Deep RAN: Time-series KPIs Prediction in a Live Cellular Networks with RNN

***Abstract*— Predictive analytics is employed by telecommunications operators to get valuable insights to become better, faster, and make data-driven decisions. To be more specific, to optimize, expand and modify the strategy of mobile network quality, many parameters are changed or features are being implemented through mobile networks every day. One of the major challenges in network maintenance is to track the side-effects of mentioned changes, in order to avoid any anomalies due to configurations, by calculating the amount of degradation or improvement. Sometimes these changes coincide with other network seasonality behavior, hence, adding complications to the network. Nowadays, usage of deep learning in general and especially Recurrent Neural Network (RNN) is a promising approach to predict time-series trends in the industry. To this extent we have conducted and implemented KPI prediction using RNN algorithm in a Mobile network with about 130000 cells of various technologies and frequencies. Model training is performed on three years of historic mobile network data. In this work, on one hand we have proposed methods to predict network KPIs in future interval, On the other hand performance analysis of different methods are investigated.**

Keywords—Recurrent Neural Network, RNN, Deep Learning, Machine Learning, Keras, Time-series, LSTM, GRU

I. INTRODUCTION

Accurate prediction of data helps decision makers in many areas of scientific, economic, and industrial activities such as finance, stock exchange, meteorology, and telecommunication. In time-series prediction, historical data is used to perform forecasting. This work covers telco network counter and KPI prediction. Since the nature of data in the telecommunication network is mainly time-series and this type of data has complex nature in its properties, such as seasonality, level, and the trend, may be challenging to predict accurately. In terms of seasonality, a time-series may exhibit complex behavior such as multiple seasonal patterns, non-integer seasonality, calendar effects, etc. Thus, the traditional methods could not support us with a precise forecast. In general, in the context of telecommunication, 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. Moreover, the prediction is more challenging during high variance segments such as holidays and sport events due to dependency of occasional events forecasting on numerous external factors such as weather, city population growth, or marketing [1]. Todays, by emerging machine learning and deep learning, in particular, we can forecast the time-series data more precisely. Since, there are different types of data in mobile networks such as counters or KPIs that show the summation of measurements like payload or traffic, or different sort of KPIs, illustrate the rate, other KPIs measure the conditions, and count of an event. Therefore, a unique prediction system cannot perform precisely for KPIs and counters as a whole. One of the most important roles of the Recurrent Neural Network (RNN) is time-series prediction as it employs the memory in its architecture. Proposed RNN system considers all factors and finds the best performance for each type of KPI or counter. The strength of this application is that it has been trained via large amount of data which is collected for a period of three years. Besides, as it is an applicable project, therefore all it’s point of failures have been recovered. This application has been implemented by Keras, which is a high-level API capable of running on Tensorflow. The contributions of this work are as follow. Section II reviews previous related work; Section III presents different types of data in a mobile network; Section IV takes a look at RNN model; Section V represents the methodology by comparing the algorithms’ performance and analyzing their sensitivity to various system parameters; Section VI shows the results of RNN configuration for each KPI; Section VII illustrates the real mobile network characteristics; and the last section expresses one of the practical conducted prediction which has been done in the mentioned mobile network.

II. RELATED WORK

Bandara et al. [1] introduced a Long Short-Term Memory (LSTM) system for time-series prediction at Uber with multiple seasonal cycles. This methodology combines a series of multi-seasonal decomposition techniques to the enhancement of LSTM learning procedure. Zhi [2] worked on Deep and Confident Prediction for Time-series at Uber. In this paper the Bayesian deep model has been introduced, that provides time-series prediction along with uncertainty estimation. Bizjak et al. [3] proposed a deep learning approach to predict the behavior of Energy Management System (EMS) devices. For this purpose, a LSTM has been employed, where the input is multivariate time-series data. Ben Taieb [4] has worked on different types of time-series data and developed the Recurrent Neural Network system for tree 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. Although, more than one time-dependent variable are there 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 future values in multi-step the predictions are computed by the estimated model for each prediction horizon. In [7], four machine learning algorithms have been presented, which predict the throughput in mobile networks, and compare the Neural Network Regression, Algorithms based on Mean Throughput, Multiple Linear Regression, and Support Vector Regression (SVR). Authors in [8] try to predict the mobile traffic in public scenes by the SOS-vSVR method, and obtain optimal traffic prediction, an symbiotic organisms search (SOS) was adopted to configure the vSVR parameters. One method has been proposed to predict the network traffic based on a deep learning technique and the Spatiotemporal Compressive Sensing (SCS), the presented method adopts a discrete wavelet transform to extract the low-pass component of network traffic that describes the long-range dependence of itself [9]. In [10] the authors leverage two large-scale real-world datasets to analyze the prediction of mobile data traffic. Three techniques have been compared in [11] to predict the traffic in mobile networks based on the machine learning approach. In this paper three techniques have been tested like

TBATS[1], Double polynomial and LSTM-RNN. The [12] has proposed a stochastic model to forecast user throughput in mobile networks. Authors in [13] discuss the possibility of applying Artificial Intelligence technology to network operation and presents some use cases to show proper prospects for AI-driven network operation.

III. Mobile Network Counters

The performance of a mobile network has been 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 defined as a 15 minutes interval but its subcategories are hourly, daily, monthly, or yearly. In general, KPIs has been calculated by PMs or counters, on the other hand, a KPI contains different counters or PM.

The PMs are classified in different types, the supported counter types in the mobile networks are listed below [6]:

- **Peg counter (PEG)**: it increases by 1 at each a specific activity. For example, the *pmErabEstabAttInit counter* is a PEG counter in LTE and it stands for the total number of initial E-RAB Establishment attempts.
- **Gauge counter (GAUGE)**: This counter can be increased or decreased depending on an activity in the system. The *pmErabMax* counter in LTE is a GAUGE counter and it defines as a peak number of simultaneous E-RAB usage.
- **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. For example, the *pmErabLevSum* is an ACC counter which denotes the peak number of simultaneous E-RAB usage.
- **Scan counter (SCAN)**: It somehow looks like the ACC.
- **Probability Density Function (PDF)**: This counter is a list of bin values. If the value falls within a certain range, the counter increases. *pmErabEstabAttInitQci* is a PDF counter and defines as the total number of initial E-RAB setup attempts per QCI.

From the time-series side, each type of PM or KPI has its own characteristics, and it should be considered separately, furthermore, a unique 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, in spite of these KPI which flactuates around 100%, certain Rate KPIs are near to zero, for example in 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, and predicts them separately.

IV. Deep Learning Model

The most important characteristic of RNNs is using the memory to process sequences of inputs. It means, unlike the Convolutional Neural Network (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). It is worth to pointing out that another advantage of RNNs is going to be the size of data in input, i.e. there is no limitation for the input and output size and structure. Mainly the GRU and LSTM have been used in this work to model the network KPI's temporal behaviors.

In LSTM and GRU networks, there is a cell memory. After training the neural network, the network can select the information to save to pass through the networks, and which part of the information can be ignored. Hence, there is a forget gate and memory cell [14].

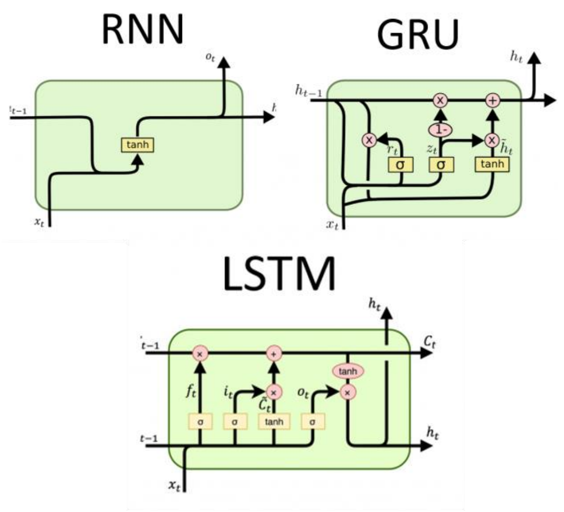

Fig 1. Recurrent Neural Network types.

Fig.1 illustrates the different types of RNN networks. In LSTM the follows notations have been defined:

h_t , C_t : Hidden layer vectors,
x_t : Output vector,
b_f, b_i, b_c, b_o : Bias vector,
W_f, W_i, W_c, W_o : Parameter matrices, and
σ, tanh: Activation functions.

In following formulas, the LSTM and GRU has been compared [15].

A. LSTM

Let x_t be the input time series for an LSTM at time t, the closed form expressions for a LSTM unit can be written as follows:

Input gate: $i_t = \sigma(w_{ih}h_{t-1} + w_{ix}x_t + b_i)$
Forget gate: $f_t = \sigma(w_{fh}h_{t-1} + w_{fx}x_t + b_f)$
Output gate: $o_t = \sigma(w_{oh}h_{t-1} + w_{ox}x_t + b_o)$

[1] Trigonometric seasonality, Box-Cox transformation, ARIMA errors, Trend, Seasonal components

where i_t, f_t, o_t are the gate parameters, w_{ih}, w_{ix}, $w_{fh}, w_{fx}, w_{oh}, w_{ox}$ are the weight vectors for the corresponding input time series, and b_i, b_f, b_o are biases for input gate, forget gate and output gate, respectively. σ represents the sigmoid activation function of the gate and h_{t-1} is the output series of previous LSTM block. After computing the states of the gates, the cell of a LSTM network computes the candidate cell state ($\tilde{c}_t$), the current cell state (c_t) and the final output (h_t) as follows:

Candidate cell state: $\tilde{c}_t = tanh(w_{ch}h_{t-1} + w_{cx}x_t + b_c)$
Cell state: $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
Final output: $h_t = o_t \odot \tanh{(c_t)}$

where tanh is the hyperbolic tangent and ⊙ means an element-wise multiplication. w_{ch}, w_{cx} and b_c are the weights and bias of the cell.

B. GRU

Like in LSTM, GRU has two gates called reset gate and update gate using sigmoid activation functions. After the gating operations, current memory, $\tilde{h}_t$ and final output, h_t are calculated since GRU does not have a separate memory cell, which is in LSTM, to compute the cell states. Let x_t be the input time series for GRU at time t, the closed form expressions for a GRU unit can be written as follows:

Reset gate : $r_t = \sigma(w_{rh}h_{t-1} + w_{rx}x_t + b_r)$
Update gate : $z_t = \sigma(w_{zh}h_{t-1} + w_{zx}x_t + b_z)$
Memory : : $\tilde{h}_t = tanh(w_{hh}(r_t \odot h_{t-1}) + w_{hx}x_t + b_h)$
Final output : $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t)$

where, r_t and z_t are the gate parameters, σ represents the sigmoid activation function of the gate, h_{t-1} is the output series of previous GRU block and ⊙ represents an element-wise multiplication. . $w_{rh}, w_{rx}. w_{zh}, w_{zx}. w_{hh}, w_{hx}$ are the weight vectors for corresponding input time series and b_r, b_z, b_h are biases for corresponding activation functions.

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 $\{y_{T+1}, \dots, y_{T+N}\}$, has been predicted. Time-series in different applications or fields can have variety resolutions (e.g. hourly, daily, monthly, yearly) with considering the T interval. In this project model, some major KPIs have been selected to train the network based on their impact on subscriber satisfaction and Operators' revenue, these KPIs are Data Payload, Success Rate KPIs, Drop Rate KPIs, and Throughput in the daily interval, and horizon (T) has been considered for seven days, this predicted time can be changed.

The proposed methodology for time-series prediction is performed in three steps.

A. *Data preprocessing*

In this step, the data should be separated into two parts train and test, since there is not any explicit labeled dataset, so the y_{T+N} are considered as the target or labeled data for y_T, which the N is next time when we want to predict. Before the dataset uses for train and test part it should be normalized.

B. *Training the model*

The training part should consider all types of KPIs, thereby we cannot use an RNN network for all sorts of data. We have different parameters for an RNN configuration like Activation Function, Optimizer, Loss Function, size of the hidden layer, batch size, number of the epoch, and type of RNN, for example simple RNN, LSTM or GRU. All these parameters have been combined and tried for this project, to see which of them has the best performance.

C. *Evaluation*

In this section profound analysis has been illustrated for the payload KPI, then based on this procedure the other KPIs have been processed. This system has been trained with all different available combinations for finding the best result. Then the Root-mean-square deviation (RMSE) has been calculated to evaluate and compare these different combinations to find the best system for each KPI. The TABLE I shows the average RMSE for each parameter, for example in RNN model rows, the 458 equals to the average of RMSE for all other parameters which using the GRU.

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

TABLE I. AVERAGE OF RMSE WITH DIFFERENT ALGORITHMS AND PARAMETERS

		Average of RMSE
RNN Model	GRU	458
	LSTM	562
Activation Function	Tanh	150
	Relu	756
	sigmoid	429
	softmax	634
Optimizer	Adagrad	389
	Adadelta	440
	Adamax	461
	Adam	196
	RMSprop	212
	sgd	1381
	Nadam	365

TABLE II. 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

The TABLE II demonstrates, RMSE per each combination, for this specific KPI the "Tanh" in activation functions, the "Adagrad" as an optimizer in GRU system has the best result. The next step is finding the optimum size of the hidden layer. Fig. 2 illustrates our analysis in another side, in this study 40 hidden layer is going to be the best performance for the Adagrad, Tanh, and GRU combination.

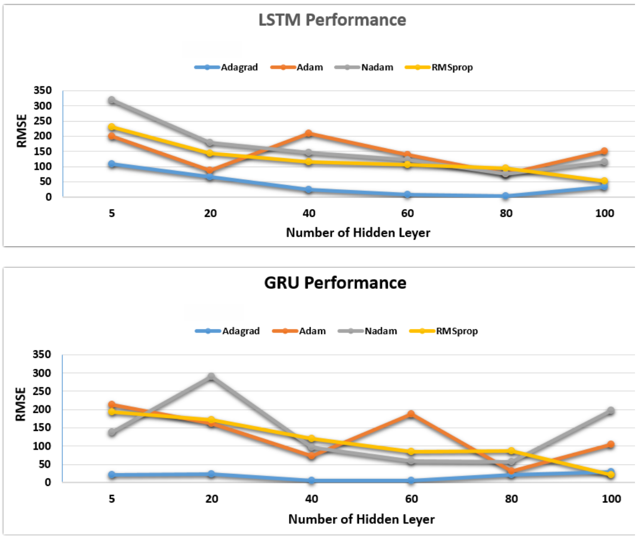

Fig. 2. Tanh activation function with verity optimizer's performance on LSTM and GRU system in different hidden layer size.

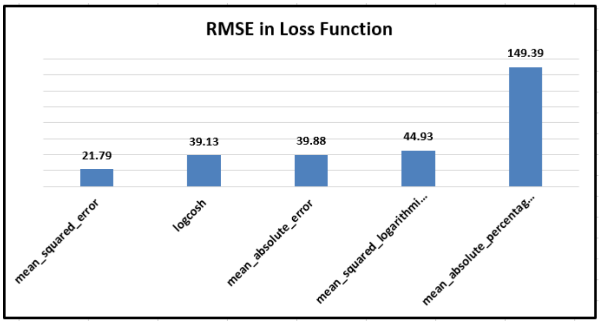

Fig. 3. RMSE in tanh, Adagrad, GRU system for different loss functions

Different types of Loss Function have been tried on the GRU with tanh, Adagrad, for Payload KPI. The result has been shown in Fig. 3. In this system Mean_Squared_Error has the best performance. Apart from these five loss functions squared_hinge, hinge, and categorical_hinge have been tried but the results were somehow outlier so they don't consider to this evaluation.

The final proposed system with appropriate parameters predicted the payload in a cluster of sites very accurately, Fig. 4 shows this accuracy.

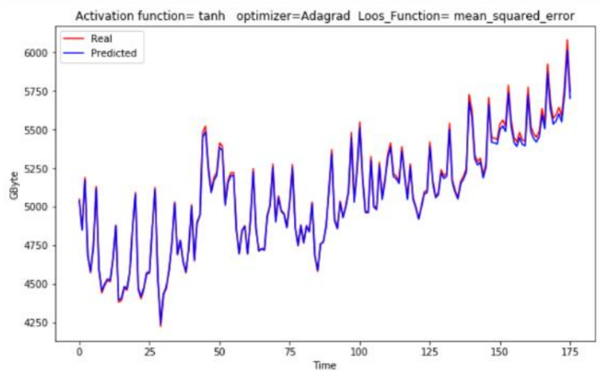

Fig. 4. Actual and prediction of Payload of sites cluster with the proposed RNN system.

VI. RNN CONFIGURATION FOR EACH KPI

In previous section the best configuration has been found for the Payload KPI, but in a Mobile networks we have many important KPIs, which need to forecast. The TABLE III shows the network configurations with all possible and effective parameters.

TABLE III. RNN CONFIGURATION

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

MAPE: mean_absolute_percentage_error
MSE: mean_squared_error
MSLE: mean_squared_logarithmic_error

The flowchart in Fig. 5 is showing the application performance in different KPIs.

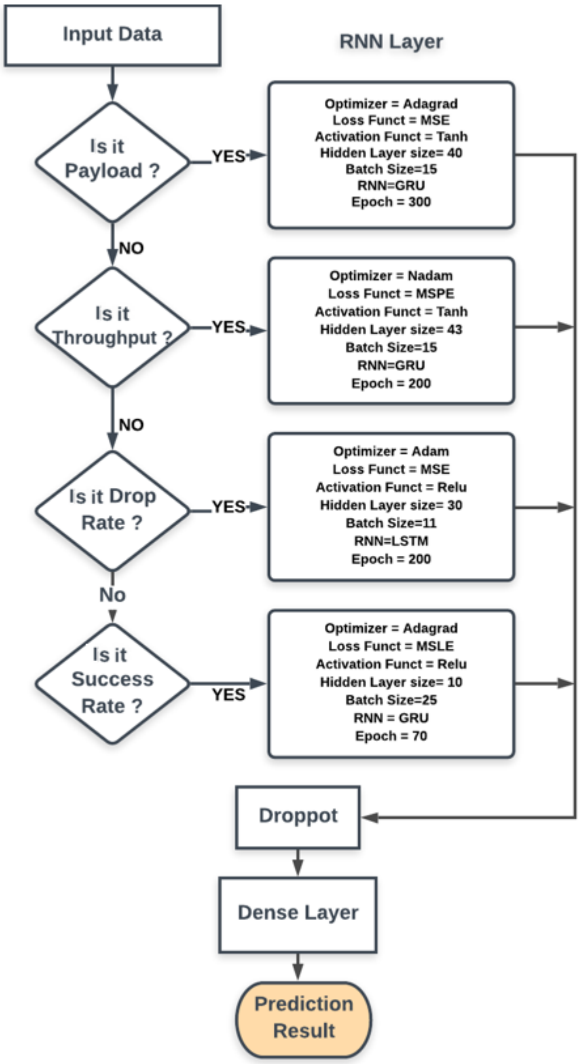

Fig. 5. The flowchart of the implemented application for different KPIs

VII. MTN IRAN-CELL CHARACTERISTICS

This application is a suitable platform for being used in Mobile Operators like MTN Iran-cell, which is officially titled Iran's biggest data operator; Iran-cell is pioneering in equipping its subscribers with the fastest and most advanced data network in the country. Besides, Iran-cell owns the largest 2G-3G-4G-4.5G RAN and modernized core mobile network, and fixed wireless TD-LTE internet services in the Middle East. It connects over 45 million people in Iran via 260,000 cell carriers in different technologies [14].

1) Performance and Optimization Challenges: As the Mobile networks advance to such large scales, the configuration, operation, optimization, and management teams will be in dire need to have intelligent tools to handle the new and diverse problems. Furthermore, such huge mobile operators these types of networks will also collect gigantic amounts of data in order to monitor, maintain and improve network stability, performance to provide better services by performing optimization actions to satisfy subscribers' expectation. In traditional approaches, some applications are used just to monitor main KPIs or revenue streams of the operator to ensure operator obligations are met and investment is protected. Based on our experiences, some most important KPI can be observed and monitored in these applications [14].

2) Application Specification: The input to our proposed platform is a set of RAN KPIs as time-series data. Main Tools characteristics are as follows:

A) The proposed plat-from is scalable since it is capable of predicting regardless of the type of KPI. It means, it is possible to input any type of KPI to the trained neural network to forecast data.

B) The technology which is used can be different network to network, so it can be used for all kinds of technologies such as 2G, 3G, 4G, 4.5G, and 5G.

VIII. PRACTICAL EXPERIMENT IN LIVE MOBILE NETWORK

In a scenario, we had activated the roaming of two Mobile Operators between some specific sites and we wanted to know if our traffic increases or lose the traffic in these sites, in general, we don't know this implementation in the network is useful or harmful. 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 KPI, it illustrates with this implementation (Blue) and without this activation (Orange), in other words the orange line shows the prediction trend.

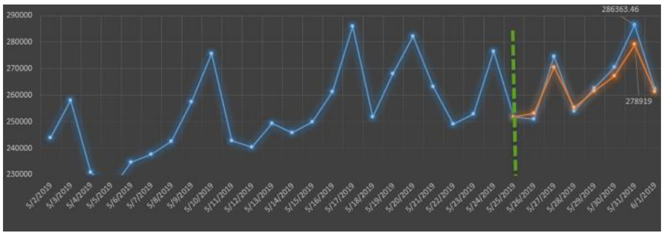

Fig.6. The practical prediction system result in a live network

REFERENCES

[1] Bandara, Kasun, Christoph Bergmeir and Hansika Hewamalage. "LSTM-MSNet: Leveraging Forecasts on Sets of Related Time-series with Multiple Seasonal Patterns." *ArXiv* abs/1909.04293 (2019): n. pag.

[2] Zhu, Lingxue and Nikolay Laptev. "Deep and Confident Prediction for Time-series at Uber." *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (2017): 103-110.

[3] Marko Bizjak, Gorazd Štumberger, Borut Žalik and Niko Lukač, "Time-series prediction for EMS with machine learning", 2019 International Conference on Engineering, Science, and Industrial Applications (ICESI).

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

[5] Popoola, Segun Isaiah, Aderemi A. Atayero, Nasir Faruk and Joke A. Badejo. "Data on the key performance indicators for quality of service of GSM networks in Nigeria" *Data in brief* (2018).

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

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

[8] 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): n. pag.

[9] 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): n. pag.

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

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

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

[13] 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.

[14] 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.

[15] 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*, vol. 1, pp. 777-797, 2020, doi: 10.1109/OJCOMS.2020.3000059.