Efficient Throughput Degradation Prediction in Telco Networks using Anomaly Detection

Rajiv Ranjan Sinha
Ericsson BCSS
rajiv.ranjan.sinha@ericsson.com

Satish Kumar Kolli, Sumit Soman
Ericsson GAIA
satish.kumar.kolli, sumit.soman@ericsson.com

Koti Prasanna Kanth Maddala
Ericsson, Netherlands
koti.prasanna.kanth.maddala@ericsson.com

***Abstract*—Proactive anomaly detection of the Key Performance Indicators (KPIs) in a telecom network is important for consistent end-user experience. In this paper, we present an application of state-of-the-art deep learning based multivariate time series Long Short Term Memory (LSTM) model which can forecast KPIs based on historical data. Our approach is able to efficiently predict KPIs by learning patterns from the time series data along with the seasonality behaviour. We also predict anomalies on this forecasted data by the application of Isolation Forest, that is tuned with the contamination hyperparameter to indicates the severity of the anomalies. We evaluate our approach on real data from a LTE network and observed good results for forecasting the throughput KPI and predicting anomalies.**

***Index Terms*—Anomaly detection, Time series, Forecasting, Telecom KPI, LSTM, Isolation Forest.**

I. Introduction

Any abnormal behavior as opposed to the general trend of data can be categorized as anomalies. Such data-points (or samples) may also be called as outliers or deviants. Anomalies can be generated by faults in machinery, or it could be an indicator to problem(s) in the performance of a network. Anomaly detection is the process of identifying those unusual patterns in the data that are not as per the expected behavior. This has diverse applications in many industry domains, for e.g., medical diagnostics, network connectivity performance, IT applications, fraud detection, among others. Capturing anomalous patterns can help uncover insights about the data. Proactively forecasting and predicting such patterns can let the owners take appropriate corrective actions to avoid such occurrences in the future. This can lead to savings in terms of time and cost, along with enhanced customer experience.

To illustrate possible anomalies, Figure 1 shows a scatter plot of representative two-dimensional data. Here, N_1 and N_2 are the areas having the maximum concentration of data points, whereas O_1 and O_2 are single isolated instances of the data, along with O_3 which is another very small, isolated collection of few data points. Similarly in Figure 2, which shows a representative plot of univariate time-series data, the huge peak in the data trend in time is an outlier. Such isolated points are the unusual behavior of the data which doesn't fit the normal pattern either in N_1 or N_2 in Figure 1, or the normal data trend in Figure 2. These outliers or anomalies potentially represent hidden insights about the data, and by analyzing them, one can understand about unexpected events or issues, depending on the data type.

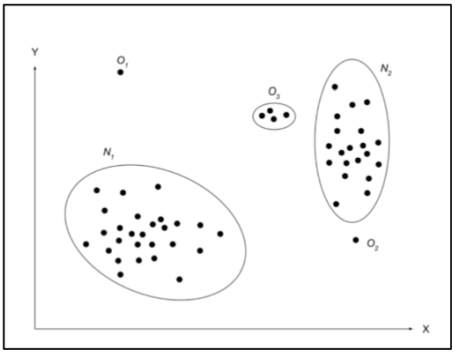

Fig. 1. Anomalies in 2-dimensional data

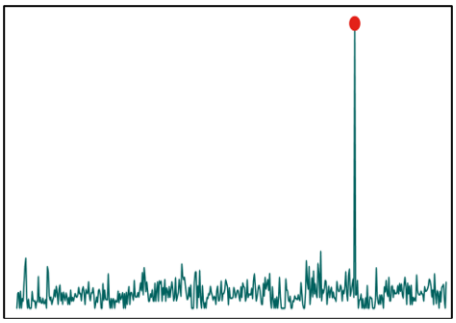

Fig. 2. Anomalies in a KPI trend

In the telecom domain, anomalies are seen very often in any KPI, which is typically monitored at various aggregation levels. This can be weekly, daily, hourly, or even at every 15 mins. The KPIs are related to the performance of the Long Term Evolution (LTE) technology air interface between User Equipments (UEs) and the base station antennas, or cells. There have been many traditional methods of monitoring and identifying the KPI anomalies by generating trends in spreadsheets, or in any other database tools as part of the network management system, and then manually performing the root cause analysis to take corrective actions. This was more of a reactive approach. In recent years, with the surge in the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques, many solutions have been developed (and are still being developed) which can proactively and automatically monitor and identify such anomalies.

There have been various works in the literature that have focused on anomaly detection using regression models [22], ensembles [9], Convolutional Autoencoders [6] and Generative Adversarial Networks [13]. Unsupervised methods for LTE networks have also been explored [3], [15], [21], including time-series clustering [10], self-organizing approaches [2], [7], [18]. Recent work has also used Federated Learning (FL) techniques through Long-Short Term Memory (LSTM) [1] for privacy-preserving anomaly detection [16], as well as the use of explainability [17] and active learning [19]. While these works have evaluated various ML and deep learning methods, they do not propose an end-to-end generic (or extensible) solution that enables proactive detection of anomalies from KPIs of a telecom network.

In this study, we choose downlink user throughput as the target KPI to be predicted, which is one of the most important KPI from the user experience standpoint. The focus of our work is to proactively capture the anomalies in this KPI by forecasting or predicting its value, based on other correlated or dependent KPIs. Examples of such dependent KPIs include resource utilization, call set up success rate, handover success rate, traffic in the cells, number of connected users and so on.

As the wireless network Radio Frequency (RF) environment is dynamic in nature, no single solution fits all the networks' behavior equally well. Also, when we develop solutions for predictive or futuristic network performance and issue identification, though a lot of work is ongoing with varying level of market claims, but on ground and in day-to-day activity of a telecom engineer, still significant manual efforts are involved. The work in this paper is performed for the same reason, that is, to ease the operational challenges faced by network engineers, with a user-friendly and easy to use solution, that can be tuned and tested for any telecom operator network. We have developed a model using LSTM to forecast the throughput KPI. Further, we have captured the target KPI anomalies with the Isolation Forest to assess the results.

The rest of this paper is organized as follows. Section II presents details of our proposed approach. Experiments and evaluation on real telecom data from a LTE network is presented in Section III. A discussion of key results and findings is included in Section IV, while conclusions and future work is presented in Section V.

II. Our Approach

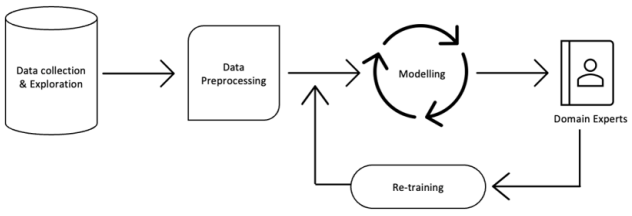

Fig. 3. Components of our Approach.

An overview of the proposed approach is shown in Fig 3. The KPI data is available in a database after aggregation from multiple cells. Data (pre)processing operations are executed to sanitize the data for the modelling phase. During training phase, modelling involves building optimal forecasting and anomaly detection models, while during inference phase, the trained models are used to obtain predictions (forecasting and anomaly detection) on the target KPI. Anomalous data points are reviewed by domain experts for root cause analysis and corrective measures, and these inputs are used in the feedback loop for re-training the models and determining their life-cycle management aspects. The detailed flowchart for our approach is shown in Fig 4. The input to the workflow is the network performance KPI data which is aggregated at a specific level, such as hourly aggregation at cell-level in our case. The data may be provided in batches, and hence may correspond to one or more specific duration(s).

The network performance KPIs, shown in Table I, comprise of the target KPI and other dependent KPIs. This is thus multivariate time series data, where we have targeted to proactively predict user downlink throughput KPI based on other dependent KPIs in the network, *viz.*, downlink physical resource block utilization, uplink Received Signal Strength Indicator (RSSI), downlink data volume, number of Radio Resource Control (RRC) connected users and call setup success rate.

TABLE I
Description of major KPIs.

KPIs	Description
DL throughput	User throughput KPI in the downlink (This is the target KPI.)
DL physical resource block utilization	Utilization of the LTE air interface Physical Resource blocks used in the downlink traffic channel.
Uplink Received Signal Strength Indicator	Uplink channel RSSI in the LTE Uplink traffic channel.
Downlink data volume	Data traffic volume in Gigabytes in the downlink direction
RRC connected users	RRC (it is a control layer protocol) layer number of connected users to the cell.
Call set up success rate	How many calls successfully set up out of all the attempts.

A. Data Pre-Processing

The first step in our approach is data cleaning and analysis, which involves correction of data types (where required), and evaluating data quality by computing the number of unique and missing values of the respective KPIs. We also perform correlation analysis of the dependent KPIs with the target KPI and selected relevant features. Subsequently, basic pre-processing steps were done, which include data normalization and selection of a validation set for evaluating the performance of our trained model.

B. Modelling

The next step is the core of our proposed approach, where we use LSTM [12] to model the time-series of the respective target KPI based on the feature (or dependent) KPIs. LSTM networks are suitable for obtaining predictions on time-series

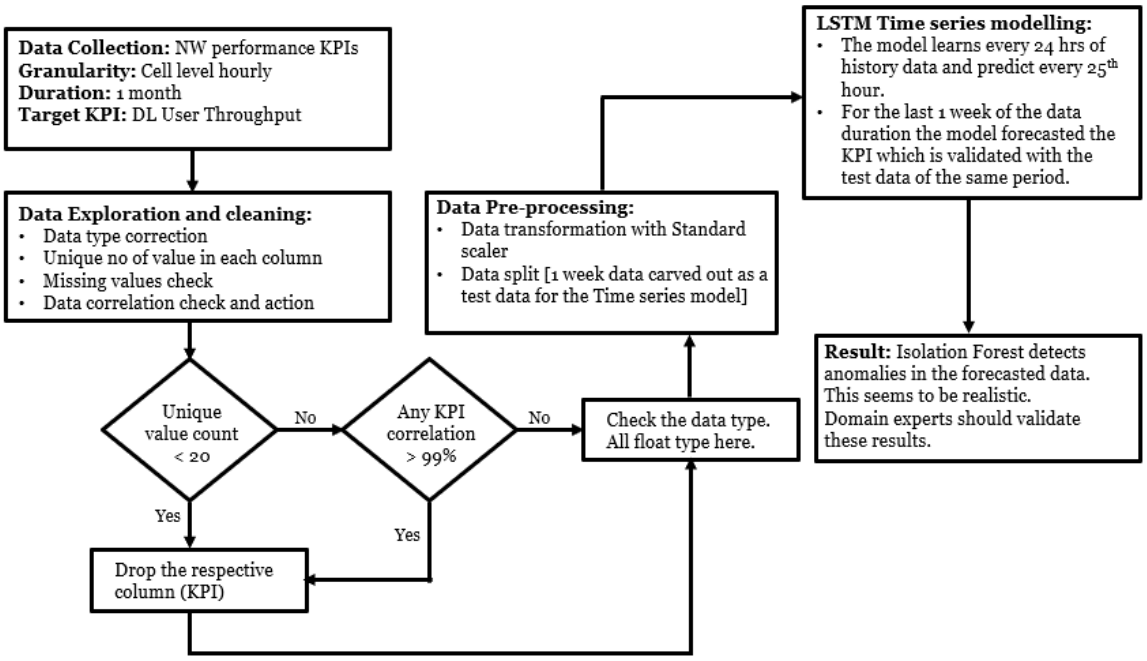

Fig. 4. Flowchart with details of our approach.

data, a basic LSTM unit comprises of a cell and an input, output, and forget gates. If the input time series to a LSTM unit is represented as $x \in \mathbb{R}^d$, and activation functions of the forget, input (update) and output gates are denoted by f_t, i_t and o_t respectively $(\in (0,1)^h)$, where cell, input and hidden state vectors are $c_t, \hat{c_t}$ and h_t $(\in \mathbb{R}^h)$, and W, U, b denote the weights (and bias) of the input and recurrent connections, then the operations of the LSTM unit at time t are given by (1)-(6), where $\odot$ represents the element-wise product.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$
$$\hat{c_t} = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c_t} \quad (5)$$
$$h_t = o_t \odot \sigma_h(c_t) \quad (6)$$

We train the LSTM to learn from the KPIs for every $t = 24$ hours of data and predict the KPI at the $(t+1)^{th}$ hour. Once we have the predicted KPIs, we use isolation forests [14], a popular anomaly detection technique, to detect anomalous KPI samples. Isolation forests generate recursive partitions (using trees) of the data sample by splitting across attribute value ranges. As such a tree is constructed, the anomalous points end up as external nodes and have short path length (from the root node). It may be noted here that the respective hyperparameters of these methods are tuned on the validation set, and this is discussed in detail in the following section, where we present experiments and evaluation of our approach on real data from a LTE network.

III. Experiments and Evaluation

A. Dataset and Environment

The dataset is collected from a live LTE network of a telecom operator. Cells are selected from different geographical locations like dense-urban, urban, sub-urban and rural areas, so to capture variations across all types of KPI data. The data is of hourly level granularity and collected over a period of 1 month. Data size is limited by the application used to fetch the data, but the intent is to collect more data to further enhance the data hungry deep learning models like LSTM. All experiments were performed in Python v3.9, using Jupyter notebooks for development. We used scikit-learn[1] and Keras[2] framework for implementations of the models used in our approach.

B. EDA, Pre-Processing and Training

Exploratory Data Analysis (EDA) was performed to understand the data distribution of each KPI. Missing value samples were dropped from the data as part of data cleaning process. Columns with very low unique value counts (less than 20) were checked and discarded, as they would not add any value in the model training. KPIs with very high feature correlation of more than 99% were also dropped before modelling. As part of data pre-processing steps, the data was transformed with sklearn Standard Scaler, that transforms each feature KPI individually to standardize the values with mean as zero and unit standard deviation.

This is calculated on z-scale on any sample data x as $z = \frac{x-\mu}{\sigma}$, where μ represents the mean and σ represents

[1]https://scikit-learn.org/stable/index.html
[2]https://keras.io/

the standard deviation of the data samples. Standard scaler is very important for any ML modeling because if some of the feature KPIs have very large mean value or high standard deviation, then it can significantly influence the objective function evaluated by the model. The last one week of data was used for validation during the model training phase.

We divided the time-series KPI data into sequence of 24 hours, which is represented by $train\_X$, and the target variable as 1 hour in future, represented by $train\_Y$. Effectively, this creates blocks of 24 hours of data with every 25^{th} hour data being the target variable to be predicted. The model is trained on the training data and prepared for forecasting. Eventually, we get $train\_X$ as $x \times 24 \times z$ attributes whereas $train\_Y$ is of $x \times 1$ size. Here x is the final training samples which is divided in blocks of 24 hours which is the second dimension. The third dimension z is the number of features or columns. In case of $train\_Y$ the second dimension is always 1 as the target feature. The sequential LSTM algorithm with 64 and 32 units is applied on $train\_X$ data array in order to predict $train\_Y$, model summary is shown in Figure 5.

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 24, 64)	19712
lstm_7 (LSTM)	(None, 32)	12416
dropout_15 (Dropout)	(None, 32)	0
dense_17 (Dense)	(None, 1)	33

Total params: 32,161
Trainable params: 32,161
Non-trainable params: 0

Fig. 5. Model Summary

The model is fitted on $train\_X$ and $train\_Y$ to learn for 10 epochs with validation split of 0.1. Both training loss and the validation loss converge quickly, as shown in Figure 6.

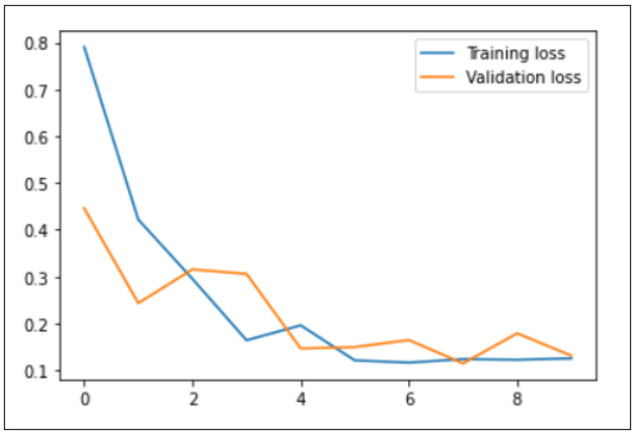

Fig. 6. Convergence of the losses

Forecasting is done for 1 week in future at hourly level. The forecasted data is inverse transformed to get the real KPI value. Figure 7 shows the overall trend of history data of 3 weeks and the forecasted data for the last 1 week period. Figure 8 shows the comparative view of the test data which was separated out in the beginning and the forecasted data in the same period.

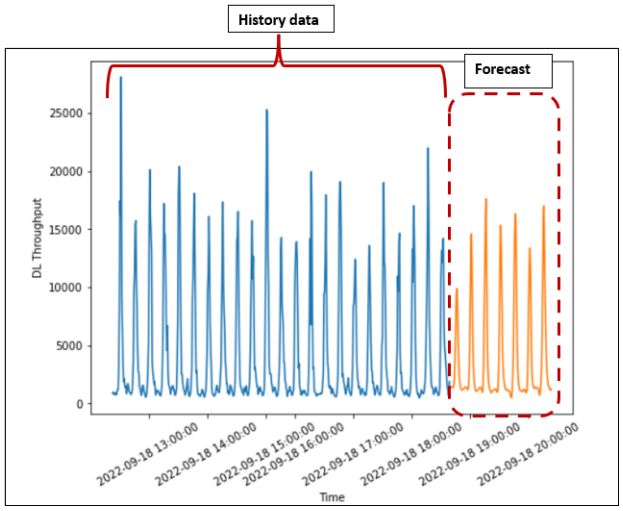

Fig. 7. Overall KPI trend from history till forecasting

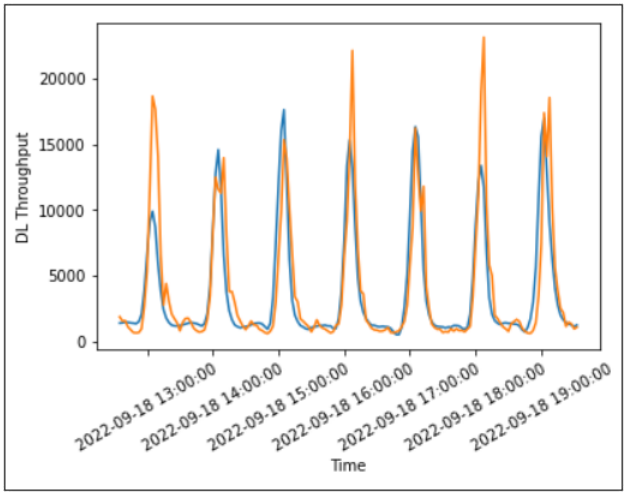

Fig. 8. Comparison of forecasted and test data

We applied Isolation Forest to detect anomalies on the forecasted data. The algorithm was trained on 3 weeks of historical data excluding the test data. This is done so that the algorithm can learn the patterns well. Then, the forecasted data was applied to predict the anomalies. One of the hyperparameter of the Isolation Forest algorithm, the contamination factor, plays a very important role here. Fine-tuning of this parameter resulted in getting valid anomaly points in the data. It was set at 0.04 in the final iteration. Figure 9 shows the anomalies detected by our approach. They are also shown in a tabular form with the corresponding time stamps in Table II.

Table III shows the anomaly categories which are captured by the algorithm. This shows that the algorithm is able to clearly differentiate between the peaks and the troughs in the

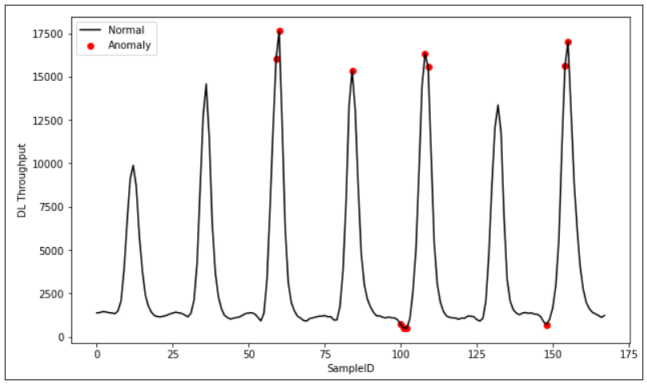

Fig. 9. Detected anomalies in the forecasted data

TABLE II
ANOMALY POINTS IN THE FORECASTED DATA

Forecasted Data Anomalies			
Sample	**DL_User_Tpt**	**Time**	**Date**
59	16043.143	01:00:00	14-10-2022
60	17621.152	02:00:00	14-10-2022
84	15326.776	02:00:00	15-10-2022
100	705.56885	18:00:00	15-10-2022
101	506.4707	19:00:00	15-10-2022
102	504.02393	20:00:00	15-10-2022
108	16325.84	02:00:00	16-10-2022
109	15559.652	03:00:00	16-10-2022
148	704.51	18:00:00	17-10-2022
154	15602.449	00:00:00	18-10-2022
155	16993.635	01:00:00	18-10-2022

KPI data. From the nature of the throughput KPI results as shown in Table II, it has been categorised as cases more than 10000 Kbps and cases less than 1000 kbps. It is understood that a good performing network should have always good user throughput for the best user experience throughout the network. The cases with throughput less than 1 Mbps in particular needs to be investigated further. The results with the time stamps can be evaluated by the domain experts for verification and further troubleshooting to take proactive corrective measures in the network.

TABLE III
HIGH LEVEL ANOMALY CATEGORIZATION

Anomaly Categories	
Very high samples $>$10000 kbps	Very low samples $<$1000 kbps
7	4

IV. RESULTS AND DISCUSSION

We observe that the forecasted data, as shown in Fig 8, is quite close to the actual test data. The Coefficient of Determination (R2) value is 0.72, while we obtain a Root Mean Squared Error (RMSE) of 2461.06 and Mean Absolute Error (MAE) of 1458.62. We worked with 1 month data for the cells in a network. For deep learning algorithms like LSTM, the more the data that is available for training, we can expect better generalization. This is also part of our proposed future work to manage longer periods of data efficiently using our approach. The forecasting accuracy would also be expected to increase with high data volume for training.

Further, the algorithm is also able to clearly differentiate between the peaks and the troughs in the KPI data as seen Fig 9. We need to study these outliers and analyze their root cause with domain experts.

From Fig 6 the convergence of training loss and the validation loss could have been smoother. Here, we observe it converges well till $6-7$ epochs. In general, we expect this to get stable if provided with more training data. With more data, the value of number of epochs can be increased to let the loss converge gradually. Our efforts illustrate a minimum viable solution that can be obtained with limited training data, which would be a useful starting point for network engineers, as handling high volumes of data at scale would require use of suitable compute and storage infrastructure. Isolation Forest algorithm for anomaly detection worked very well.

V. CONCLUSION AND FUTURE WORK

This work presented a LSTM and isolation forest based proactive anomaly detection approach for network throughput KPI using other dependent KPIs. The approach could be explored further for scalability and efficiency with data intensive networks so that we have accurate forecasting for both long term and short term, along with anomaly detection. We obtained good results for the anomaly detection which can be discussed further with the domain experts for root cause analysis and preventive actions. The inference period can also be customized as per the KPI or solution requirements. Also, the LSTM model developed in our approach can be used as a standalone model for capacity forecasting. We plan to explore other uni-variate approaches using Prophet and ARIMA models for the time series based forecasting and anomaly detection.

With ever rising technology and network complexity with 5G networks, it is very important to catch network issues at an early stage to take all possible actions proactively. This will not only increase efficiency and reduce costs but also improve customer experience. As part of future work, our approach can be tested more with data from different networks. This would increase its reliability and accuracy further. Similar models can be replicated for other KPIs in the network depending on the nature of the KPI and data [4], [5], [8], [11], [20], [22]. This can also be deployed as a service-based solution on cloud, where the end-user (owner) can fetch the results on an easy-to-use web-based interface.

REFERENCES

[1] SM Abdullah Al Mamun and Mehmet Beyaz. Lstm recurrent neural network (rnn) for anomaly detection in cellular mobile networks. In *Machine Learning for Networking: First International Conference, MLN 2018, Paris, France, November 27–29, 2018, Revised Selected Papers 1*, pages 222–237. Springer, 2019.

[2] Levente Bodrog, Marton Kajo, Szilard Kocsis, and Benedek Schultz. A robust algorithm for anomaly detection in mobile networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[3] Kalle Burbeck and Simin Nadjm-Tehrani. Adaptive real-time anomaly detection with incremental clustering. *information security technical report*, 12(1):56–67, 2007.

[4] Pedro Casas, Pierdomenico Fiadino, and Alessandro D'Alconzo. Machine-learning based approaches for anomaly detection and classification in cellular networks. In *TMA*, 2016.

[5] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[6] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pages 1–5. IEEE, 2018.

[7] Gabriela Ciocarlie, Ulf Lindqvist, Kenneth Nitz, Szabolcs Nováczki, and Henning Sanneck. Dcad: Dynamic cell anomaly detection for operational cellular networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–2. IEEE, 2014.

[8] Gabriela Ciocarlie, Ulf Lindqvist, Kenneth Nitz, Szabolcs Nováczki, and Henning Sanneck. On the feasibility of deploying cell anomaly detection in operational cellular networks. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–6. IEEE, 2014.

[9] Gabriela F Ciocarlie, Ulf Lindqvist, Szabolcs Nováczki, and Henning Sanneck. Detecting anomalies in cellular networks using an ensemble method. In *Proceedings of the 9th international conference on network and service management (CNSM 2013)*, pages 171–174. IEEE, 2013.

[10] Borislava Gajic, Szabolcs Nováczki, and Stephen Mwanje. An improved anomaly detection in mobile networks by using incremental time-aware clustering. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1286–1291. IEEE, 2015.

[11] Imed Hadj-Kacem, Sana Ben Jemaa, Sylvain Allio, and Yosra Ben Slimen. Anomaly prediction in mobile networks: A data driven approach for machine learning algorithm selection. In *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE, 2020.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Jiajia Huang, Ernest Kurniawan, and Sumei Sun. Cellular kpi anomaly detection with gan and time series decomposition. In *ICC 2022-IEEE International Conference on Communications*, pages 4074–4079. IEEE, 2022.

[14] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[15] Mahmoud Nour, Mina Awad, Mina Kamel, Mostafa Essa, and Nashwa Abdelbaki. Anomaly detection using unsupervised learning in lte mobile network. In *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 195–199. IEEE, 2021.

[16] Jiaming Pei, Kaiyang Zhong, Mian Ahmad Jan, and Jinhai Li. Personalized federated learning framework for network traffic anomaly detection. *Computer Networks*, 209:108906, 2022.

[17] Juan M Ramírez, Fernando Díez, Pablo Rojo, Vincenzo Mancuso, and Antonio Fernández-Anta. Explainable machine learning for performance anomaly detection and classification in mobile networks. *Computer Communications*, 2023.

[18] Pichanun Sukkhawatchani and Wipawee Usaha. Performance evaluation of anomaly detection in cellular core networks using self-organizing map. In *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 1, pages 361–364. IEEE, 2008.

[19] José Antonio Trujillo, Isabel de-la Bandera, Jesús Burgueño, David Palacios, Eduardo Baena, and Raquel Barco. Active learning methodology for expert-assisted anomaly detection in mobile communications. *Sensors*, 23(1):126, 2022.

[20] Song Wang, Juan Fernando Balarezo, Sithamparanathan Kandeepan, Akram Al-Hourani, Karina Gomez Chavez, and Benjamin Rubinstein. Machine learning in network anomaly detection: A survey. *IEEE Access*, 9:152379–152396, 2021.

[21] Simon Wanjiru et al. *Long Term Evolution anomaly detection and root cause analysis for data throughput optimization*. PhD thesis, University of Nairobi, 2020.

[22] Jun Wu, Patrick PC Lee, Qi Li, Lujia Pan, and Jianfeng Zhang. Cellpad: Detecting performance anomalies in cellular networks via regression analysis. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 1–9. IEEE, 2018.