparison of the neural network regression, algorithms based on mean throughput, multiple linear regression, and support vector regression (SVR).

Authors in [16] attempt to predict the mobile traffic in public scenes by the SOS-vSVR method, and obtain optimal traffic prediction, a symbiotic organisms search (SOS) was adopted to configure the vSVR parameters. A method has been proposed in [17] to predict the network traffic based on a deep learning technique and a spatiotemporal compressive sensing (SCS) approach. This method adopts a discrete wavelet transform to extract the low-pass component of network traffic that describes the long-range dependence of itself. In [18], the authors leverage two large-scale real-world datasets to analyze the prediction of mobile data traffic. Three techniques have been compared in [19] to predict the traffic in mobile networks based on machine learning approaches. These methods include trigonometric seasonality, Box-Cox transformation, ARIMA errors, trend, seasonal components (TBATS), Double polynomial and LSTM-RNN. [20] proposed a stochastic model to forecast the user throughput in mobile networks. Authors in [21] discuss the possibility of applying AI-based technologies to mobile network operations and presents some use cases to show proper prospects for AI-driven network operation. In [22], the authors forecast the traffic on 470 access points (APs). Various statistical properties of traffic data have been studied in this paper, such as cross-correlations and auto-correlations within and across different groups of APs. It also studies the relation between traffic generated and the number of connected users and uses seasonal auto-regressive integrated moving average (SARIMA), Holt-Winters, gated recurrent unit (GRU), long short-term memory (LSTM), and convolutional neural network (CNN), and combination of these methods for forecasting traffic usage.

III. MOBILE NETWORK DATASET

The performance of a mobile network is analyzed and monitored with KPIs and performance management (PMs) or counters. The PMs come from the aggregation of all connections in a mobile cell during a reporting output period (ROP). Usually, an ROP is defined as a 15-minute interval but its subcategories are hourly, daily, monthly, or yearly. In general, KPIs are calculated by PMs or counters, and a KPI dataset contains different counters or PM.

The PMs are classified into different types, the supported counter types in the mobile networks are listed below [14]:

- **Pegging a counter (PEG):** A type of counters which increases by 1 at each specific activity. Each count that enters this container will result in an increase of the PEG. But, leaving events will have no impact on the PEG. For instance, Call Drop is a type of PEG counter.
- **Gauge counter (GAUGE)**: This counter can be increased or decreased depending on an activity in the system. It shows the maximum number of counts that is in the container at one-time instance during the ROP. On the other hand, each count that enters this container will result in an increase. Each leaving will result in a decrease of the count. The maximum number of LTE users in a ROP is a sort of a GAUGE counter.
- **Accumulator counter (ACC)**: An accumulator counter increases by the value of a sample. It points out the total sum of all values taken during a certain time. The payload and traffic data are in this category.
- **Probability density function (PDF)**: This counter is a list of bin values. If the value falls within a certain range, the counter increases. The total number of initial E-UTRAN Radio Access Bearer (E-RAB) setup attempts per quality channel indicators (QCI) is a type of PDF.

From the time-series side, each type of PM or KPI has its own characteristics, and it should be considered separately. Hence, a single prediction system cannot be used for all sorts of PMs. In addition to these categories, some KPIs indicate the rate of measurement such as call setup success rate (CSSR) or radio access bearer (RAB) establishment success rate take values around 100%, while certain Rate KPIs take values near to zero, for example, the drop call rate (DCR) should tend to the zero. Many KPIs point out a value such as throughput, or payload and have an increasing or decreasing trend in the long term. Another sort of KPI shows the average of measurement. Comprehensively, this paper tries to consider all types of KPIs and counters, thus predicts them separately.

IV. PROPOSED DEEP LEARNING MODEL

The most important characteristic of RNNs is using the memory to process sequences of inputs. It means, unlike the convolutional neural networks (CNN) architecture, where there is no memory to capture the temporal correlation, RNNs have the capabilities to capture the sequential correlations and information. There are three types of recurrent neural networks, the simple RNN, LSTM, and gated recurrent unit (GRU) [23].

There are a number of differences between the GRU and LSTM. One missed feature of the LSTM unit in comparison with the GRU is the controlled exposure of the memory content. On the other hand, the GRU exposes its full content without any control. Besides, another difference is in the corresponding reset gate, or the location of the input gate. The LSTM unit computes the new memory content without any separate control of the amount of information flowing from the previous time step. On the other hand, the GRU controls the information flow from the previous activation when computing the new, candidate activation, but it does not independently control the amount of the candidate activation being added [24].

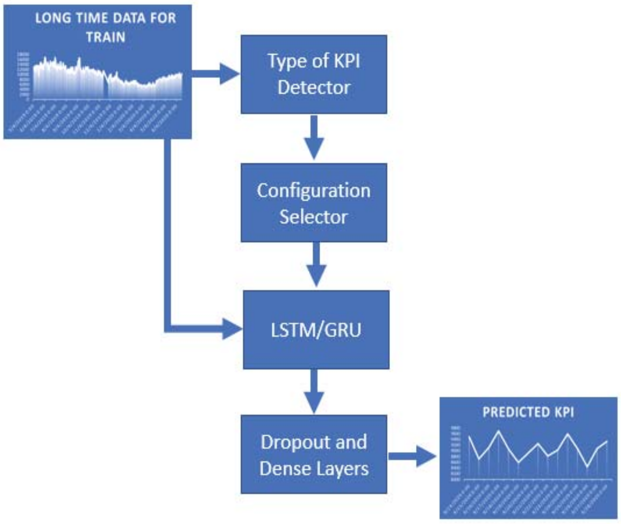

Fig. 1 DeepRANKPI architecture

Fig. 1 shows the architecture of the proposed model. In the first step, the model detects the type of KPI, and based on this detection, the proper hyper-parameters such as the number of epochs, batch size, learning rate, activation function, dropout rate, number of hidden layers and units, loss function, and optimizer are assigned to this system. Then, the dataset is passed through the RNN and Dropout/Dense layers and the prediction trend appears in the output. The flowchart in Fig. 2 shows the application algorithm for different KPIs.

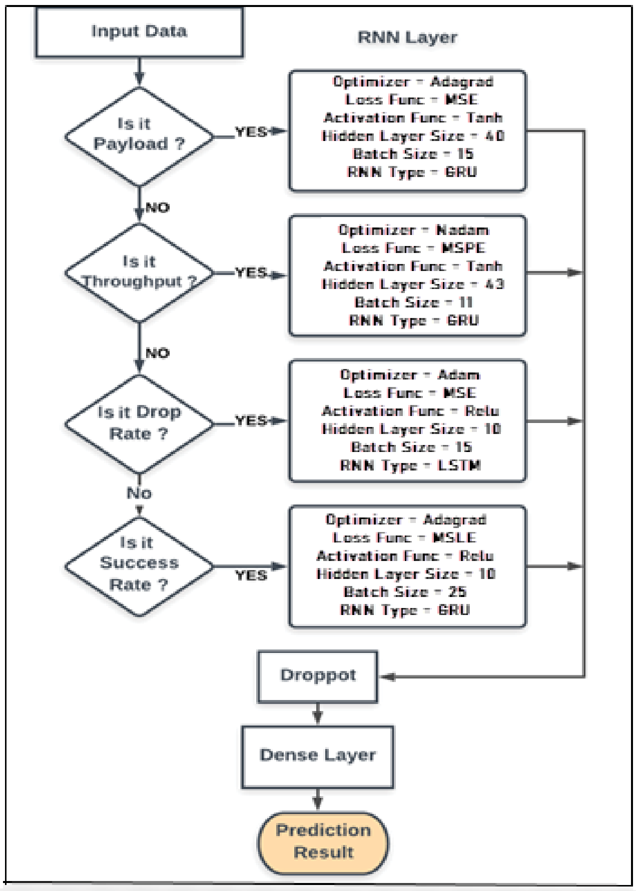

Fig. 2. The flowchart of the implemented application for different KPIs

V.METHODOLOGY

In time-series prediction, with given a univariate time-series $\{y_1, \dots, y_T\}$ including T observations, the N next observations of the time-series$\{y_{T+1}, \dots, y_{T+N}\}$ can be predicted. Time-series in different applications or fields can have a variety of resolutions (e.g. hourly, daily, monthly) with considering the T interval. In this model, some major KPIs have been selected to train the network based on their impact on the subscriber satisfaction and operators' revenue. These selected KPIs include data payload, success rate KPIs, drop rate KPIs, and throughput. The horizon (T) refers to how far in the future we can predict this trend and it can be in the range of days/weeks/months, although when it goes more than seven days, the error increases. Fig. 4 illustrates the error and time horizons.

A. Data preprocessing

In this step, the data should be separated into two parts of training and test datasets. Four years of KPI of mobile network has been divided into the training and test datasets by 80/20 percentage. Since, there is not any explicit labeled datasets, hence $y_{T+N}s$ are considered as the target or labeled data for y_N, which T refers to the next time when we want to predict. Before using the dataset, it should be cleaned and some data points such as outliers need to be removed. The dataset has been normalized by using the below formulation:

$$X = Scale\ (DataSet)$$
$$\mu = \mathrm{minimum}(X)$$
$$Normalized = -\mu * X$$

For each sample, the mean and standard deviation have been computed in the Scale function.

B. Training the model

The training part should consider all types of KPIs, thereby we cannot use a unique system for all sorts of KPIs. The hyperparameters should be tuned based on each KPI. Some of the most important parameters are the active function, optimizer, loss function, size of the hidden layer, batch size, or type of RNN (GRU or LSTM). This part of the system has been designed dynamically. It means that a detector finds the types of KPI thus, the model sets the best configuration and hyperparameters for the specific input KPI. TABLE I shows the network configurations with all possible and effective parameters for some KPIs.

TABLE I. RNN CONFIGURATION

	Payload or Traffic	Throughput	Drop Rate	Establish Success Rate
Optimizer	Adagrad	Adagrad	Adadelta	Adagrad
Loss Function	MSE	MSE	MSE	MSLE
Activation Function	Tanh	Tanh	Tanh	Relu
Hidden Layer	50	43	30	10
Batch size	15	15	11	25
RNN	GRU	GRU	LSTM	GRU
Epoch	300	200	200	70

In table I, the Loss Functions are abbreviated as following

- MAPE: mean_absolute_percentage_error
- MSE: mean_squared_error
- MSLE: mean_squared_logarithmic_error

C. Evaluation

In this section, profound analysis has been illustrated for the throughput KPI, then based on this procedure the other KPIs have been processed. This system has been trained with different combinations of hyperparameters for finding the best result. To evaluate and compare the different combinations, the root-mean-square deviation (RMSE) has been calculated to find the best system for each KPI.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_{i} - \tilde{y}_i)^2}$$

TABLE II. SOME OF THE BEST TYPES OF CONFIGURATIONS AND THEIR RMSE FOR A KPI

Type	Active Function	Optimizer	Loos function	RMSE
1	relu	Adagrad	logcosh	217.8822
3	relu	Adagrad	mean squared error	308.64
5	relu	Adagrad	mean squared logarithmic error	139.2121
6	tanh	Adagrad	logcosh	135.5813
10	tanh	Adagrad	mean squared logarithmic error	100.3922
11	relu	Adadelta	logcosh	123.4924
13	relu	Adadelta	mean squared error	150.8979
15	relu	Adadelta	mean squared logaritmic error	104.5797
16	tanh	Adadelta	logcosh	111.819
18	tanh	Adadelta	mean squared error	161.3405
20	tanh	Adadelta	mean squared logarithmic error	88.01413

Table II lists some top ranks of different configurations which are defined as types. Besides, Fig. 3 illustrates loss trends in 100 epochs for these different types of configuration when they predict the throughput.

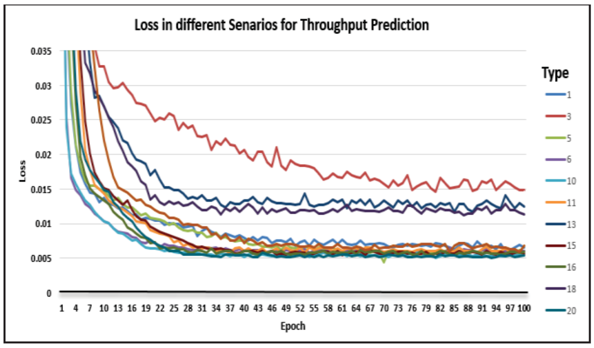

Fig. 3. Loss trend in different types of RNN configuration in 100 epochs

Fig. 4 shows the RMSE in different time horizons, it shows when this system predicts KPI in less than 7 days, is more accurate than one week ahead.

Fig. 4. RMSE in Throughput Prediction for different horizon

TABLE III. RMSE RESULT WITH DIFFERENT ACTIVATION FUNCTIONS AND OPTIMIZERS

RMSE in LSTM with 40 Hidden layer		Actication Function			
		relu	sigmoid	softmax	tanh
Optimizer	Adadelta	50.0802	717.866	386.268	26.2446
	Adagrad	86.1045	542.79	1513.89	23.6462
	Adam	175.858	246.41	267.773	208.198
	Adamax	279.579	356.74	381.273	52.2745
	Nadam	165.266	116.517	99.2077	144.938
	RMSprop	166.641	198.218	270.228	114.563
	sgd	85.5328	2710.5	3193.28	93.7166

RMSE in GRU with 40 Hidden layer		Actication Function			
		relu	sigmoid	softmax	tanh
Optimizer	Adadelta	92.6716	339.238	516.909	170.179
	Adagrad	71.7681	463.059	439.995	5.26802
	Adam	240.912	80.1843	263.753	73.6884
	Adamax	239.792	198.739	318.687	120.23
	Nadam	232.385	231.546	121.561	96.9658
	RMSprop	265.241	116.154	239.814	120.376
	sgd	126.107	1512.97	3134.74	76.687

For a selected KPI (Payload/Traffic), this system has been run to find the best configuration. TABLE III demonstrates, the RMSE per each combination. The "Tanh" function as an activation function, "Adagrad" as an optimizer, and Mean_Squared_Error as a loss function in the GRU system presented the best result with the lowest RMSE.

VI.RESULT AND PRACTICAL EXPERIMENT IN A LIVE MOBILE NETWORK

A. System result

The final proposed systems with appropriate parameters which have been presented in table III predict the payload, throughput, drop rate, and established success rate. The following figures show the accuracy by depicting the real and predict trends on daily basis.

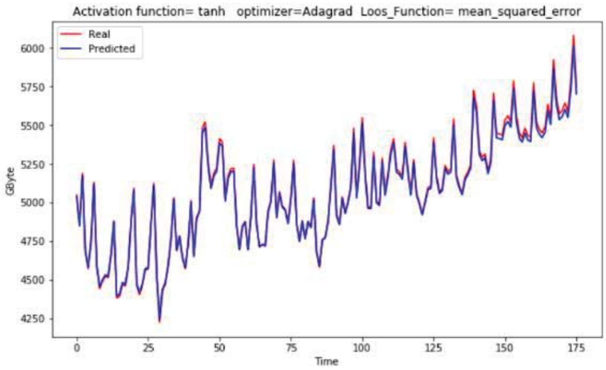

Fig. 5. Actual and prediction of Payload of sites cluster with the proposed RNN system on daily basis.

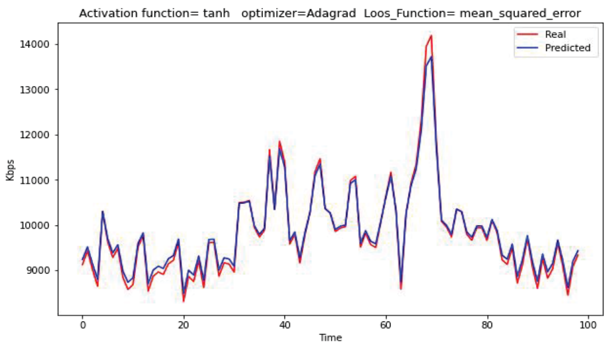

Fig. 6. Actual and prediction of throughput in a city with the proposed RNN system, and day interval.

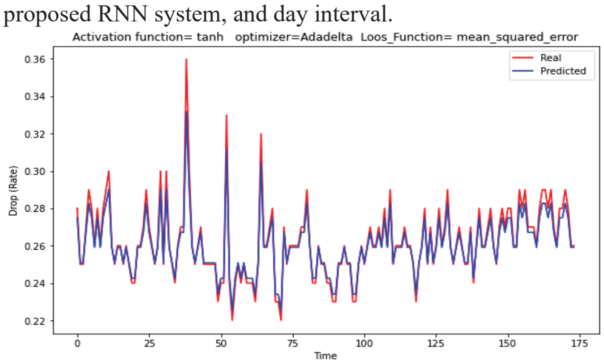

Fig.7. Actual and prediction of drop rate KPI on daily basis with the proposed RNN system.

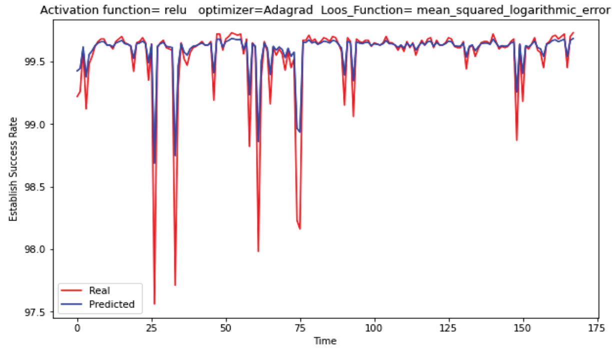

Fig. 8. Actual and prediction of daily KPI of ERAB establish success rate with the proposed RNN system.

B. Practical experiments

In this section, we present the performance of our model in a practical experiment. Several changes should be implemented in different parts of mobile networks every day, such as parameter re-tuning, strategy changes in optimization part, or cutover and expansion of fiber or microwave link in transmission department. The most important phase of these activates would be seeing the positive or negative impact of them, to decide whether they should be kept or reverted back. In this experiment, we activated the roaming between two mobile operators for some specific sites and we wanted to know if the traffic increases or decreases in these sites. In general, we do not know whether revenue was gained or lost with this implementation. By using this proposed RNN method, the KPI trend can be predicted to see the impact. Fig.6 shows the result of this deep learning system on the payload, it illustrates with this implementation (Blue) and without this activation (Orange), in other words, the orange line shows the prediction trend.

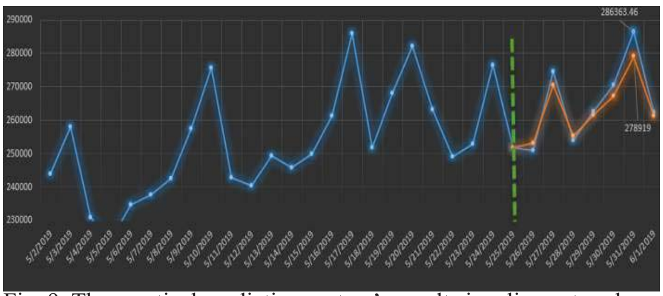

Fig. 9. The practical prediction system's results in a live network

The second presented application is the throughput analysis. An optimization action has been taken, and the result of the prediction system in Fig. 10 shows that the throughput improves after this action. It means that without this action, the throughput may be in the vicinity of 8.3 Mbps, but by means of this optimization, the throughput has been improved to 9 Mbps. Therefore, the results recommend to keep this change.

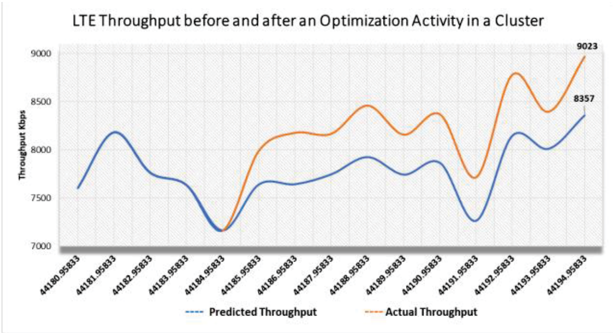

Fig. 10. The Throughput Prediction after an Optimization action.

References

[1] Han, Shuangfeng, I Chih-Lin, Gang Li, Sen Wang and Qi Sun. "Big Data Enabled Mobile Network Design for 5G and Beyond." IEEE Communications Magazine, 2017.

[2] Kibria, Mirza Golam, Kien Nguyen, Gabriel Porto Villardi, Ou Zhao, Kentaro Ishizu and Fumihide Kojima. "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks." IEEE Access 6, 2018.

[3] Peltonen, Ella, Mehdi Bennis, Michele Capobianco, Mérouane Debbah, Aaron Yi Ding, Felipe J. Gil-Castiñeira, Marko Jurmu, Teemu Karvonen, Markus Kelanti, Adrian Kliks, Teemu Leppänen, Lauri Lovén, Tommi Mikkonen, Ashwin Rao, Sumudu Samarakoon, Kari Seppänen, Pawel Sroka, Sasu Tarkoma and Tingting Yang. "6G White Paper on Edge Intelligence." ArXiv abs/2004.14850 (2020).

[4] Elmossallamy, Mohamed A., Hongliang Zhang, L. Song, Karim G. Seddik, Zhu Han and G. Li. "Reconfigurable Intelligent Surfaces for Wireless Communications: Principles, Challenges, and Opportunities." IEEE Transactions on Cognitive Communications and Networking.

[5] Letaief, Khaled Ben, Wei Chen, Yuanming Shi, Jun Zhang and Ying-Jun Angela Zhang. "The Roadmap to 6G: AI Empowered Wireless Networks." IEEE Communications Magazine, 2019

[6] Monica Paolini, "AI and Machine Learning: Why now? Network optimization in the age of 5G", Senza Fili 2019, intel

[7] Klaine, Paulo Valente, Muhammad Ali Imran, Oluwakayode Onireti and Richard Demo Souza. "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks." IEEE Communications Surveys & Tutorials 19 (2017).

[8] https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/

[9] Adhikari, R. and R. Agrawal. "An Introductory Study on Time Series Modeling and Forecasting." ArXiv abs/1302.6613 (2013).

[10] Janne Riihijärvi and Petri Mähönen, "Machine Learning for Performance Prediction in Mobile Cellular Networks", IEEE Computational Intelligence Magazine, 2018.

[11] Bhorkar, A., Kecheng Zhang and J. Wang. "DeepAuto: A Hierarchical Deep Learning Framework for Real-Time Prediction in Cellular Networks." ArXiv abs/2001.01553 (2020).

[12] Souhaib Ben Taieb, "Machine learning strategies for multi-step-ahead time-series forecasting", PhD thesis, Département d'InformatiqueUniversité Libre de Bruxelles, Belgium, 2014.

[13] Luong-Vy Le, Do Sinh, Li-Ping Tung, Bao-Shuh Paul Lin. "A practical model for traffic forecasting based on big data, machine-learning, and network KPIs" 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2018.

[14] STUDENT BOOK, "LTE L19 Performance Management and Optimization", Ericsson AB 2019.

[15] Liu, Yan and Jack Y. B. Lee. "An Empirical Study of Throughput Prediction in Mobile Data Networks." GLOBECOM 2014 (2014).

[16] Zheng, Xiaoliang, Wenhao Lai, Hualiang Chen and Shen Fang. "Data Prediction of Mobile Network Traffic in Public Scenes by SOS-vSVR Method." Sensors (Basel, Switzerland) 20 (2020).

[17] Nie, Laisen, Xiaojie Wang, Liangtian Wan, Shui Yu, Houbing Song and Dingde Jiang. "Network Traffic Prediction Based on Deep Belief Network and Spatiotemporal Compressive Sensing in Wireless Mesh Backbone Networks." Wireless Communications and Mobile Computing 2018 (2018).

[18] Chen, Guangshuo. "Spatiotemporal Individual Mobile Data Traffic Prediction." RT-0497, INRIA Saclay - Ile-de-France. 2018. ffhal-01675573v2f.

[19] KWETCZER, LOUISE ABRAHAMSSON and Jakob Stigenberg. "Mobile Network traffic prediction Based on machine learning." (2018).

[20] Bui, Nicola, Foivos Michelinakis and J. Widmer. "A Model for Throughput Prediction for Mobile Users." (2014).

[21] Tomoyuki Otani, Hideki Toube, Tatsuya Kimura, Masanori Furutani," APPLICATION OF AI TO MOBILE NETWORK OPERATION", ITU Journal: ICT Discoveries, Special Issue No. 1, 13 Oct. 2017.

[22] S. P. Sone, J. J. Lehtomäki and Z. Khan, "Wireless Traffic Usage Forecasting Using Real Enterprise Network Data: Analysis and Methods," in IEEE Open Journal of the Communications Society, 2020, doi: 10.1109/OJCOMS.2020.3000059.

[23] Tanhatalab, Mohammad Rasoul, Hossein. Yousefi, Hesam Mohammad Hosseini, Mostafa Mofarah Bonab, Vahid Fakharian and Hadis Abarghouei. "Deep RAN: A Scalable Data-driven platform to Detect Anomalies in Live Cellular Network Using Recurrent Convolutional Neural Network.", SAMI, Slovakia 2020.

[24] Chung, J., Çaglar Gülçehre, Kyunghyun Cho and Yoshua Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." ArXiv abs/1412.3555 (2014).