60th International Scientific Conference on Information, Communication and Energy Systems and Technologies, Ohrid, North Macedonia, June 26–28, 2025

A Comparative Analysis of Anomaly Detection Techniques in Cellular Data

Nikol Gotseva[1] , Atanas Vlahov[2], Roland Mfondoum[3], Antoni Ivanov[4] and Vladimir Poulkov[5]

Abstract – **In this paper we compare various anomaly detection techniques for cellular network data using a multivariate dataset from Milan and Trentino. We evaluate traditional statistical methods (Z-score, IQR), machine learning (One-Class SVM), and deep learning approaches (LSTM-Autoencoder, Transformer). Experimental results indicate that deep learning models significantly outperform traditional methods in terms of both accuracy and efficiency. The Transformer model achieved 96.5% accuracy in 23 epochs, compared to 93% by the LSTM-Autoencoder in 40 epochs and 86% by the One-Class SVM.**

Keywords – **Anomaly detection, Outlier detection, Cellular network analytics, Transformer-based anomaly detection, LSTM-Autoencoder, One-Class SVM**

I. INTRODUCTION

Cellular networks are the backbone of modern communication systems. They facilitate a wide range of everyday activities, such as phone calls, video streaming, and IoT data transmission. Every day, these activities generate enormous amounts of real-time data, which makes it challenging to maintain their precise and reliable operation. An essential part of their maintenance is anomaly detection (AD), which is the process of identifying unusual patterns in the operational data. Their source could be technical faults, cybersecurity threats, or inefficient system performance.

Manual analysis of vast streams of key performance indicators and log events is impractical due to the high amount of workforce and time required. To solve this issue automated AD algorithms were introduced. These solutions provide significant benefits for network operators: they enable real-time fault identification and irregularity detection, reduce downtime and supports data-driven network planning by uncovering long-term behavioral patterns and usage trends [1].

Conventional statistical algorithms such as Z-Score and Interquartile Range (IQR) have been established as effective techniques for automated anomaly detection through the years. They provide a simple and interpretable basis for identifying values that deviate from the normal operating patterns. However, with the increasing dynamicity and complexity of today's cellular networks, these methods become less and less effective. Their reliance on predefined thresholds and rules makes them unsuitable for the highly variational and nonlinear nature of the network's operational data. These drawbacks have led to a trend toward using Machine Learning (ML) in the AD process. One-Class Support Vector Machine (SVM) and Isolation Forest are among the most widely used machine learning algorithms for AD, due to their ability to adjust to dynamic data patterns, without the need for pre-defined thresholds and rules. Additionally, with the recent advancements in Deep Learning (DL), algorithms such as Long short-term memory (LSTM) and Transformer models [2], have also demonstrated excellent performance in time series analysis [3], offering accurate real-time anomaly identifications [4]. These advanced approaches can help reduce false alarms and increase the systems' reliability, by offering more precise and adaptable detection of anomalous behavior in the network's operational data[5].

In this paper we present a comparative analysis of various AD techniques ranging from traditional statistical methods to modern ML and DL approaches. We evaluate their effectiveness in handling the dynamic nature of the multivariate cellular network traffic and discuss their accuracy and limitations. As an outcome, the Transformer model achieved an accuracy of 96.5%, while the LSTM-Autoencoder achieved 93.01%. Notably, the OC-SVM accuracy was 86.01% without the need of any finetuning.

II. OVERVIEW OF AD TECHNIQUES

The methods for anomaly detection can be classified into statistical methods, ML methods and DL methods.

A. *Statistical methods*

Statistical methods can be divided into two categories, such as parametric and non-parametric methods. An example of a parametric method is Gaussian-based (GB) detection, which assumes normally distributed data around its mean value [6]. The Z-score method is Gaussian-based, and it gives a measure of the distance to the mean value of each observation by standard deviation.

[1]Nikol Gotseva is with Technical University of Sofia, Faculty of Telecommunications, ul. Professor Georgi Bradistilov 11, 1756 Sofia, Bulgaria, E-mail: ngoceva@tu-sofia.bg

[2]Atanas Vlahov is with R&D&I Consortium, Laboratory of Intelligent Communication Infrastructures, blvd. Tsarigradsko shose 111, 1784 Sofia, Bulgaria, E-mail: ici-lab@sofiatech.bg

[3]Roland Mfoundoum is with Technical University of Sofia, Faculty of Telecommunications, ul. Professor Georgi Bradistilov 11, 1756 Sofia, Bulgaria, E-mail: mfondoum2000@gmail.com

[4]Antoni Ivanov is with Technical University of Sofia, Faculty of Telecommunications, ul. Professor Georgi Bradistilov 11, 1756 Sofia, Bulgaria, E-mail: asivanov@tu-sofia.bg

[5]Vladimir Poulkov is with Technical University of Sofia, Faculty of Telecommunications, ul. Professor Georgi Bradistilov 11, 1756 Sofia, Bulgaria, E-mail: vkp@tu-sofia.bg,

979-8-3315-2655-9/25/$31.00 ©2025 IEEE

A z-score can be calculated using the following Z- score formula:

$$z = (X - \mu) / \sigma \quad (1)$$

where z is Z-Score, X = Value of Element, μ = Population Mean, σ = Population Standard Deviation. When the Z-score is above 2 or below -2, it is considered that the data point is unusual or significant. When the score is above 3 or below -3, the data point is considered an outlier.[7]

Another statistical method is the Interquartile range (IQR). Quartiles [8] are statistical measures that divide a dataset into four equal parts. Q1 separates the lowest 25% of the data from the rest. Q2 (Median) divides the data into two equal halves. Q3 separates the highest 25% of the data from the rest. The IQR is a measure of variability that indicates the range of the middle 50% of the data. It is calculated as: IQR = Q3 – Q1. We can identify potential outliers by looking for data points that are more than 1.5 times the interquartile range (IQR) below the first quartile (Q1) or above the third quartile (Q3). This means any value outside the range [Q1 - 1.5 × IQR, Q3 + 1.5 × IQR] is considered a potential outlier.

Non-parametric statistical methods are designed to overcome the limitations of parametric approaches when the underlying data does not follow a Gaussian distribution. For example, histogram-based and kernel-based. [6]

B. ML based methods

Distance-based models are among the most prominent ML-based methods for anomaly detection. Typically, these methods use supervised learning and utilize the spatial distribution of points in a space S and assume that in an n-dimensional dataset, a distance d can be calculated between any two of the observations. Cluster-based methods leverage unsupervised learning, operating under the assumption that related observations tend to be spatially grouped. They follow a two-step process: first, clusters are identified within the data; then, observations or groups of observations are classified as outliers or not. Angle-based methods assess the variance of the angle between vectors of a point to others to calculate their proximity. Support Vector Machines (SVMs) is a supervised ML technique that aims to find hyperplanes that divide any dataset represented into spatially separated classes with the maximum margin between them. It is suitable for classification tasks; however, it is impractical for unlabeled streaming data. An adaptation of SVM suitable for anomaly detection is One-Class SVM (OC-SVM), which is an unsupervised method that trains the model to learn normality in data and detects deviation in new data [6]. It is trained only on the normal data to learn the boundaries of these points. This boundary defines what is considered a region of normality. Then, it can classify any points that lie outside the boundary as outliers. The main difference between the standard SVM and OC-SVM is that the latter is characterized by a hyperparameter ν which is used to control the sensitivity of the support vectors. [9]. OC-SVMs are a powerful and adaptable technique used in anomaly identification that separates regular patterns from anomalous events. Interestingly, OC-SVM implements a unique strategy by training only the majority class, which stands for normal cases. Thus, the requirement for labeled anomalous data in the training process is eliminated. This is especially helpful in real-world situations where it can be difficult to get enough labeled samples since abnormalities are rare events.

C. DL based methods

DL models enable the exploration of complex temporal and feature connections making them an excellent tool for tasks such as time-series analysis. Among these, reconstruction-based models are particularly effective for AD. They are trained to reconstruct the original signal from its input and identify anomalies within the data based on the reconstruction error. A notable class of reconstruction-based models is the LSTM, which is widely used due to its ability to model sequential dependencies. LSTM-Autoencoders encode time-series data into a latent space and decode it back to its original form. Abnormalities are found when the reconstruction error is higher than a certain value. Despite their effectiveness, LSTM models struggle with finding long-term dependencies, especially in noisy data. Additionally, they require extensive training time and computational resources.

Due to their ability to process multiple tasks at once, transformer models provide an alternative way to find anomalies in time series data. Unlike LSTMs, Transformers can model long-range dependencies more efficiently. For instance, TranAD [10] is a transformer-based network that is specifically designed for anomaly detection in time-series data. It uses encoder-decoder architecture together with attention mechanisms to focus on critical time points. TranAD prioritizes both low training costs and high detection accuracy. It learns global patterns and deviations from the original. The Informer network [11], another model, also optimized for forecasting long sequences and anomaly detection. It utilizes a probe-sparsed self-attention mechanism and is also able to focus on key temporal features.

III. METHODOLOGY

In this paper, we present a reconstruction-based model that leverage a transformer architecture to detect anomalies in cellular data. Initially, we were motivated by the success of transformer models in detecting anomalies in ECG data [12]. Their ability to capture unusual patterns both local and global proved highly effective. Given these results, we decided to investigate the performance of the same architecture in the context of cellular network data. To evaluate this, we compare the performance of the transformer model to other approaches, including LSTM-Autoencoders, Interquartile Range (IQR), Z-score, and One-Class SVM.

A. Dataset Description

The telecommunication component of the multi-source dataset for the city of Milan and the Province of Trentino [13] is used. Each of these two areas is divided into grids

corresponding to squares with size of about 235 x 235 meters. Milan is composed of 1,000 squares and Trentino of 6,575 [14]. The Semantics and Knowledge Innovation Lab (SKIL) of Telecom Italia provided the Call Detail Records (CDRs) by which the traffic values were measured. Every time a user interacts with the network, a new CDR is created where the time of the interaction and the Radio Base Station (RBS), which was assigned by the operator, are recorded. The user`s geographical location can be obtained from the corresponding RBS because the coverage maps C_{map} show the portion of territory which each RBS serves. Every interaction is associated with the coverage area v of the RBS which handled it. Thus, the generated CDRs are aggregated according to the grid they belong to. The number of records $S_i(t)$ in a grid square i at time t is computed as follows:

$$S_i(t) = \sum_{v \epsilon C_{map}} R_v(t) \frac{A_{v \cap i}}{A_v} \quad (2)$$

where Rv,j(t) is the number of records in the coverage area v at time t, A_v is the surface of the coverage area v and $A_{v \cap i}$ is the surface of the spatial intersection between v and square i. The records are temporally aggregated in time slots of ten minutes. The number of records in the datasets $S_i'(t)$ is calculated:

$$S_i'(t) = S_i(t).k \quad (3)$$

where k is a constant defined by Telecom Italia, which hides the true number of calls, SMS and connections. The dataset provides measurements for a two-month period from November 1st, 2013 to January 1st, 2014 and the information is geo-referenced to the city of Milan and to the Province of Trentino. The collected data shows the level of interaction between the users and the network. The features included in the used dataset are: SquareId(GridId) , Time Interval, SMS-in activity, SMS-out activity, Call-in activity, Call-out activity, Internet activity, which are described in [14].

B. Data processing

The initial data consists of 62 text files corresponding to the 62 days of measurements. These files were initially concatenated into a unified dataset. The data was aggregated and each sample within this dataset represents an hourly measurement. To reduce the computational complexity in the early stages of model development and evaluation, the dataset was decreased by including only 6 out of the 10 top grids with the highest internet usage: 5059, 5159, 5262, 5061, 5259, and 6064. The time interval and the country code features were removed because of their irrelevance for the current analysis.
A thorough examination of the data revealed that there are inconsistencies in the measurements after 22nd December, probably due to the holiday season. To ensure data quality and model reliability, these anomalous measurements were excluded from the dataset.
The final dataset, with the shape of (8928, 7) including the startTime attribute, was chronologically sorted and split into training and testing datasets. The training set included data up to December 11th, 2013, while the testing set included the remaining measurements up to December 22nd.

The startTime feature was transformed as the index for the dataset. To incorporate temporal information into the model, time-based features such as hour, day of week, day of month, and month were extracted from the startTime timestamps. GridIDs and timestamps were excluded from the datasets during training, but were kept for visualization purposes. The dataset was standardized for ensuring optimal model performance.

C. Anomaly injection

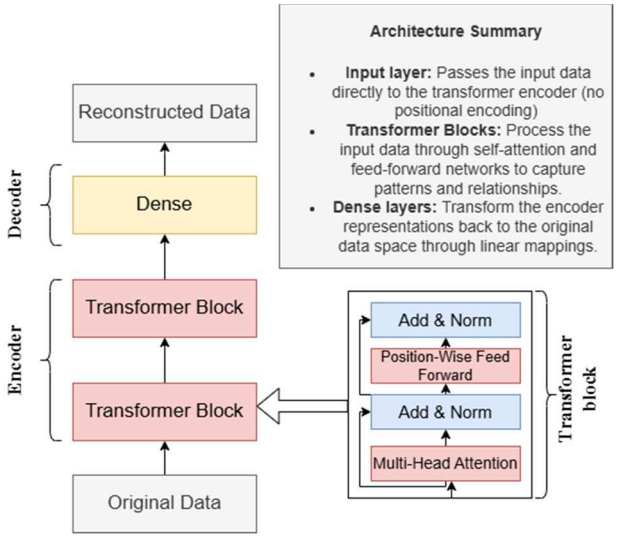
Since the original dataset consists of normal network traffic data, all measurements were initially assigned with anomaly label 0. Artificial anomalies that were introduced to the testing dataset are described in Table 1.

TABLE I
INJECTED ANOMALIES

Anomaly type	**Grid ID**	**Date**	**Time**	**Anomaly label**
Internet spike	5059	14.12.2013	10am-8pm	1
SMSIn drop	All grids	18.12.2013	10am-8pm	2
CallOut drop	All grids	16.12.2013	10am-8pm	3

The anomalies described in Table 1 simulate real network issues. The spike in usage represents Distributed Denial of Service (DDoS) attack and the sudden drops simulate network failure. These anomalies were strategically injected during peak hours between 10 AM to 8PM. At this timeslot the network traffic is typically higher which reflects the most productive hours of the day.

D.Transformer-based model



. Fig. 1. Transformer architecture

The model developed in [12] was partially replicated and applied for anomaly detection scenarios in cellular networks.

From the original transformer architecture just the transformer encoder block is utilized for the development of unsupervised model for anomaly detection. By training the model on only the normal data, it learns its distribution. Then it can detect anomalies by calculating the loss function between the reconstructed data and the original data.

The architecture consists of a transformer encoder with two encoder layers and a simple decoder. The first encoder layer creates hidden representations of the input. In order to create higher level representations the output of the first encoder layer is given as input to the second encoder layer. Each transformer block serves as an encoder layer and is built around two core components: a multi-head self-attention mechanism and a position-wise fully connected feedforward network (FFN). Both sublayers have a residual connection and layer normalization. The model architecture is presented on Fig. 1

The steps to detect anomalies with transformer neural network are the following:

1) Training on normal time series data
2) Determine a threshold for the reconstruction error
3) Test the trained model with new time series data
4) Every data point that has reconstruction error above the fixed threshold is labeled as an anomaly

This principle is central to both the Transformer and the LSTM-Autoencoder models in this work.

E. LSTM-Autoencoder

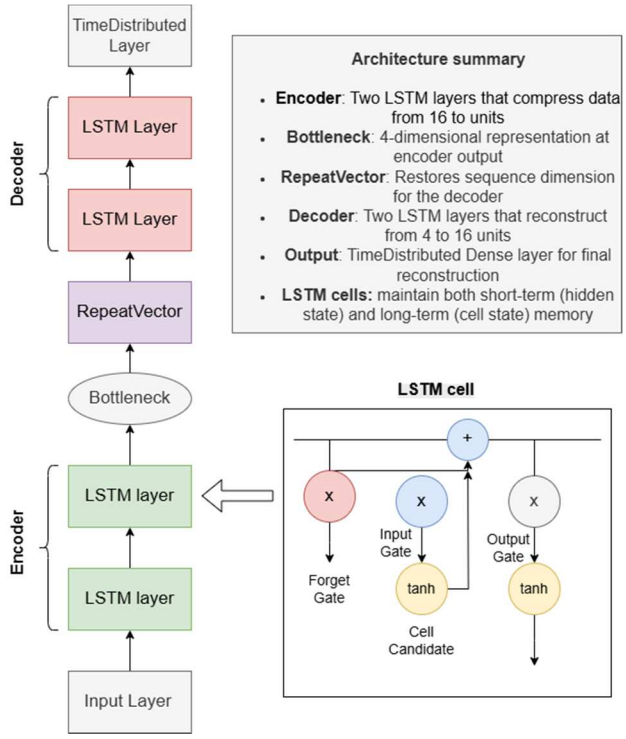


Fig.2 LSTM-Autoencoder architecture

The LSTM-Autoencoder follows the same approach. First, it is trained on normal time-series data which will be efficiently encoded. However, when unseen abnormal data is given to the network, the decoder will not be able to reconstruct it since this data is an unfamiliar pattern. Anomaly is detected when the reconstruction error exceeds the threshold. In this study we used the architecture of the LSTM-Autoencoder presented on Fig.2.

IV. RESULTS

To make identical conditions and evaluate accurately the two DL models, we used the same parameters during the training phase. Data points were categorized as normal or anomalous using an 80th percentile and Mean Squared Error (MSE) was used as the loss function. In the testing phase both models accurately detected many of the artificially introduced anomalies, but they also identified additional points as abnormal. Presumably the cause of this behavior is that reconstruction-based AD techniques may mistakenly consider the variations from the reconstructed pattern as abnormalities. We must also consider that real raw cellular data was used, which may imply that the data originally had irregularities.

A. Transformer

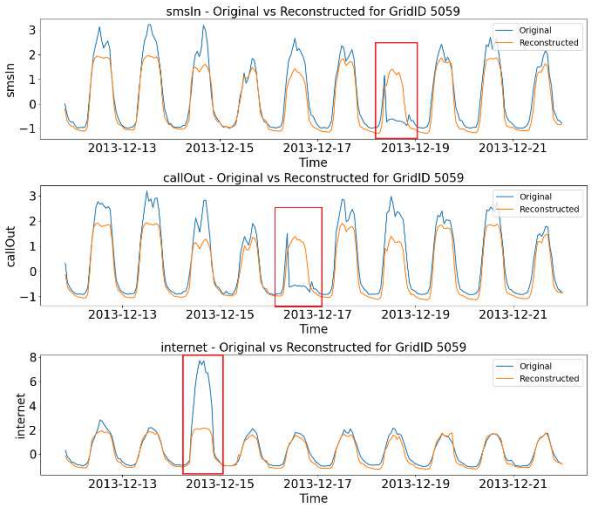


Fig. 3 Transformer reconstructed vs. original data for grid 5059

The Transformer model demonstrated satisfactory reconstruction performance, as visualized on Fig. 3 for grid 5059. At first the model was trained and tested on normal traffic data only. When the loss function for both training and testing was MSE, it identified 294 anomalies within the test dataset. Afterwards we injected 143 anomalies into the test dataset, as described in methodology section, to evaluate the model`s ability to detect specific anomaly types. A new instance of the model was then trained on the original non-anomalous training data and re-evaluated on the modified test dataset. The model was trained for 23 epochs, as the ultimate number to mitigate

overfitting. In the testing phase individual loss calculation for each data point in a batch manner was applied. Finally, 138 of the 143 injected anomalies were detected and accuracy of 96.5% was achieved.

B. LSTM-Autoencoder

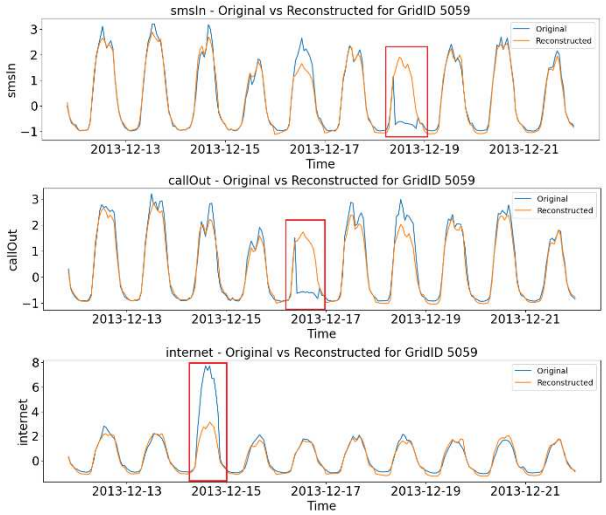


Fig.4 LSTM-Autoencoder reconstructed vs. data signals for grid 5059

MSE was utilized as the loss function. At first no regularization was applied. The injected anomalies were implemented as described in the Anomaly Injection section. When trained over 10 epochs, the model detected 129 anomalies. However, after a learning curves analysis, there were signs of overfitting. Ultimately with 40 training epochs, MSE loss, a validation split of 0.05 and no regularization, the model overcame the overfitting. It achieved 93.01 % accuracy, but this improvement was at the cost of an increased computational time.

C. One-Class SVM, IQR and Z-score

By applying One-Class SVM to the test dataset containing injected anomalies, we successfully identified 294 anomalous points, 123 out of the 143 injected. This impressive result highlights the model`s ability to detect anomalies without requiring complex training.

Applying the IQR algorithm to the test dataset containing injected anomalies, we identified only 10 of the 143 anomalies. These anomalies were primarily located in the internet traffic due to their significant deviation from the median values, making them more easily detectable.

A threshold of 3 was applied to our Z-score model, resulting in the identification of 10 out of 143 injected anomalies within the internet traffic dataset. This performance is consistent with the anomaly detection capabilities demonstrated by the IQR-based approach.

TABLE II
COMPARISON OF THE MODELS

Model	Epochs	Found Anomalies	Accuracy	Execution time
Transformer	23	138	96.50%	29.45s
LSTM-Autoencoder	40	133	93.01%	77.13s
OC-SVM	-	123	86.01%	0.266s
Z-Score	-	10	6.99%	0.215s
IQR	-	10	6.99%	0.111s

V. CONCLUSION

This study assessed how well five anomaly detection models performed on cellular network data. These models performed differently, and each had distinct advantages and disadvantages. Statistical models like Z-score and IQR did not work well at all and couldn't pick up on the complex patterns of the data. This result demonstrates that traditional methods for finding anomalies are becoming less useful for the dynamic nature of today's cellular network operational data. Despite its simplicity, One-Class SVM detected 123 anomalies without any complex training, demonstrating the efficiency of a lightweight machine learning method in identifying anomalies in this specific dataset. The Transformer model achieved a solid 96.5% accuracy for this dataset, outperforming the LSTM-Autoencoder, which reached approximately 93% accuracy. The Transformer required only 23 training epochs and 29.45 seconds of execution time, whereas the LSTM-Autoencoder needed 40 epochs and 77.13 seconds, highlighting the Transformer's superior efficiency.

VI. FUTURE WORK

To further evaluate the studied models, we plan to add labels to the dataset, so we can calculate more comprehensive performance metrics such as F1 score, recall, precision and confusion matrices. To do so a voting system comprised of high-performing anomaly detection algorithms will be implemented. In this system each algorithm will independently analyze the dataset and decide if specific data point is anomalous or not. If the majority of algorithms mark the same data points as anomalous it will be labeled as such. This consensus-driven approach will allow us to generate reliable labels that can be used by us and the research community for future model evaluation and benchmarking.

ACKNOWLEDGEMENT

This work was funded by the European Union NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, BG-RRP-2.005 - "Twinning" with Project No. BG-RRP-2.005-0002 titled "Twinning for Excellence in Research in Sustainable Future Communication Networks in the Context of a Green Economy – GREENBEAT".

REFERENCES

[1] Zhang, Chaoyun & Patras, Paul & Haddadi, Hamed. (2018). Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys & Tutorials. PP. 10.1109/COMST.2019.2904897.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008)

[3] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In NeurIPS, 2019.

[4]Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly Transformer: Time series anomaly detection with association discrepancy. In ICLR, 2022.

[5] Li, B., Zhao, S., Zhang, R., Shi, Q. and Yang, K. (2019), Anomaly detection for cellular networks using big data analytics. IET Communications, 13: 3351-3359. https://doi.org/10.1049/iet-com.2019.0765

[6] Mfondoum, R. N., Ivanov, A., Koleva, P., Poulkov, V., & Manolova, A. (2024). Outlier Detection in Streaming Data for Telecommunications and Industrial Applications: A Survey. *Electronics*, *13*(16), 3339. https://doi.org/10.3390/electronics13163339

[7]Z-score in statistics, available on[https://www.geeksforgeeks.org/z-score-in-statistics/], last accessed: 13.05.2025

[8]Quartile, available on[https://www.investopedia.com/terms/q/quartile.asp], last accessed: 13.05.2025

[9] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. 2020. Network Anomaly Detection Using LSTM Based Autoencoder. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '20). Association for Computing Machinery, New York, NY, USA, 37–45. https://doi.org/10.1145/3416013.3426457

[10] Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data.

[11] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting

[12] Alamr, Abrar, and Abdelmonim Artoli. 2023. "Unsupervised Transformer-Based Anomaly Detection in ECG Signals" *Algorithms* 16, no. 3: 152. https://doi.org/10.3390/a16030152

[13] A multi-source dataset of urban life in the city of Milan and the Province of Trentino Dataverse, available on [https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGZHFV], last accessed: 13.05.2025

[14] Barlacchi, G., De Nadai, M., Larcher, R. *et al.* A multi-source dataset of urban life in the city of Milan and the Province of Trentino. *Sci Data* 2, 150055 (2015). https://doi.org/10.1038/sdata.2015.55