Method of detecting anomalies in IOT device traffic based on statistical analysis using the modified z score*

Mykola Stetsiuk [1,*,†], Volodymyr Anikin [1,†], Olena Pyrch [1,†] , Oleksandr Kozelskiy[1,†] and Abdel-Badeeh M. Salem[2,†]

[1] *Khmelnytskyi National University, Khmelnytskyi, Instytutska street 11, 29016, Ukraine*
2 Ain Shams University, El-Khalyfa El-Mamoun Street Abbasya, Cairo, Egypt

Abstract

The article proposes a statistical method for detecting anomalies in the network traffic of Internet of Things (IoT) devices, which does not require the use of machine learning methods or pre-training on labeled data. The concept of the approach is based on building a profile of the normal behavior of each device using a modified Z-index that uses the median and median absolute deviation (MAD) to increase the robustness to noise, outliers and traffic irregularities. Anomaly events are defined as those that fall outside the limits of statistically justified acceptable intervals. The method is supplemented with a mechanism for filtering insignificant deviations using weighting factors that take into account the influence of individual parameters on the overall behavior of the device. Additionally, exponential smoothing and a cumulative deviation index are used to detect both short-term and long-term anomalies. The proposed approach provides high adaptability to changes in the network activity of IoT devices, does not require large computing resources and can be used in real-time in monitoring systems. The practical implementation of the method involves the generation of device activity logs with controlled anomalies, which allows assessing the accuracy and sensitivity of the algorithm. Various types of deviations were simulated, including changes in the frequency of messages, the number of connection errors, the intervals between messages and authorization attempts. The method showed the highest efficiency in detecting temporal anomalies, demonstrating an average accuracy of over 84%. Since the method does not depend on prior information about the types of attacks, it is particularly suitable for protecting dynamic and heterogeneous IoT infrastructures.

Keywords

IoT security, anomaly detection, network traffic analysis, machine learning, Autoencoder, intrusion detection, cybersecurity threats.1

Intelitsis'25: The 6th International Workshop on Intelligent Information Technologies & Systems of Information Security, April 04, 2025, Khmelnytskyi, Ukraine

* Corresponding author.

† These authors contributed equally.

mykola.stetsiuk@khmnu.edu.ua (M. Stetsyuk); anikin_volodymyr@khmnu.edu.ua(V. Anikin); pyrchov@khmnu.edu.ua (O. Pyrch); oleksandr.kozelskiy@khmnu.edu.ua (O. Kozelskiy); abmsalem@yahoo.com (Abdel-Badeeh M. Salem)

0000-0003-3875-0416 (M. Stetsyuk); 0000-0003-3395-2764 (V. Anikin); 0009-0002-6485-4462 (O. Pyrch); 0009-0002-7157-6499 (O. Kozelskiy); 0000-0003-0268- 6539 (Abdel-Badeeh M. Salem);

1. Introduction

The world is rapidly entering the Internet of Things (IoT) era, with the number of connected devices expected to exceed 75 billion by 2025. At the same time, the massive expansion of IoT poses serious challenges to cybersecurity, with over 77.9 million attacks recorded in 2023, up 37% from 2022. Botnets such as Mirai and Mozi exploit vulnerable IoT devices to carry out DDoS attacks, data theft, and industrial espionage, threatening critical infrastructure. Traditional security methods cannot handle dynamic IoT traffic, which requires new algorithms for real-time threat detection.

This paper proposes a combined approach that combines statistical methods (modified Z-score, Rosner test, Holt-Winters method) with machine learning (Autoencoder) for effective anomaly detection. Experiments conducted on the generated IoT traffic logs demonstrate that this combination significantly reduces the level of false positives and ensures detection accuracy. The proposed approach allows to adapt the security system to the specificities of the IoT infrastructure and to operate stably even with limited resources.

2. Analysis of known solutions

The paper [1] presents an entropy-based method for identifying IoT devices based on calculating the entropy of network traffic parameters. The method uses the Random Forest machine learning algorithm to classify devices based on traffic characteristics in various network scenarios. The proposed approach demonstrated high efficiency, achieving 94% device classification accuracy.

The key element of the study was the smart home experimental setup, which allowed collecting IoT device traffic and creating the IoTTGen tool for traffic modeling and emulation. The authors compared synthetic and real traffic and analyzed their entropy properties to assess the impact of anomalies. The experimental results confirmed significant differences in traffic behavior depending on the device type and scenario, which emphasizes the importance of the entropy approach for anomaly detection. [2]

The authors [3,4] examined in detail the locality-aware anomaly detection (LSAD) method for detecting anomalies in the network traffic of IoT devices. The method is based on the use of the Nielsims hash function, which allows for efficient matching of network traffic patterns. The main advantage of LSAD is that there is no need to pre-extract features from the data, which significantly simplifies its adaptation compared to traditional machine learning methods. As part of its work, LSAD generates signatures from the traffic of protected devices and determines the threshold value T by calculating the average similarity of hashes. Further comparison of hashes of new flows with the threshold value allows for quick detection of abnormal deviations. A comprehensive evaluation of LSAD effectiveness was conducted based on a dataset of 15 different types of attacks on IoT devices, including ARP spoofing, Ping of Death, and TCP SYN flooding. Experimental results show that the method provides an average true positive rate of over 97% using only one minute of network traffic for analysis. Particular attention is paid to comparing LSAD with popular single-class machine learning models, where the proposed method outperformed analogs in the accuracy of detecting volumetric attacks. In addition, LSAD demonstrated high performance with minimal computational costs, making it suitable for implementation in real-world IoT systems with limited resources.

In [5], a locality-aware anomaly detection (LSAD) method was considered for detecting anomalies in the network traffic of IoT devices. The method is based on the Nielsims hash function for matching traffic patterns and detecting deviations. The advantage of LSAD is the ease of adaptation, since it does not require preliminary feature extraction from the data, unlike classical machine learning methods.

The principle of LSAD is to create signatures of protected traffic and determine a threshold value T based on the average hash similarity. Comparing new flows with this threshold value allows for effective anomaly detection. The effectiveness of LSAD was experimentally evaluated on a set of 15 attack types (ARP Spoofing, Ping of Death, TCP SYN Flooding). The method demonstrated a true positive rate of over 97% when analyzing one-minute traffic. Compared to single-class ML models, LSAD demonstrated higher accuracy and minimal computational costs, which makes it suitable for implementation in resource-constrained IoT systems. In addition, the k-means algorithm was used to study the behavioral clustering of devices in the LoRaWAN network. The analysis of over 997 thousand packets from 2169 devices allowed us to identify five main clusters of network activity. The authors studied the difference between stable and anomalous device behavior patterns. [6] The clustering approach using internal verification indices (WCSS, Davis-Bouldin) confirmed the robustness of the model. The results demonstrate the potential of combining cluster analysis with anomaly methods to improve the cybersecurity of IoT networks.

The authors in [7] study an anomaly detection method in IoT cybersecurity using ensemble machine learning and Bayesian hyperparameter sensitivity analysis. They propose a unified approach that combines multiple models into an ensemble using Bayesian optimization to tune the hyperparameters. The authors detail a hyperparameter optimization method using the Tree Parzen Estimation (TPE) algorithm. They analyze the impact of key model parameters, such as the number of trees in a random forest and the minimum leaf size in a decision tree, on the anomaly detection accuracy. [8] Experimental results on the IoTID20 and IoT-23 datasets demonstrated that ensemble models, especially XGBoost and Stacking, achieve the best performance in terms of accuracy (F1 score above 96%), outperforming traditional models. The authors highlight the advantages of ensemble learning in reducing false positives and stability of results. Thus, the study demonstrates the effectiveness of ensemble machine learning for anomaly detection in IoT networks and the importance of hyperparameter tuning to improve model performance.

In the papers [9,10], the authors propose a novel approach to detect global anomalies in distributed IoT systems using direct communication between devices (device-to-device communication). The method is based on the WAFL-Autoencoder (Wireless Ad Hoc Federated Learning) model, which allows devices to jointly train autoencoders without transmitting local data. A special feature of the study is the introduction of the concepts of local and global anomalies. Local anomalies are rare for a device, but can be common to others. In contrast, global anomalies are rare deviations in the entire device network. To effectively search for global anomalies, the authors developed a coordinated thresholding algorithm in a distributed environment. Experimental results on standard datasets (MNIST, Fashion-MNIST) demonstrated high accuracy in detecting global anomalies with a low false positive rate. The proposed WAFL-Autoencoder outperformed traditional centralized approaches, validating the effectiveness of fully distributed training and its potential for secure and efficient monitoring of IoT networks.

In the papers [11,12], the authors explore the use of deep learning (DL) methods to detect anomalies in IoT network traffic. They review state-of-the-art deep anomaly detection (DAD) approaches and implement an ensemble model based on the KDD Cup 99 dataset. The authors highlight the advantages of DL in discovering complex patterns in high-dimensional data, especially for IoT environments.[13]

The research methodology involves the use of various DL models such as GAN, CNN, LSTM, and AutoEncoder. Normalization, feature scaling, and hyperparameter selection methods were used for the experiments.[14] The main focus was on the interpretability of the models and their performance.

Experimental results showed that the CNN+LSTM ensemble model with Random Forest classifier achieved the highest accuracy of 98.22%, outperforming the other implemented models (AE and GAN). The ensemble also demonstrated the lowest false positive and false negative rates. The authors highlight the potential of combining DL models and ensemble approaches for efficient monitoring of IoT network traffic.[15]

The paper [16] presents a systematic mapping of research on anomaly detection in industrial equipment using IoT devices and machine learning (ML) algorithms.[16]

The focus is on three types of equipment: milling tools, hydraulic systems, and bearings, as they are the most prone to wear. The key sensors for anomaly detection were identified: vibration, temperature, current, and pressure sensors.[17,18] The effectiveness of a combined approach using multiple sensors to improve the detection accuracy is highlighted. The authors also reviewed common ML algorithms: neural networks (MLP, LSTM, CNN), outlier detection algorithms (OCSVM, Isolation Forest), and heuristic methods. It is noted that ensemble models and autoencoders in combination with data preprocessing (FFT, PCA, normalization) demonstrate the highest accuracy. Industry challenges are specifically addressed, including limited resources of edge devices, lack of outlier data, and the need to regularly retrain models to adapt to changes in manufacturing processes.[19]

In [14], the use of CNN, LSTM, and their combined CNN-LSTM neural networks for analyzing IoT device traffic and detecting attacks is investigated. The KDDCup99, NSL-KDD, UNSW-NB15, WSN-DS, and CCIoT2023 datasets were used to train the models, allowing us to test the effectiveness of the approach in various scenarios.[20,21]

The main problem noted by the authors is the heterogeneity of IoT networks, which complicates the global configuration of security systems. The proposed CNN-LSTM-based solution provides high accuracy, but the effectiveness of the method is highly dependent on the type of attack and device class, which may affect the detection results.[22]

The main drawback is the high computational complexity, which limits the possibility of implementing such systems in real IoT networks. This confirms the relevance of research in the direction of lightweight statistical methods that can ensure the effectiveness of risk identification without spending a lot of resources.[23]

The study presents an interesting solution for modeling a DDoS attack scenario on specialized information systems. The authors use a stochastic network model that allows them to estimate the potential capabilities of attackers, the time it takes to implement an attack, and its impact on critical resources [24]. The process of analyzing the parameters of the network environment, which includes the characteristics of the probability of attack actions and their impact on the target system, is very important [25]. A key feature of the application is the ability to modify this model to detect unusual network traffic behavior and develop protective measures. The combination of deterministic and stochastic methods allows not only to recreate the cyber-attack scenario, but also to assess the effectiveness of protective measures. However, given the dynamic nature of attacks on IoT devices, the proposed approach can be complemented with machine learning techniques, which will improve the accuracy of anomaly detection in data streams.

3. Detecting traffic anomalies

Anomalies in IoT device traffic are deviations from expected behavior that can indicate security risks or technical defects. It is important to identify such deviations quickly and accurately.

However, the above-mentioned methods lose their utility in large data sets due to their sensitivity to noise and outliers. To overcome these issues, an approach based on the z-index correction method is proposed, which provides high robustness and accuracy due to the use of median and median absolute deviation (MAD).

To detect anomalies, it is first necessary to analyze the data coming from IoT devices. Such information includes activity timestamps, connection parameters, error messages, network resource usage, and other characteristics that indicate the device's performance.

Based on this information, a set of key characteristics is established, which makes it possible to estimate the real behavior of the device. This is necessary to establish a training network whose output reflects anomalous activity.

To assess the normality of each device, the range of acceptable values is calculated using the Corrected z-index method. The upper and lower limits separating normal values from abnormal values are defined:

$$U_{norm} = median + threshold * MAD_{lower} \tag{1}$$

$$L_{norm} = median - threshold * MAD \tag{2}$$

where threshold is a parameter that determines the level of permissible deviation, MAD is the median absolute deviation, which is a stable measure of data dispersion.

To determine the deviation of a particular device, the normalized deviation function is introduced:

$$D(x)\ \frac{x - median}{MAD} \tag{3}$$

Any values outside these limits are considered anomalies and moved to the next level of analysis.

The anomaly detection process in IoT device traffic consists of a few basic steps, which allow sequential analysis of all device parameters, identification of deviations from the norm, and assessment of the critical level of the detected anomalies.

In the first step: anomalies in the values of each parameter for different devices are detected. The values obtained for this purpose are compared with the limits of normal behavior. If the value exceeds the specified threshold, it is flagged as a possible anomaly.

$$Z_{mod} = \frac{0.675 * (x - median)}{MAD} \tag{4}$$

$$|Z_{mod}| > threshold \tag{5}$$

where threshold is the set threshold coefficient that determines the permissible deviation (usually 3).

The aggregated average deviation for the entire data set can be expressed as:

$$M_{dev}\frac{1}{n}\sum_{i=1}^{n}|Z_{mod}| \tag{6}$$

where is the total number of records in the sample.

In the second stage, after identifying potential anomalies, it is necessary to determine their main characteristics. To do this, each detected anomaly receives such attributes as the device identifier, the name of the parameter that has exceeded the normal limits, and the intensity of the anomaly.

The intensity of the anomaly is defined as the ratio of the deviation of the parameter to its limit:

$$Intensity = \frac{|x - boundary|}{|boundary|} \tag{7}$$

where boundary is the corresponding upper or lower limit of the normal range. Anomaly Duration is the period of time during which the parameter was outside the normal range.

Anomaly Duration is an indicator of how long the device was in an anomaly state:

$$Duration = t_{end} - t_{start} \tag{8}$$

In the third stage, we filter out insignificant anomalies. Accordingly, in order to reduce the number of false positives, filtering of insignificant anomalies is used. Each parameter has its own influence weight, which allows filtering out insignificant deviations that do not pose a threat. Filtering is carried out using the weight coefficient:

$$Filtered = \sum_{i=1}^{n} W_i * |Z_{mod,i}| \tag{9}$$

where W_i is the significance factor for each parameter. This procedure allows you to focus on the most critical anomalies and avoid analyzing random parameter fluctuations.

Not all detected anomalies are critical for the system. To avoid false positives, a filtering mechanism for low-significant anomalies is used.

To adaptively determine the anomaly level, a weight factor is introduced for each device:

$$W_i \frac{Z_{mod\ i}^2}{\sum_{j=1}^{n} Z_{mod\ j}^2} \tag{10}$$

The final anomaly score for a device is calculated using a weighted sum:

$$A_{score} = \sum_{i=1}^{n} W_i * |Z_{mod\ i}| \tag{11}$$

In the fourth stage, the total number of anomalies for each device is counted, its activity indicators are determined, and the duration of anomalous behavior is analyzed. The integral anomaly indicator is also calculated:

$$A_{score} = \frac{\sum_{i=1}^{n} Interval_{anomalu,i}}{\sum_{i=1}^{n} Interval_{total,i}} \quad (12)$$

where $Interval_{anomalu,i}$– is the total time spent in the anomaly state, and Interval_(anomalu,i) is the total duration of observation.

For the practical implementation of the method, a software prototype in Python was developed. Each algorithmic component is presented as a separate module, which allows easy integration of the method into any IoT device monitoring system.

This approach provides high efficiency in detecting anomalies in real time, allowing for rapid response to possible risks and threats in IoT networks.

To reduce the impact of random fluctuations in the data, exponential smoothing is used:

$$S_t = \alpha Z_{mod,t} + (1-\alpha)S_{t-1} \quad (13)$$

Де S_t– smoothed value, a α – smoothing coefficient (0 < α < 1).).

To detect long-term patterns of anomalous activity, a cumulative deviation indicator is introduced:

$$\mathrm{C}_t = \sum_{k=1}^{t} e^{-\lambda\kappa} |Z_{mod,k}| \quad (14)$$

After determining the normal limits of device behavior, the process of anomaly detection is carried out. The input data for this stage are the calculated normal values of the features and their characteristics. The output data is a list of devices that demonstrate a significant deviation from normal behavior. The abnormal deviation is calculated as:

$$A_{final} = \sum_{i=1}^{n} W_i * |Z_{mod\ i}| \quad (15)$$

This technique allows for more accurate detection of long-term anomalies and minimizes the impact of single spikes, providing a more accurate analysis of network traffic of IoT devices.To evaluate the proposed method for detecting anomalies in device traffic, an activity log generation mechanism was developed. It is based on the use of distributions of input device attributes, such as timestamps, login attempts, network addresses, connection errors, etc. The generation took place with a random deviation of parameters within the normal range of values for each device. This allows us to test the robustness of our approach in real scenarios of IoT systems.

Based on the input data, frequency distributions of the main parameters of the devices were determined, and then artificial anomalies were created with a certain probability. To assess the effectiveness of the z-index-adjusted method, anomalous values were added to the base logs, after which the accuracy of their detection was analyzed.

he generated logs underwent a correlation assessment stage between parameters to determine the degree of dependence between device attributes. Spearman's correlation coefficient was used for this, as it effectively captures non-linear relationships between variables. Additionally, independent anomalies were created in the form of random deviations in the data, allowing for a more comprehensive evaluation of the method’s robustness against false positives and its sensitivity to different types of deviations.

Generating a device activity log consists of several key steps. First, devices are selected from the total set according to the initial statistical distributions of their activity, ensuring that the dataset reflects real-world variability in network behavior. Then, activity timestamps for each device are defined and generated using a normally distributed time series, establishing a structured baseline for expected device activity. Random deviations are introduced according to the confidence interval parameter defined by the Modified Z-score, simulating fluctuations that naturally occur in network traffic.

A set of device attributes, such as login attempts, network connections, and error messages, is identified as these parameters often serve as key indicators of anomalous behavior. A correlation analysis is conducted to determine interdependencies between parameters, allowing structured patterns to be generated for further anomaly detection and classification. Additionally, control anomalies are introduced as values that fall outside the normal limits established using the Modified Z-score. This controlled injection of anomalies enables the assessment of the method's effectiveness compared to traditional detection approaches, ensuring its reliability in real-world cybersecurity scenarios. This approach ensures a systematic and statistically sound method for evaluating anomaly detection algorithms, highlighting the strengths and weaknesses of the applied techniques while maintaining a realistic representation of device activity in IoT environments.

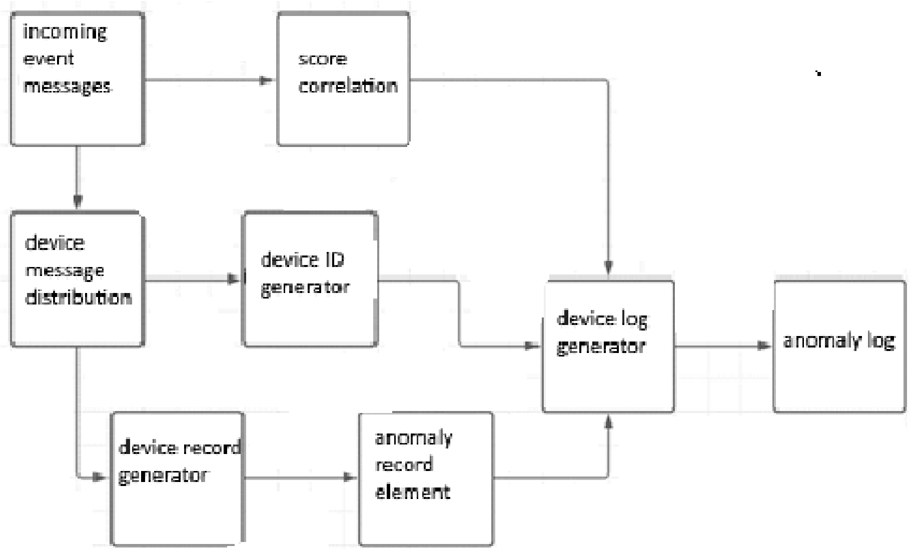

Figure 1. Scheme of processing and recording anomalies in the log.

The process of generating IoT device activity logs is performed taking into account the correlation characteristics of the data and modeling anomalous patterns. For this, the adjusted Z-index method is used, which allows you to effectively identify significant deviations in the behavior of devices. The generated log includes both normal data and controlled anomalies that are used to test the accuracy of detecting non-standard behavior.

In the process of generating an activity log, test anomalies are created for each device that correspond to different behavior parameters. Among them, several main categories can be distinguished.

An unusual number of device messages is formed by changing the data exchange rate. It can either increase, which may indicate a potential attack, or decrease, which may indicate technical failures or interference in the network.

An unusual number of authorization attempts determines possible security violations. An increased number of unsuccessful logins may indicate a hacking attempt or problems with authentication settings.

An unusual series of messages arriving in quick succession is an important indicator of anomalous behavior. If a device transmits many similar requests without interruption, this may be the result of a software failure or system overload.

Unusual time intervals between messages allow you to identify anomalies associated with changes in the device's interaction with the network. For example, unexpected delays or, conversely, too high a transmission rate can be signs of malfunctions.

An unusual number of connection errors is an indicator of malfunctions in the device or its interaction with the network. A high error rate can be associated with network instability, hardware problems, or attempts at unauthorized intervention.

The process of generating anomalies allows you to test the effectiveness of the method by determining its ability to detect different types of deviations in device behavior.

After generating the activity log, device profiles were formed, normal behavior limits were determined, and deviation analysis was performed. The total number of unique devices in the test log was 4804.

The quantitative assessment of the detected anomalies was carried out by comparing the found anomalies with the total number of injected deviations. The average level of anomaly detection accuracy was 79 percent.

The method best detected anomalies associated with changes in the time intervals between device messages. Detecting such deviations is effective because the time patterns are well modeled and have a clear structure that allows you to track changes.

4. Evaluation of the effectiveness of the method

The effectiveness of the proposed approach is confirmed by numerical calculations, which demonstrate higher accuracy and stability compared to classical statistical methods. The use of a modified Z-estimator allows reducing the impact of single outliers, ensuring resistance to sudden changes in traffic. Sensitivity of the method (Recall) = 89.96% indicates its ability to effectively detect real anomalies, while the accuracy (Precision) = 86.72% confirms a low proportion of false positives. Thanks to this, the method not only improves the quality of detection, but also adapts to different scenarios of IoT devices, reducing dependence on rigidly set threshold values.

$$Precision = \frac{TP}{TP + FP} * 100\% \tag{16}$$

$$Precision = \frac{242}{242 + 37} * 100\% = 86.72 \tag{17}$$

Where TP (True Positives) – true positive detections of anomalies, FP (False Positives) – false positives. The experimental results showed that the average accuracy value for all features is 86.72%, which indicates a low proportion of false positives and high reliability of identified anomalies. The sensitivity of our method is 89.96%, which means that the method effectively finds most of the real anomalies, minimizing their omission.

$$Recall = \frac{TP}{TP + FN} * 100\% \tag{18}$$

$$Recall = \frac{242}{242+27} * 100\% = 89.96\% \tag{19}$$

For a comprehensive assessment of the effectiveness of the anomaly detection method, the F1-score is used, which is the harmonic mean between accuracy (Precision) and sensitivity (Recall). This metric allows us to assess the balance between correct anomaly detection and minimizing false positives. A high F1-score indicates the stability of the algorithm and its ability to provide reliable identification of anomalous events without an excessive number of false positives.

The mathematical justification and calculation of the F1-score for our method are presented below.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{20}$$

$$F1 = 2 * \frac{86.72 * 89.96}{86.72 + 89.96} = 88.32\% \tag{21}$$

To assess the effectiveness of the proposed method, an analysis of the main metrics characterizing its ability to detect anomalies in the traffic of IoT devices was conducted. The table shows quantitative indicators, which include the total number of anomalies in the generated log (AGJ), the number of actually detected anomalies (DA), as well as the distribution of true positive (TP), false negative (FN) and false positive (FP) cases. The key metric for assessing the accuracy of the algorithm is Precision, which determines the proportion of correctly detected anomalies among all detections. Additionally, the ratio of TP to (TP + FN) allows us to assess the sensitivity (Recall), which determines the ability of the algorithm to detect true anomalies without missing them. For complex analysis, the F1-score is used, which is the harmonic mean between precision and sensitivity, which gives a balanced assessment of the detection quality.

Table 1.
Quantification of detected anomalies in IoT device traffic

Feature	AGJ	*DA*	TP	FN	FP	Presision
MF	82	763	71	6	5	87%
LI (	31	48	30	1	17	83%
MR	41	34	29	8	5	69%
MI	81	88	75	6	7	91%
EC	47	42	37	6	5	75%
TC	282	288	242	27	37	84%

As can be seen from Table 1, the method demonstrates the highest accuracy for anomaly detection in the message interval (MI) category, confirming its effectiveness in detecting deviations in temporal patterns. The lowest accuracy is observed in the analysis of repeated messages (MR), indicating the potential need for additional model optimization for this type of anomaly. The overall accuracy of the algorithm is 84%, indicating its stability and efficiency compared to traditional statistical approaches.

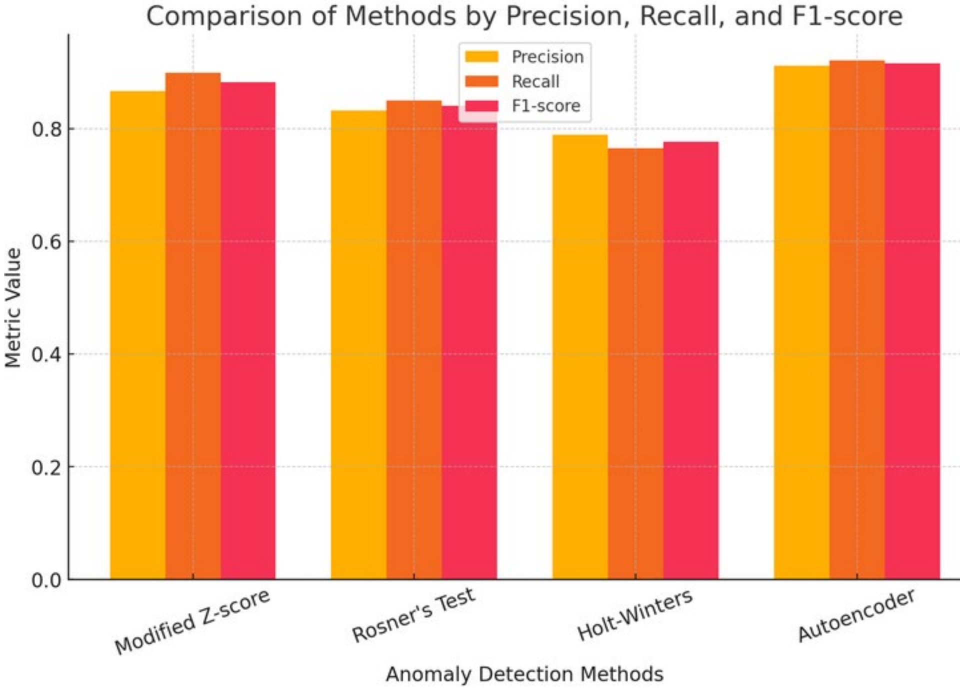

Figure 2. Comparison of Anomaly Detection Methods Based on Precision, Recall, and F1-score.

The graph shows a comparison of anomaly detection methods based on the Precision, Recall, and F1-score metrics. Autoencoder shows the best results, indicating its ability to accurately and completely identify anomalies.

Modified Z-score and Rosner's Test have similar performance, but are slightly inferior to Autoencoder.

Holt-Winters shows the worst metrics, indicating its less effective recognition of anomalous patterns in the data.

The graph illustrates the detection of anomalies in the outgoing traffic of the used IoT devices.

Normal traffic is indicated by a dotted line, and the red line shows anomalous traffic. Black crosses indicate isolated anomalous events that differ significantly from the average level of activity. Spikes and drops in intensity indicate possible attacks or atypical network behavior that requires detailed analysis.

The graph shows the distribution of IoT device traffic parameters. Yellow represents normal values, orange represents potential anomalies, and pink represents stable data.

It can be seen that parameter 2 (orange) has a wide spread, which may indicate the presence of anomalous values, while parameter 3 (pink) is more stable and concentrated. Such analysis allows you to better detect deviations in traffic and assess the likelihood of attacks.

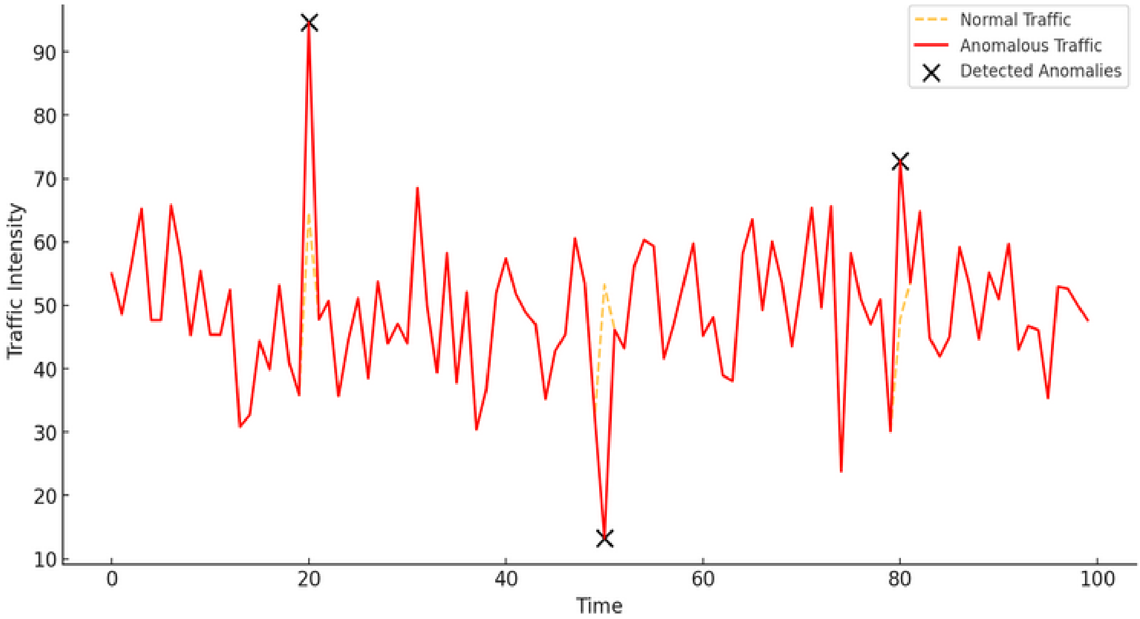

Figure 3. Anomaly Detection in IoT Device Traffic Over Time.

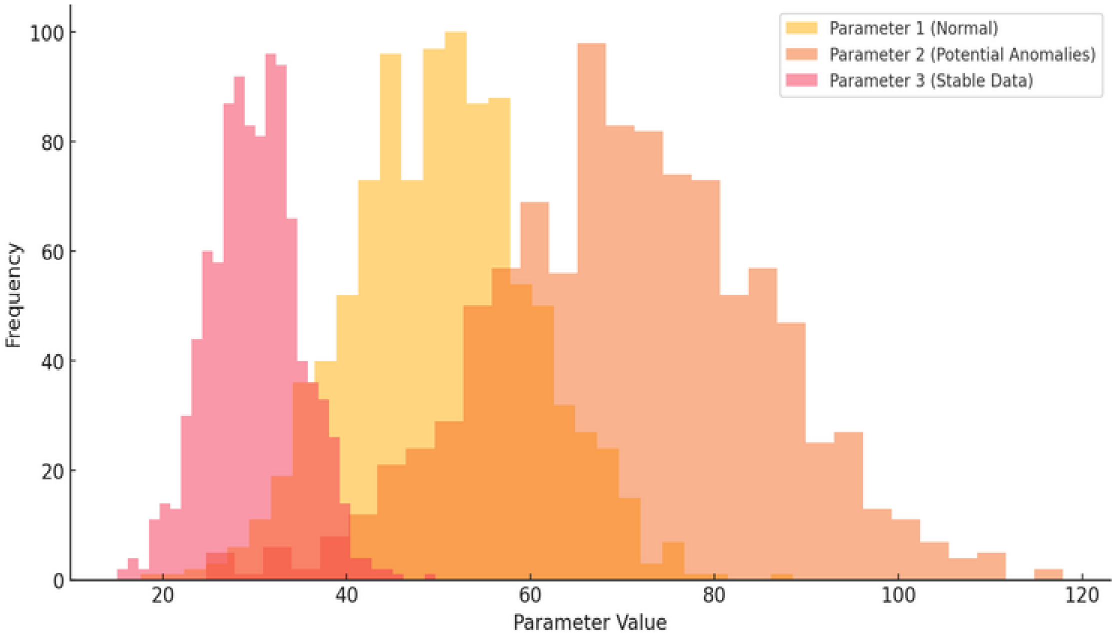

Figure 4. Distribution of IoT Device Traffic Parameters.

The boxplot illustrates an outlier analysis of IoT device traffic parameters. It can be seen that parameter 2 has the largest spread and number of outliers, indicating potential anomalies. Parameter 1 also contains several outliers, while parameter 3 is the most stable.

Such an analysis allows us to assess the distribution of values and detect atypical behavior of network traffic.

The method performed worst in detecting anomalies associated with the number of repeated messages. This may be due to the fact that small changes in the frequency of messages fell within the range of normal values, which required a more precise adjustment of the boundaries of normal behavior.

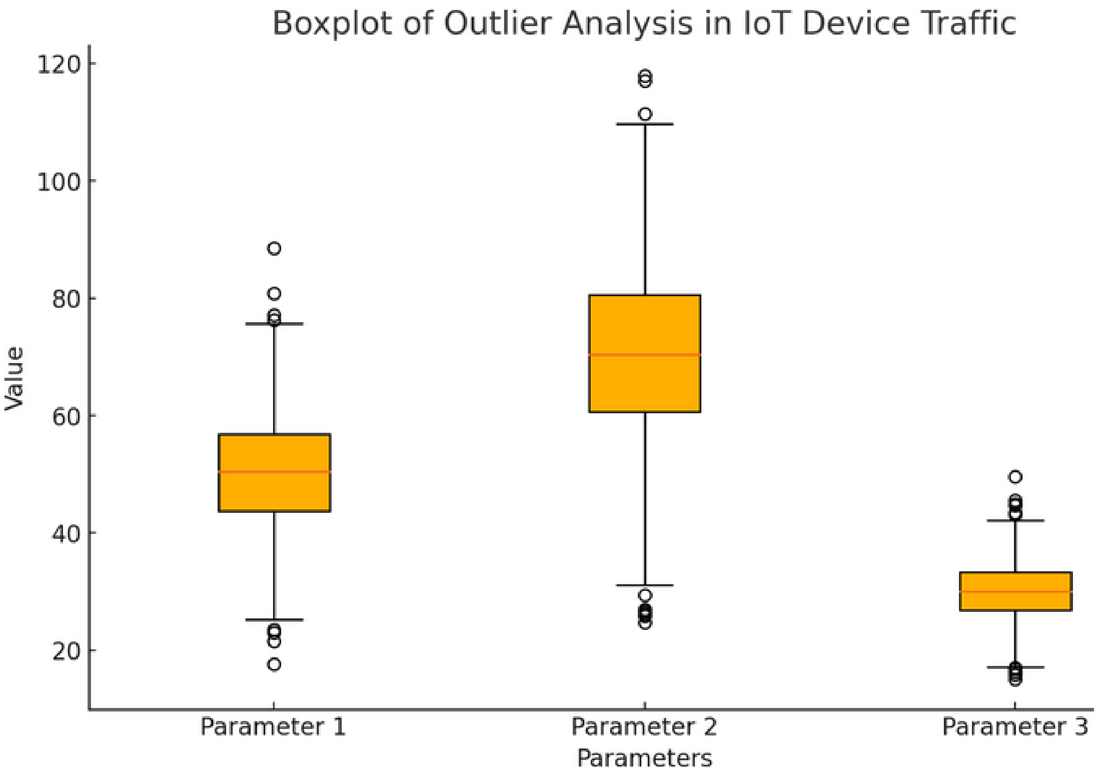

Figure 5. Boxplot of Outlier Analysis in IoT Device Traffic.

5. Conclusions

This paper proposes an approach to detecting anomalies in the traffic of Internet of Things devices based on statistical methods. The method includes constructing numerical characteristics of device profiles, determining their normal values, and searching for deviations from the norm.

The modified Z-estimator, Rosner test, and Holt-Winters method were used for analysis, which allow detecting both single and multiple anomalies, as well as analyzing time series. These methods allow for effective detection of deviations without the need for training on labeled data, which is a key advantage in the IoT environment.

Numerical evaluation of the approach's effectiveness showed that the average anomaly detection accuracy for all features was 84%, with the best results (91%) obtained for the analysis of message intervals. The method provides high computational speed and can be used in environments with limited computing resources.

The advantages of the proposed approach include its adaptability to different types of devices, the ability to work with unlabeled data, and the minimal computing power requirements.

The main direction of further research is to expand the set of features and compare the effectiveness of different statistical methods for anomaly detection in IoT networks.

Declaration on Generative AI

AI tools were used solely as translation and proofreading aids. All content was originally authored by the submitting party.

References

[1] B. Charyyev, M.H. Gunes. Detecting Anomalous IoT Traffic Flow with Locality-Sensitive Hashes. IEEE, Taiwan, 2020. DOI:10.1109/GLOBECOM42002.2020.9322559.

[2] H. Nguyen-An, T. Silverston, T. Yamazaki, T. Miyoshi. IoT Traffic: Modeling and Measurement Experiments. IoT, 2 (2021), 140–162. DOI:10.3390/iot2010008.

[3] A. Nicheporuk, O. Dariychuk, S. Danchuk. Model of Process for Ensuring Fault Tolerance in Internet of Things Networks. Computer Systems and Information Technologies, 2 (2024), 14–20. DOI: 10.31891/csit-2024-2-2.

[4] D. Garlisi, A. Martino, J. Zouwajhed, J. Pourahmiri, F. Cuomo. Exploratory approach for network behavior clustering in LoRaWAN. Journal of Ambient Intelligence and Humanized Computing, 2021. DOI:10.1007/s12652-021-03121-z.

[5] Z. Li, F. Farid, A. Bello, F. Sabrina. Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis. Cybersecurity, 2024. DOI:10.1186/s42400-024-0023-8.

[6] H. Ochiai, T. Nishidate, E. Tomiyama, Y. Sun, H. Esaki. Detection of Global Anomalies on Distributed IoT Edges with Device-to-Device Communication. arXiv, 2024. DOI:10.48550/arXiv.2407.11308.

[7] M. Liu, L. Yao. IoT Network Traffic Analysis with Deep Learning. arXiv, 2024. DOI:10.48550/arXiv.2404.04964.

[8] S.G. Chubchenko, E.S. Koch, M.C. Moura-dos-Santos, R.L. Mota, D.M. Vieira, L.C. Andrade, D.R. Araújo. Anomaly Detection in Industrial Machinery using IoT Devices and Machine Learning: a Systematic Mapping. IEEE Access, 2023. DOI:10.1109/ACCESS.2023.0320007.

[9] Y. Klots, N. Petliak, S. Martsenko, V. Tymoshchuk, I. Bondarenko. Machine Learning system for detecting malicious traffic generated by IoT devices. in: 2nd International Workshop on Computer Information Technologies in Industry 4.0, CITI 2024, volume 3742, 2024, pp. 97–110.

[10] M. Stetsyuk, V. Cheshun, Y. Stetsyuk, O. Kozelskiy, A.-B.M. Salem. A model of a DDoS attack scenario on elements of specialized information technology and methods of combating cybercriminals. in: Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelIITSIS 2024), CEUR Workshop Proceedings, vol. 3675, Khmelnytskyi, Ukraine, 28 March 2024, pp. 260–269. ISSN 1613-0073.

[11] A.K. Jain, H. Shukla, D. Goel. A comprehensive survey on DDoS detection, mitigation, and defense strategies in software-defined networks. Cluster Comput., June 2024. DOI:10.1007/s10586-024-04966-z.

[12] A. Girma, M.A. Guo, J. Irungu. Identifying Shared Security Vulnerabilities and Mitigation Strategies at the Intersection of APIs, Application Level and OS of Mobile Devices. in: Arai, K. (Eds.), Proceedings of the Future Technologies Conference (FTC), volume 2 of FTC 2022. Lecture Notes in Networks and Systems, vol. 560, Springer, Cham, 2023, pp. 1–10. DOI:10.1007/978-3-031-18458-1_34.

[13] N.A.A.H. Al-Sarray, S. Demir. A Cybersecurity Procedure to Vulnerabilities Classification of Windows OS Based on Feature Selection and Machine Learning. in: Rasheed, J., Abu-Mahfouz, A.M., Fahim, M. (Eds.), Forthcoming Networks and Sustainability in the AIoT Era - FoNeS-AIoT 2024. Lecture Notes in Networks and Systems, vol. 1035, Springer, Cham, June 2024, pp. 18–29. DOI:10.1007/978-3-031-62871-9_18.

[14] N. Rai, J. Grover. Analysis of crypto module in RIOT OS using Frama-C. J. Supercomput., 80 (2024), 18521–18543. DOI:10.1007/s11227-024-06171-0.

[15] H. Dymova. Study of Cryptographic Security of Computer Networks. Computer-integrated technologies: education, science, production, 15 (2025), 15–19. DOI:10.36910/6775-2524-0560-2024-57-02.

[16] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi. Internet of Things security: A survey. Journal of Network and Computer Applications, 88 (2017), 10–28. DOI:10.1016/j.jnca.2017.04.002.

[17] S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini. Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76 (2015), 146–164. DOI:10.1016/j.comnet.2014.11.008.

[18] R. Roman, P. Najera, J. Lopez. Securing the Internet of Things. Computer, 44 (2011), 51–58. DOI:10.1109/MC.2011.291.

[19] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, D. Qiu. Security of the Internet of Things: Perspectives and challenges. Wireless Networks, 20 (2014), 2481–2501. DOI:10.1007/s11276-014-0761-7.

[20] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao. A survey on security and privacy issues in Internet-of-Things. IEEE Internet of Things Journal, 4 (2017), 1250–1258. DOI:10.1109/JIOT.2017.2694844.

[21] J. Granjal, E. Monteiro, J.S. Silva. Security for the Internet of Things: A survey of existing protocols and open research issues. IEEE Communications Surveys & Tutorials, 17 (2015), 1294–1312. DOI:10.1109/COMST.2015.2388550.

[22] R.H. Weber. Internet of Things – New security and privacy challenges. Computer Law & Security Review, 26 (2010), 23–30. DOI:10.1016/j.clsr.2009.11.008.

[23] A. Mosenia, N.K. Jha. A comprehensive study of security of Internet-of-Things. IEEE Transactions on Emerging Topics in Computing, 5 (2017), 586–602. DOI:10.1109/TETC.2016.2606384.

[24] M.A. Khan, K. Salah. IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82 (2018), 395–411. DOI:10.1016/j.future.2017.11.022.

[25] O. Savenko, S. Lysenko, A. Kryschuk, Multi-agent based approach of botnet detection in computer systems. In: Communications in Computer and Information Science 291 (2012) 171–180. https://doi.org/10.1007/978-3-642-31217-5_19.