Hybrid system for detecting abnormal traffic in IoT*

Nataliia Petliak[1,†], Yurii Klots[1,*,†], Mikolaj Karpinski[2,3,†], Vira Titova[1,†] and Dmytro Tymoshchuk[2,*,†]

[1] *Khmelnytskyi National University, 11, Instytuts'ka str., Khmelnytskyi, 29016, Ukraine*
[2] *Ternopil Ivan Puluj National Technical University, 56, Ruska str. Ternopil, 46001, Ukraine*
[3] *University of the National Education Commission, 2 Podchorążych str, Krakow, 30084, Poland*

Abstract

The paper proposes a hybrid method for detecting abnormal traffic based on a step-by-step combination of signature classification by features, self-similarity analysis, and fuzzy inference. The proposed model formalizes network traffic in the form of signatures, which allows for accurate packet classification in real time. Attention is paid to optimizing the detection process by using the Hurst coefficient to assess self-similarity and fuzzy decision-making systems for processing uncertain data. Experimental studies were conducted on the basis of two relevant datasets - WSN-DS and CICIoT2023. According to the test results, the proposed method has demonstrated high anomaly detection accuracy (over 99%), high completeness, and balanced accuracy, which indicates its effectiveness in practical conditions of IoT systems operation. In addition, the proposed approach can significantly reduce the network channel utilization (up to 6.28% on WSN-DS and up to 1.93% on CICIoT2023), which is accompanied by an increase in the CPU load, leaving an acceptable balance of resources for systems with limited capabilities.

Keywords

Internet of Things, intrusion detection, signature classification, self-similarity analysis, fuzzy inference, cybersecurity

1. Introduction

The Internet of Things (IoT) is a distributed information and communication system (ICS) that encompasses billions of devices connected to a global network that can automatically collect, analyze, and exchange data. Thanks to the active development of network infrastructure and the reduction in the cost of hardware components, IoT is rapidly spreading in various areas of human activity, from everyday life to industry, creating the basis for smart ecosystems [1]. The basic idea of IoT is to integrate sensors, actuators, and software into everyday objects to improve their functionality. Smart homes, cities, transportation networks, medical facilities, agricultural complexes, and industrial enterprises are all now becoming part of a single information environment. For example, in agriculture, sensors monitor soil moisture and microclimate, in beekeeping, they track the movement of the bee colony and honey consumption in winter [2], in medicine, they monitor patients' vital signs, and in cities, IoT helps manage energy efficiency, transportation, and even garbage containers [3-4]. Biosensors integrated with IoT allow monitoring various physiological parameters [5-7]. IoT is also used to measure pollution concentrations, collect and process data in real time for quick decision-making by air monitoring systems [8]. Rapid growth makes IoT one of the main drivers of digital transformation. On the one hand, it opens up new opportunities for automation and efficiency, and on the other hand, it creates challenges in terms of security, privacy, and scalability [9].

* *CITI'2025: 3rd International Workshop on Computer Information Technologies in Industry 4.0, June 11–12, 2025, Ternopil, Ukraine*
[1*] Corresponding author.
[†] These authors contributed equally.
npetlyak@khmnu.edu.ua (N. Petliak); klots@khmnu.edu.ua (Y. Klots); mikolaj.karpinski@uken.krakow.pl (M. Karpinski); titovav@khmnu.edu.ua (V. Titova); dmytro.tymoshchuk@gmail.com (D. Tymoshchuk)
0000-0001-5971-4428 (N. Petliak); 0000-0002-3914-0989 (Y. Klots); 0000-0002-8846-332X (M. Karpinski); 0000-0001-8668-4834 (V. Titova); 0000-0003-0246-2236 (D. Tymoshchuk)

Cyberattacks on IoT systems are becoming increasingly large and complex, encompassing a wide range of methods that threaten the integrity, confidentiality, and availability of digital environments[10]. The most common are DDoS attacks, in which IoT devices become infected with malware and turn into botnets that can paralyze entire services by overloading network traffic[11-13]. Other common attack vectors include port scanning, brute force password guessing, SQL injections, cross-site scripting (XSS), and man-in-the-middle (MITM) attacks, which allow attackers to intercept or modify traffic between devices [14]. Hackers are also actively using unencrypted communication channels and typical password protection, which is a characteristic flaw of many IoT devices. Attacks on IoT applications can be aimed at blocking access to services, stealing sensitive data, or gaining control over computing resources. In a smart home, hackers can gain access to video surveillance cameras, locks, or thermostats, which creates not only digital but also physical risks. In smart cities, attacks can disrupt lighting, transportation, or environmental monitoring systems. Attackers are using modified versions of well-known botnets, such as Mirai or Mozi, which use automated vulnerability exploitation to create large botnets. The use of cryptomining malware is also common, where IoT devices are used to mine cryptocurrencies without the owners' knowledge. [15-16]

Intrusion Detection and Prevention Systems (IDS/IPS) are essential for securing IoT networks, as traditional security tools often fail to cope with the peculiarities of these environments – a large number of heterogeneous devices, limited computing resources, and high traffic dynamics. IDS/IPS analyzes network activity in real time, detecting and neutralizing anomalies, malicious activities, and unauthorized access attempts. There are two main types of IDSs: signature-based, which recognize attacks based on known patterns, and anomaly-based, which detect deviations from normal behavior, making them particularly useful for identifying zero-day attacks. In the IoT environment, machine learning and deep learning methods are increasingly used to allow such systems to adapt to new threats and improve detection accuracy, reducing the number of false positives [17-19]. IPS systems go even further, blocking threats in real time by changing access rules or terminating suspicious connections. Deep neural networks provide flexibility and scalability for such solutions, enabling them to analyze high-dimensional data typical of IoT networks. IDS/IPS detect a wide range of threats: from DDoS attacks and phishing to sophisticated botnets and malware intrusions [20-21]. However, despite their effectiveness, modern IDS/IPS systems have a number of significant limitations. First, deep learning models are resource-intensive and difficult to scale on devices with limited computing power. Secondly, they require complex configuration, constant updates, and skilled administration, which is not always feasible in a large-scale IoT environment. In addition, the problem of data imbalance and the need for high-quality samples for model training remain open issues. Given these shortcomings, it is becoming increasingly important to explore alternative solutions that are lighter, more energy-efficient, easier to maintain, and less dependent on complex architectures that can provide a basic but reliable level of protection without overloading the system [22].

Section 2 provides an overview of related work. Section 3 describes the models used for traffic classification and anomaly detection. Section 4 presents the proposed hybrid method combining signature classification, self-similarity analysis, and fuzzy inference. Section 5 presents experimental results and discussion, including performance evaluation on WSN-DS and CICIoT2023 datasets. Finally, Section 6 summarizes the paper.

2. Related works

The authors of [23] propose a comprehensive solution for detecting and mitigating DDoS attacks in IoT networks by creating an intelligent system based on three algorithms: exponentially weighted moving average, cumulative sum algorithm, and K-nearest neighbors method. The combination of these methods allows to increase the accuracy of anomaly detection, reduce the frequency of false positives, and compensate for the shortcomings of each individual approach due to the backup principle of decision-making. Paper [24] proposes a method for detecting DDoS and DoS attacks in

IoT networks by using a conditional tabular generative adversarial network. The main idea is to create synthetic samples of network traffic that reliably imitate legitimate behavior patterns and use this data to train machine and deep learning algorithms. The proposed IDS combines the capabilities of generative modeling and discriminative analysis. Publication [25] describes a method for detecting DoS and DDoS attacks based on deep learning using an improved architecture of the convolutional neural network DCBAM (DenseNet with Block Attention). The method covers four main stages: data preprocessing, data augmentation using conditional GAN for class alignment, deep feature extraction using pyramidal attentional attention, and selection of the most significant features.

Paper [26] proposes an LPCOCN approach for detecting anomalies at the edge of IoT networks, which combines efficient feature selection, their conversion to a visual format, and classification using a capsule neural network. The system is based on four key modules: feature selection using a multi-layer LPCO algorithm, image formation from selected features using Manhattan distance, model training using a capsule network, and final anomaly detection. To evaluate the proposed approach, five network traffic datasets are used, including NSL KDD, UNSW NB, CICIDS, CSE-CIC-IDS, and UNSW Bot-IoT. The authors of [27] propose a comprehensive approach to detecting attacks in IoT networks based on the analysis of the full BoT-IoT dataset, combining machine learning, big data processing strategies (Hadoop-Spark), and advanced methods of synthetic data generation. Paper [28] investigates the effectiveness of the k-nearest neighbors algorithm for detecting botnet attacks in IoT networks, focusing on overcoming its limitations when working with large amounts of data. To improve the classification accuracy in a large set of Bot-IoTs, the researcher applies three feature selection strategies: information gain, direct selection, and backward elimination. Combining these techniques with the kNN algorithm allows identifying the most relevant attributes.

Paper [29] proposes an intelligent intrusion detection system for IoT networks that combines the use of a recorder, sensors, and artificial intelligence to improve cybersecurity. The authors also describe a structured methodology based on machine learning, including the use of five different classifiers to analyze network traffic and identify potential threats. Paper [30] proposes a method for detecting attacks in the IoT based on the architecture of a double convolutional neural network (CNN-CNN). The peculiarity of the approach is the use of two interconnected CNN models: the first one performs an automated selection of the most informative features from raw network traffic, and the second one classifies traffic into normal and malicious. This approach overcomes the limitations of traditional methods due to the ability of deep learning to detect hidden patterns and correlations in data that go unnoticed by other techniques. Article [31] is devoted to a framework for detecting botnets in IoT networks, combining the capabilities of graph neural networks with the temporal dynamics of IoT communications. The proposed Graph-based Gated Convolutional Network model uses a multi-graph structure with timestamps to accurately model the complex connections and traffic patterns inherent in botnet attacks. The method is based on a two-tier architecture that handles both spatial and temporal characteristics of the graph through message transmission, aggregation, and updating of node and edge embeddings using GRU. The authors of [32] propose a hybrid method for detecting botnet attacks in IoT networks by combining the SMOTE resampling method with a deep recurrent neural network to overcome the problem of severe data imbalance in a multi-class environment. By creating synthetic minority samples using SMOTE, the researcher ensures a balance between classes, which allows the DRNN model to train efficiently on a representative dataset. Thanks to its architecture, which includes a recurrent layer and several dense layers, the DRNN is able to study time dependencies in network traffic and detect hidden patterns of botnet activity.

Summarizing the results of the considered approaches to detecting attacks in IoT networks, we can conclude that they are highly effective in terms of classification accuracy, ability to adapt to complex traffic patterns, and detect anomalies in real time. The use of modern methods, such as CTGAN, GGCN, CNN-CNN, DRNN or SMOTE-DRNN, demonstrates significant potential for ensuring reliable cybersecurity in IoT environments. However, despite their technical advantages,

the implementation of these solutions is accompanied by a number of significant challenges. In particular, the complexity of the models often leads to high requirements for computing resources, long training time, and the need for qualified specialists to support, configure, and scale them. In addition, deploying such systems in a real-world IoT environment is technically challenging and requires careful integration with existing infrastructure. In this regard, it is important to search for alternative solutions that can strike a balance between the effectiveness of threat detection and the practicality of implementation - by simplifying models [33], reducing their power consumption, optimizing training algorithms, and developing more accessible tools.

3. Traffic classification model

This study proposes a formalized approach to representing incoming traffic in the form of signatures, which allows for accurate packet classification in ICS networks. Each signature is described by a multi-component tuple. Based on this representation, a hybrid method for detecting abnormal traffic has been developed, consisting of three blocks: signature classification by features, self-similarity analysis, and fuzzy inference.

One of the fragments of the proposed model is the formation of signatures that describe each individual network packet through a set of its features. The incoming traffic is represented as a set of signatures:

$$D=\bigcup_{i=0}^{N_d}\{d_i\} \quad (1)$$

where d_i – element of the set, input generated signature; i – initial value; N_d – number of elements of the input signatures set.

The packet signature will allow you to uniquely identify the source of the traffic that initiates it, determine the size of the packet and the time when it was analyzed.

The packet signature will be represented as a multi-component tuple:

$$d=\langle d^1, d^2, \ldots, d^k \rangle, \quad (2)$$

where k – number of components in the tuple.

For the case under consideration, let's assume $k=9$ and define these components: d^1 - source IP address; d^2 - destination IP address; d^3 - source port; d^4 - destination port; d^5 – protocol used for data transmission; d^6 – traffic intensity, defined in bits/s; d^7 - time of receipt of the package for inspection; d^8 – MAC address of the device that sends data from the network; d^9 – package size.

Each generated signature will belong to the set D:

$$\forall\, d_i \in D \quad (3)$$

The next part of the model is the process of classifying traffic by features to classify traffic into allowed, forbidden, and requiring further analysis using sets of allowed, forbidden, and undefined signatures.

Represent the set of allowed signatures as follows:

$$D^G=\bigcup_{i=0}^{N_g}\{d_i\} \quad (4)$$

where N_g – number of allowed signatures; d_i – signature assigned to the set of allowed signatures.

Let's represent the set of forbidden signatures as follows:

$$D^B = \bigcup_{i=0}^{N_b} \{d_i\} \quad (5)$$

where N_b – number of forbidden signatures; d_i – signature assigned to the set of forbidden signatures.

Represent the set of undefined signatures as follows:

$$D^U = \bigcup_{i=0}^{N_u} \{d_i\} \quad (6)$$

where N_u – number of undefined signatures; d_i – signature that is assigned to the set of undefined signatures.

The process of traffic classification based on signature analysis is represented as a partitioning of the set D:

$$D^G = \left\{ d_j \in D \middle| If\ \exists\, d_i \in D^G\ that\ d_i^1 = d_j^1 \wedge\ d_i^2 = d_j^2 \right\} \quad (7)$$

$$D^B = \left\{ d_j \in D \middle| If\ \exists\, d_i \in D^B\ that\ d_i^1 = d_j^1 \vee\ d_i^2 = d_j^2 \right\} \quad (8)$$

$$D^U = \left\{ d_j \in D \middle| If\ d_j \notin D^G \wedge\ d_j \notin D^B \right\} \quad (9)$$

where d_j – the signature of the incoming traffic being analyzed.

As sets D^G, D^B та D^U are partitions of D the following statements are true:

$$D = D^G \cup D^B \cup D^U \quad (10)$$

$$D^G \cap D^B = \varnothing,\ D^G \cap D^U = \varnothing,\ D^B \cap D^U = \varnothing \quad (11)$$

Next, we will describe a fragment of the traffic classification process based on self-similarity, which involves the use of a similarity metric between new signatures and previously verified ones that are part of the set of allowed ones. Self-similarity is an effective criterion that minimizes the need to use resource-intensive fuzzy methods when evaluating unknown traffic patterns. Its advantage lies in the ability to identify stable patterns of behavior in traffic even when it varies in the time dimension. By analyzing consecutive time intervals, it is possible to verify new signatures based on previously obtained data, providing a dynamic update of the authorized traffic profile without the need to completely review the accumulated records.

To quantify the degree of self-similarity, the Hearst coefficient is used to determine the level of correlation between the characteristics of signatures. This ensures high accuracy in classifying new packets - whether they belong to legitimate traffic or require more detailed inspection. This approach is important for a timely response to potential cyber threats, as it provides an effective balance between processing speed and detection accuracy. Thanks to the self-similarity property, the amount of traffic that remains undetected is significantly reduced, thereby optimizing the monitoring process and increasing the overall efficiency of the anomaly detection system.

To take into account traffic self-similarity, we define the following time parameters (Fig. 1) $t_1 - t_3$ – the time interval for analyzing signatures for self-similarity, consisting of two sub-intervals, $t_1 - t_2$ – the time interval of signatures previously analyzed for self-similarity, $t_2 - t_3$ – the time interval of new signatures.

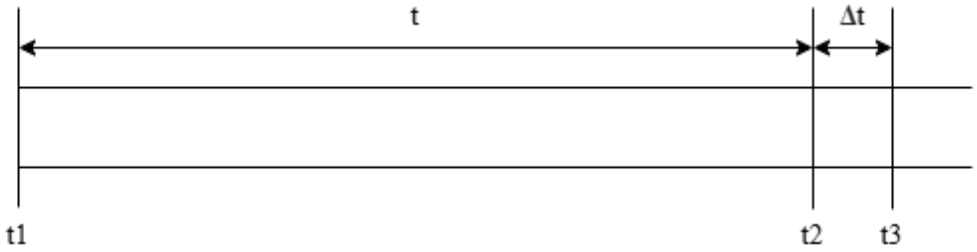

Figure 1: Time intervals for traffic classification

Define the set M of signatures previously analyzed for self-similarity:

$$M=\left\{d_i \in D^G \cup D^U \mid t_1<d_i^7 \le t_2\right\} \tag{12}$$

and a set M' of new signatures that have not been analyzed for self-similarity before:

$$M'=\left\{d_i \in D^G \cup D^U \mid t_2<d_i^7 \le t_3\right\}, \tag{13}$$

where d_i^7 – time of receipt of the package for inspection.

Define the similarity function:

$$y=f(T), \tag{14}$$

where T – signature set, $T \subset \mathrm{D}$; y – numerical similarity measure, $y \in [0,1]$.

To determine the similarity of a packet signature to a set of previously defined packet signatures, we define the function $S(n)$ of the standard deviation.

$$S^j(n)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i^j-\overline{d^j})^2}, \tag{15}$$

where $n=|T|$ – number of signatures in a set T; j – component number in the tuple; d_i^j – is the value of the j-th component of the i-th tuple of the set; $\overline{d^j}$ – average value of the j-th component of tuples of the set.

And the function $R(n)$ range of the accumulated deviation of the function:

$$R^j(n)=\max_{i=1,n}\sum_{m=1}^{k} d_i^j-\overline{d^j}-\min_{i=1,n}\sum_{m=1}^{k} d_i^j-\overline{d^j},\ k=1,n \tag{16}$$

Determine the Hurst coefficient for each component of the tuple:

$$H^j=\frac{\ln\left(\frac{R^j(n)}{S^j(n)}\right)}{\ln(n)} \tag{17}$$

Define the Hurst coefficient for the packet signature as the average of the Hurst coefficients for each component of the tuple:

$$H=f(T)=\frac{1}{k}\sum_{j=1}^{k} H^j \tag{18}$$

where j – number of components in the tuple.

The process of classifying undefined traffic based on signature similarity analysis is presented as follows:

$$D^G = \left\{ D^G \cup M^{'} \middle| f\left(M \cup M^{'}\right) > 0{,}5 \right\}, \quad (19)$$

$$D^U = \left\{ D^U \setminus M^{'} \middle| f\left(M \cup M^{'}\right) > 0{,}5 \right\}, \quad (20)$$

After the analysis is completed, the set D^U will contain signatures that are undefined or different from the allowed signatures and require further analysis.

To unambiguously classify the signatures left over from the previous stages, we will use the process of fuzzy detection of abnormal traffic.

The structural organization of the fuzzy inference system for classifying the features of signature elements involves the integration of the following main components: fuzzification, membership functions, fuzzy inference mechanism, and defuzzification. The system's workflow is implemented in the following sequence: the initial input data X is sent to the fuzzification module, where it is transformed into fuzzy values $X^{'}$ by applying the appropriate membership functions. The next stage involves performing fuzzy logical inference, which involves implication, aggregation, activation, and accumulation based on a given base of fuzzy rules, resulting in the output fuzzy data $Y^{'}$. At the final stage, the defuzzification procedure transitions from fuzzy results $Y^{'}$ to a clear output value Y, which serves as the basis for the final decision. The process of fuzzification provides transformation of deterministic input data into fuzzy terms to account for possible undefined uncertainties in the input characteristics. For example, if the parameter d^6 is 10,000 bits/s, its intensity will be classified as “above average” using the membership function. The fuzzy inference algorithm is implemented through a sequence of four core operations. At the implication stage, the input fuzzy values $X^{'}$ are transformed into the corresponding output fuzzy sets. Aggregation combines the results of implication obtained from different rules for each term of the output variable. The activation stage evaluates the degree of truth of each rule, usually using minimization methods. Accumulation performs the integration of the obtained initial fuzzy sets into a single set $Y^{'}$. Defuzzification is responsible for converting the obtained fuzzy conclusion into a clear value of Y, which is necessary for making a specific decision. As a result of the system's operation, the output variable result is formed, which takes on one of two intervals: “allowed” [0, 50] or ‘forbidden’ [50, 100]. If the signature is classified as “allowed”, the data is transmitted and the signature is registered in the database of allowed connections. If the signature is identified as “ forbidden”, the system blocks the connection by the source IP and MAC addresses, and the corresponding signature is added to the register of forbidden connections.

Membership functions determine the degree to which each input value belongs to certain terms. Define a linguistic variable ld, that takes one of the values from the term set $ld \in \left\{ ld^1, ld^2, \ldots, ld^m \right\}$, where m is the number of terms. Form the linguistic variables used to analyze network traffic and corresponding to the components of tuple (2): $ld^1, ld^2, \ldots, ld^k$. It should be noted that the model of the process of fuzzy detection of abnormal traffic does not use the first component of the tuple, the source IP address (d^1), the third - the source port (d^3), seventh - time of receipt of data for verification (d^7) and the eighth - the MAC address of the device (d^8). Since the source IP address $\left(d^1\right)$, which determines the device that initiates the connection and transmits the data, is not a reliable indicator for analysis, as it can change due to the use of dynamic IP, NAT, or proxy servers, as well as be subject to spoofing. Source port $\left(d^3\right)$, The source port, which identifies the incoming point of communication on the sender's side, is usually randomly generated by the operating system, making it unpredictable and of little importance in the context of fuzzy anomaly classification. Packet arrival time $\left(d^7\right)$, expressed in a 24-hour format, depends on many external

factors, such as network delays, peak loads, and changes in traffic depending on the time of day. This creates a significant level of variability, which can make it difficult to correctly fuzzify abnormal patterns. MAC address of the device (d^8), that sends the data is also not used due to its local nature, as this address is not transmitted through routed networks and can be easily spoofed or changed at the device level. The exclusion of these parameters is aimed at improving the accuracy of the model by eliminating variables that may introduce unnecessary noise. Thus, only those characteristics that provide stability and high differentiation between normal and potentially threatening traffic remain for fuzzy analysis.

The second component in the tuple is the destination IP address (d^2), which can be displayed as a character variable that takes one of the values of the set of identifiers $ld^2 \in \left\{ld_1^2, ld_2^2, ld_3^2\right\}$, namely: ld_1^2=«IP address of the allowed traffic»; ld_2^2=«IP address of unspecified traffic»; ld_3^2=«IP address of abnormal traffic».

The fourth component in the tuple is the destination port (d^4), which can be displayed as a character variable that takes one of the values of the set of identifiers $ld^4 \in \left\{ld_1^4, ld_2^4, ld_3^4\right\}, namely:$ ld_1^4=«port of normal typical traffic»; ld_2^4=«port of undefined traffic»; ld_3^4=«port of abnormal traffic».

The fifth component in the tuple is the protocol (d^5), which can be displayed as a character variable that takes one of the values of the set of identifiers $ld^5 \in \left\{ld_1^5, ld_2^5, ld_3^5\right\}$, namely: ld_1^5 =«secure protocols»; ld_2^5=«neutral protocols»; ld_3^5=«undefined protocols»; ld_4^5=«suspicious protocols»; ld_5^5=«dangerous protocols».

Traffic intensity in the ICS (d^6) can be considered from two approaches: objective and subjective. Objective traffic intensity is defined as the ratio of the amount of information transmitted through the communication channel for a certain period of time to the total available network resource. This approach is based on measuring and analyzing the actual performance of the system, for example, the number of packets or bytes passing through the system per unit of time. Subjective traffic intensity reflects the assessment of experts or users of the system regarding the level of congestion of communication channels at a certain moment or during a certain period. It is based on perceptions or estimates of system performance, as well as on forecasts of system performance depending on the current or expected volume of information transmitted. In situations where obtaining accurate statistics is difficult, experts use a logical-linguistic approach. In this case, the traffic intensity is represented by a linguistic variable (ld^6) with a basic term set, which allows formalizing qualitative assessments and ensuring their convenient interpretation for further analysis:

$$ld^6 = \cup_{i=1}^{m} ld_i^6 \tag{21}$$

where m – the number of terms for which the order ratio is valid $ld_1^6 < ld_2^6 < \ldots < ld_m^6$.

For example, when m=5 for the specified linguistic variable, you can form sets of terms

$ld^6 = \cup_{i=1}^{5} ld_i^6 = \{«low», «below\ average», «average», «above\ average», «high»\}$,

which is represented by fuzzy numbers with corresponding membership functions. You can also enter other values of the terms, such as “very low (VL)”, “very high (HV)”, etc. The ninth component in the tuple is the package size (d^9). This component can be represented numerically or through a linguistic variable:

$$ld^9 = \cup_{i=1}^{y} ld_i^9 \tag{22}$$

where y – the number of terms for which the order ratio is valid $ld_1^9 < l\,d_2^9 < \ldots < ld_y^9$.

For all fuzzy sets that are used, the membership function is defined as the intersection of all sets:

$$\mu_{M_{d^2}} \cap \mu_{M_{d^4}} \cap \mu_{M_{d^5}} \cap \mu_{M_{d^6}} \cap \mu_{M_{d^9}} = \left(\mu_{M_{d^2}}(x), \mu_{M_{d^4}}(x), \mu_{M_{d^5}}(x), \mu_{M_{d^6}}(x), \mu_{M_{d^9}}(x)\right), \quad (23)$$

where $M_{d^2} = \left\{m_{d^2 i}\right\}_{i=0}^{4,3 \cdot 10^9}$ – he set of possible values of the destination IP address; $M_{d^4} = \left\{m_{d^4 i}\right\}_{i=0}^{65535}$ - the set of possible values of destination ports; $M_{d^5} = \left\{m_{d^5 i}\right\}_{i=0}^{36}$ - the set of possible protocols; $M_{d^6} = \left\{m_{d^6 i}\right\}_{i=0}^{10 \cdot 10^9}$ - the set of possible values of traffic intensity; $M_{d^9} = \left\{m_{d^9 i}\right\}_{i=0}^{65535}$ - a set of possible values of packet size.

A fuzzy knowledge base contains a set of fuzzy rules that define the relationships between input and output variables. Each rule has the form “If X, then Y”.

4. Hybrid method for detecting abnormal traffic

We propose a hybrid method for detecting abnormal traffic (Fig. 2), which combines the advantages of several approaches – signature classification, self-similarity analysis, and fuzzy inference. Such a multi-level structure provides more accurate, reliable, and adaptive anomaly detection, reducing the number of undefined cases and optimizing security system resources.

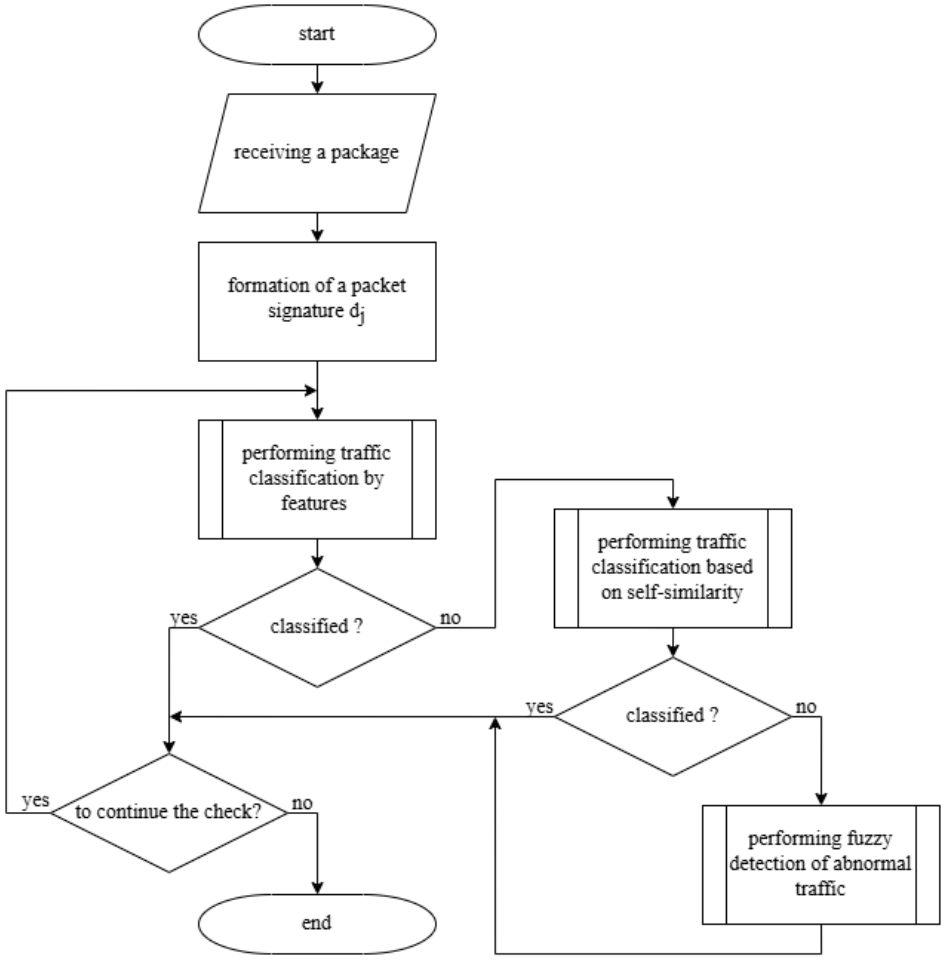

Figure 2: Algorithm of the hybrid method for detecting abnormalities in ICS

The input data are: packets from the original data stream, a set of original signatures (D), the set of allowed signatures D^G, a set of forbidden signatures D^B, a set of undefined signatures D^U, time intervals of the reference sample and the verification sample, a list of rules. The steps of the method are as follows.

Step 1. Connect the file with the set of allowed signatures (D^G).

Step 2. Connect the file with the set of forbidden signatures (D^B).

Step 3. Connect a file with a set of undefined signatures (D^U).

Step 4. Upload a file with a list of rules.

Step 5. Forming a packet signature d_j from the original data stream (D), that you want to check.

Step 6. Performing the traffic classification method by features. If the signature is classified, you go to step 7. Otherwise, you go to step 9.

Step 7. If you need to continue checking the signatures, you go to step 5. Otherwise, you go to step 8.

Step 8. Finish the work.

Step 9. Perform the traffic classification method based on self-similarity. If the signature is classified, you go to step 7. Otherwise, you go to step 10.

Step 10. Performing a fuzzy method for detecting abnormal traffic. Go to step 7. The result of the work is an unambiguous classification of signatures as allowed or abnormal. Fig. 3 shows a diagram of data interaction in the hybrid method for detecting abnormal traffic. The set of initial data (D) is the source of input information for the traffic classification block by features (1). Also, the input data for this method is the set of forbidden signatures D^B(2) and the set of allowed signatures D^G(3). The result of the method is a set of undefined signatures D^U(4). The input to the traffic classification block based on self-similarity is a set of allowed signatures D^G(8) and a set of undefined signatures D^U(9). As a result of the block operation, the signatures are written to the set of allowed signatures D^G(7) and are removed from the set of undefined signatures D^U(10) or the sets remain unchanged. The input data for the fuzzy method of detecting abnormal traffic is a set of undefined signatures D^U(11) and a set of rules (12). As a result, the signature method from the set of undefined signatures D^U classifies and overwrites to the set of forbidden signatures D^B(5) or a set of allowed signatures D^G(6), by removing them from the set of undefined signatures D^U(13).

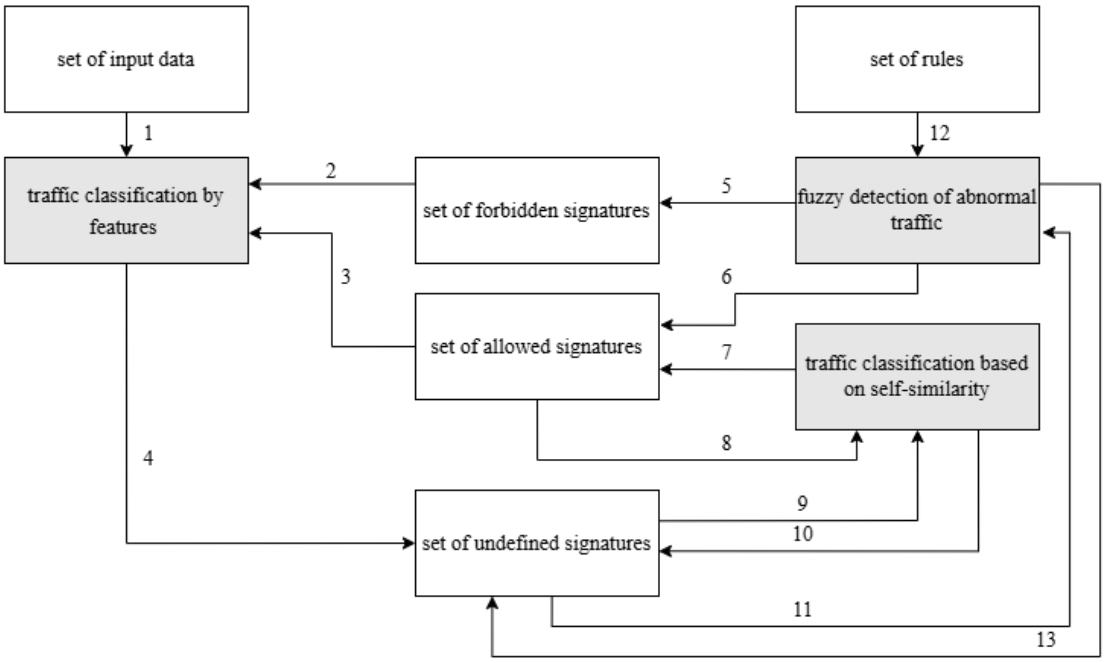

Figure 3: Scheme of data interaction in a hybrid method for detecting abnormal traffic in ICS

Based on the proposed hybrid method of detecting abnormal traffic, a system model was formed that covers the full cycle of network flow processing - from capturing raw traffic to making a decision on the security of the connection. At the initial stage, the specialized module integrates with the network environment, where it performs preliminary data processing: filtering by

features, normalizing formats, and aggregating packets into sessions. This preparation provides a high-quality basis for further traffic analysis and accurate detection of potential threats.

The received data is transferred to the traffic classification module, which identifies signatures according to the sets of allowed and forbidden connections. The signature database is dynamically updated in real time, which allows detecting both known threats and modified or new attack variants. At the same time, a traffic classification subsystem based on self-similarity compares current characteristics with models of typical user and attacker behavior, detecting abnormal deviations, including zero-day attacks.

The main element of the model is a fuzzy abnormal traffic detection unit that uses linguistic variables and membership functions to generate an integrated risk score. The system operates on a fuzzy logic rule base that allows for a generalized threat level assessment based on parameters such as transmission speed, number of connection attempts, signature matching, etc.

The logical inference subsystem combines the results of the classification, self-similarity, and fuzzy logic modules to provide a comprehensive decision-making mechanism. If undefined or contradictory situations are detected, the system can direct the flow for additional verification, and if the threat level is high, initiate interaction with other cybersecurity components, including firewalls and intrusion prevention systems.

All events recorded during traffic processing are recorded in a centralized logging system, including IP addresses, ports, timestamps, and protocol types. This data is used not only for auditing and incident analysis, but also for feedback, which allows us to refine classification rules and update self-similarity models.

5. Experimental results and discussion

General requirements for the efficiency of solving the problem solved by an anomaly detection system can be expressed through the following well-known and frequently used quality metrics: TP (True Positive) - the number of events that are correctly classified as abnormal and are in fact so; FP (False Positive) - the number of events that are mistakenly identified as abnormal, but are not; TN (True Negative) - the number of events that are not classified as abnormal and are not in fact so (i.e., are normal); FN (False Negative) - the number of events that are not identified as abnormal, but are in fact abnormal. FPs are often referred to as first-order errors and FNs as second-order errors.

Table 1 illustrates the results of testing the developed method using two different datasets - WSN-DS and CICIoT2023. For each dataset, the number of abnormal and allowed records was analyzed, and the rates of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications were calculated. In particular, 149,865 records were processed during the WSN-DS testing, of which 136,027 were abnormal and 13,838 were allowed. The method provided 13,187 correctly detected anomalies (TP) and 135,701 correctly identified valid records (TN), with 326 cases of false positive (FP) and 651 cases of false negative (FN) classification. During the CICIoT2023 testing, a much larger amount of data was processed - 9,337,316 records, where 9,117,677 were abnormal and 219,639 were allowed. According to the test results, 218,904 true positive classifications (TP) and 9,112,357 true negative classifications (TN) were obtained, while 5,320 false positive (FP) and 735 false negative (FN) records were detected. The results obtained indicate the high efficiency of the proposed method in processing both medium and large volumes of network traffic typical of Internet of Things (IoT) systems.

Table 1
Testing with a dataset

Dataset	Number of abnormal records	Number of allowed records	Total records	TP	TN	FP	FN
WSN-DS	136027	13838	149865	13187	135701	326	651
CICIoT2023	9117677	219639	9337316	218904	9112357	5320	735

Based on these metrics, the following characteristics can be calculated:
Recall - the ratio of correctly classified positive samples to the total number of positive samples:

$$Recall = \frac{TP}{TP + FN} \quad (24)$$

Precision - the proportion of correctly identified abnormal events among all events that the system has identified as abnormal:

Accuracy is the share of correctly detected and incorrectly not detected events among all events:

$$Precision = \frac{TP}{TP + FP} \quad (25)$$

Accuracy is the share of correctly detected and incorrectly not detected events among all events:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (26)$$

F1-score is the average value between Recall and Precision:

$$F1\,score = \frac{Recall + Precision}{2} \quad (27)$$

These metrics allow us to assess the reliability of the detection system and analyze its performance in different conditions. The reliability of the datasets is presented in Table 2.

Table 2
Characteristics of reliability assessment when using a dataset

Dataset	Acccuracy	Precision	recall	F1-score
WSN-DS	99,35	97,59	95,30	96,43
CICIoT2023	99,94	97,63	99,67	98,64

Table 2 provides a comparative characterization of the performance of the hybrid method for detecting abnormal traffic based on the WSN-DS and CICIoT2023 datasets, evaluating it by key metrics: accuracy, positive prediction accuracy, recall, and F1-measure. In the case of the WSN-DS set, the method showed an overall accuracy of 99.35%, which indicates a high quality of classification. The precision value reached 97.59%, indicating a small number of false positives. The completeness of the classification was 95.30%, meaning that most anomalies were detected

correctly. The F1-measure, which reflects the balance between accuracy and completeness, was 96.43%. In contrast, testing on CICIoT2023 demonstrated an even higher overall accuracy of 99.94%, with precision at 97.63% and extremely high completeness at 99.67%, meaning almost complete coverage of the detected abnormal records. The F1-measure in this case was 98.64%, confirming the balance of the results. The results are graphically presented in Fig. 4.

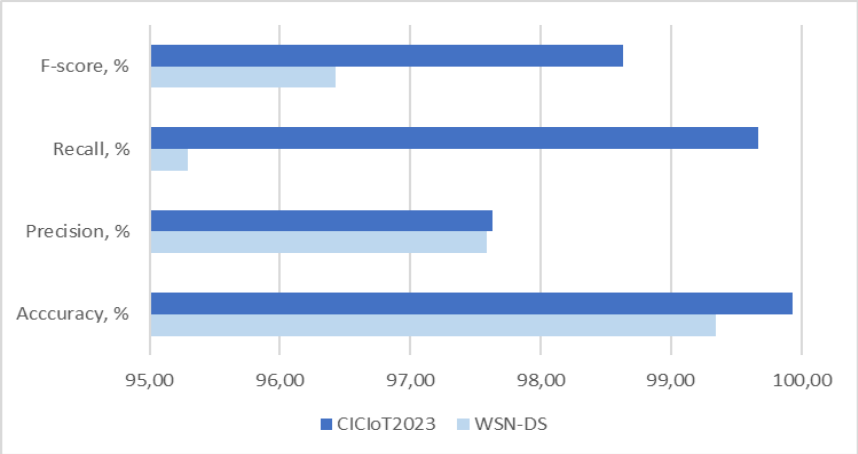

Figure 4: Diagram of the characteristics of reliability assessment when using a dataset

To evaluate the impact of the proposed method, we tested the systems under a load of normal traffic (~200 Mbps) with sequential experiments.

Formula (28) allows us to calculate the average CPU utilization during the operation of the traffic analysis system as part of the experiment. It takes into account the variable load during different time intervals:

$$\overline{CPU} = \frac{\sum_{i=1}^{n-1} CPU_i * (t_{i+1} - t_i)}{\sum_{i=1}^{n-1} (t_{i+1} - t_i)} \tag{28}$$

where CPU_i - processor load on a time interval; $\overline{CPU}$ - the average value of the CPU load for the selected time period. It is not a simple arithmetic average, but a weighted average that takes into account the duration of each interval; t_i and t_{i+1} - key points in time at which measurements were made; $t_{i+1} - t_i$ - is the duration of the interval during which a certain load was recorded.

Figures 5 and 6 illustrate the comparison of channel utilization and central processing unit (CPU) load with and without the proposed method on the WSN-DS and CICIoT2023 datasets.

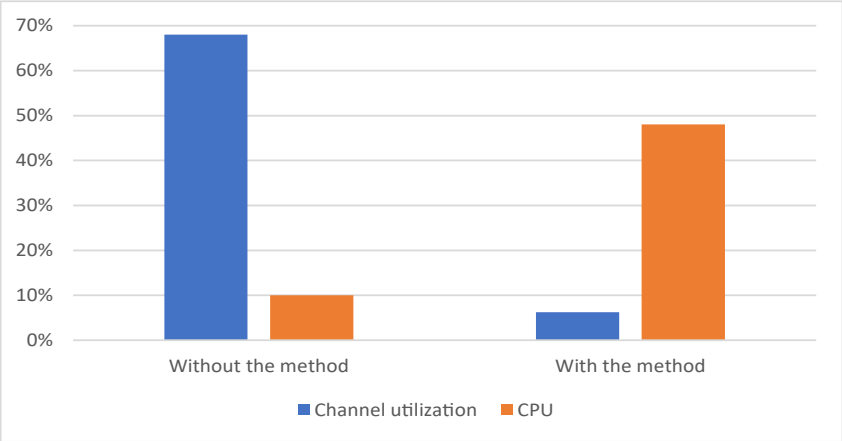

Figure 5: Comparison of channel utilization and CPU load on the WSN-DS dataset

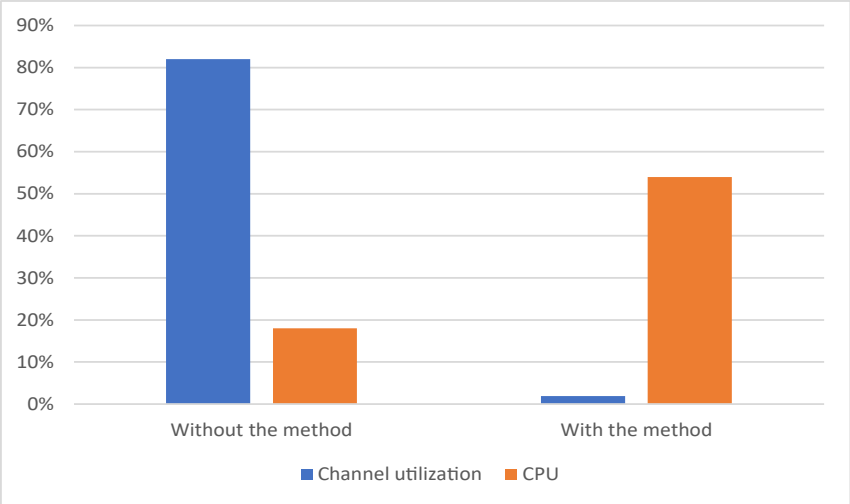

Figure 6**:** Comparison of channel utilization and CPU load on the CICIoT2023 dataset

For the WSN-DS dataset, the use of the proposed approach reduces the channel utilization from 68.00 % to 6.28 %, which is accompanied by an increase in the CPU load from 10.00 % to 48.00 %. A similar trend is observed for the CICIoT2023 dataset: the channel utilization decreases from 82.00 % to 1.93 %, while the CPU load increases from 18.00 % to 54.00 %. The results obtained indicate the effectiveness of the proposed method in reducing the network load, accompanied by a moderate increase in computational costs, which is acceptable for use in IoT networks.

6. Conclusions

This paper proposes a hybrid method for detecting abnormal traffic in IoT networks based on a step-by-step combination of signature classification, traffic self-similarity analysis, and fuzzy inference. The proposed model allows to effectively formalize network traffic in the form of signatures, providing high accuracy of packet classification in real time. The use of the Hurst coefficient to assess the self-similarity of traffic and fuzzy decision-making systems for processing uncertain data has significantly increased the adaptability and reliability of threat detection. Experimental studies on the WSN-DS and CICIoT2023 datasets have demonstrated the high efficiency of the proposed approach. An overall classification accuracy of more than 99% was achieved, as well as high Precision, Recall, and F1 measures, which indicates a balance between correct anomaly detection and minimization of false positives. An additional advantage of the method is a significant reduction in network load with an acceptable increase in the load on computing resources.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to grammar and spell check, and improve the text readability. After using the tool, the authors reviewed and edited the content as needed to take full responsibility for the publication's content.

References

[1] E. I. Elsedimy, S. M. M. AboHashish, An intelligent hybrid approach combining fuzzy C-means and the sperm whale algorithm for cyber attack detection in IoT networks, Sci. Rep. 15.1 (2025). doi:10.1038/s41598-024-79230-4.

[2] Koroliuk, R., Nykytyuk, V., Tymoshchuk, V., Soyka, V., Tymoshchuk, D. Automated monitoring of bee colony movement in the hive during winter season. CEUR Workshop Proceedings, (2024), 3842, pp. 147-156

[3] I. Mutambik, Enhancing iot security using GA-HDLAD: A hybrid deep learning approach for anomaly detection, Appl. Sci. 14.21 (2024) 9848. doi:10.3390/app14219848.

[4] A. K. Al Hwaitat, H. N. Fakhouri, Adaptive cybersecurity neural networks: an evolutionary approach for enhanced attack detection and classification, Appl. Sci. 14.19 (2024) 9142. doi:10.3390/app14199142.

[5] V. Martsenyuk, A. Klos-Witkowska, A. Sverstiuk, Stability investigation of biosensor model based on finite lattice difference equations, in: Difference equations and discrete dynamical systems with applications, Springer International Publishing, Cham, 2020, pp. 297–321. doi:10.1007/978-3-030-35502-9_13.

[6] V. Martsenyuk, A. Sverstiuk, I. S. Gvozdetska, Using differential equations with time delay on a hexagonal lattice for modeling immunosensors, Cybern. Syst. Anal. 55.4 (2019) 625–637. doi:10.1007/s10559-019-00171-2.

[7] V. P. Martsenyuk, I. Y. Andrushchak, P. N. Zinko, A. S. Sverstiuk, On application of latticed differential equations with a delay for immunosensor modeling, J. Autom. Inf. Sci. 50.6 (2018) 55–65. doi:10.1615/jautomatinfscien.v50.i6.50.

[8] Stanko, A., Wieczorek, W., Mykytyshyn, A., Holotenko, O., Lechachenko, T. Real-time air quality management: Integrating IoT and Fog computing for effective urban monitoring. CEUR Workshop Proceedings, (2024), vol. 3742, pp. 337–357

[9] A. G. Ayad, N. A. Sakr, N. A. Hikal, A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks, J. Supercomput. (2024). doi:10.1007/s11227-024-06409-x.

[10] Stanko, A., Duda, O., Mykytyshyn, A., Totosko, O., Koroliuk, R. Artificial Intelligence of Things (AIoT): Integration Challenges, and Security Issues. CEUR Workshop Proceedings, (2024), 3842, pp. 92–105

[11] Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V. Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, (2024), 3842, pp. 184 – 195

[12] N. Zagorodna, M. Stadnyk, B. Lypa, M. Gavrylov, R. Kozak, Network attack detection using machine learning methods, Chall. Natl. Def. Contemp. Geopolit. Situat. 2022.1 (2022) 55–61. doi:10.47459/cndcgs.2022.7.

[13] Stetsyuk, M., Cheshun, V., Stetsyuk, Y., Kozelskiy, O., Salem, A.-B.M. A model of a DDoS attack scenario on elements of specialized information technology and methods of combating cybercriminals. CEUR Workshop Proc., (2024), 3675, 260–269.

[14] J. I. Iturbe-Araya, H. Rifà-Pous, Enhancing unsupervised anomaly-based cyberattacks detection in smart homes through hyperparameter optimization, Int. J. Inf. Secur. 24.1 (2024). doi:10.1007/s10207-024-00961-6.

[15] M. Almasre, A. Subahi, Create a realistic iot dataset using conditional generative adversarial network, J. Sens. Actuator Netw. 13.5 (2024) 62. doi:10.3390/jsan13050062.
[16] H. Kamal, M. Mashaly, Enhanced hybrid deep learning models-based anomaly detection method for two-stage binary and multi-class classification of attacks in intrusion detection systems, Algorithms 18.2 (2025) 69. doi:10.3390/a18020069.
[17] H. Kamal, M. Mashaly, Robust intrusion detection system using an improved hybrid deep learning model for binary and multi-class classification in iot networks, Technologies 13.3 (2025) 102. doi:10.3390/technologies13030102.
[18] Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T. Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, (2024), 3896, pp. 1-11.
[19] Klots Y., Petliak N., Martsenko S., Tymoshchuk V., Bondarenko I. Machine Learning system for detecting malicious traffic generated by IoT devices. CEUR Workshop Proceedings, (2024), 3742, pp. 97 - 110
[20] F. Safarov, M. Basak, R. Nasimov, A. Abdusalomov, Y. I. Cho, Explainable lightweight block attention module framework for network-based iot attack detection, Future 15.9 (2023) 297. doi:10.3390/fi15090297.
[21] Titova, V., Klots, Y., Cheshun, V., Petliak, N., Salem, A.-B.M.. Detection of network attacks in cyber-physical systems using a rule-based logical neural network. 1st International Workshop on Intelligent and CyberPhysical Systems, ICyberPhyS 2024, Volume 3736, 2024, Pages 255-268
[22] Y. Klots, N. Petliak and V. Titova, Evaluation of the efficiency of the system for detecting malicious outgoing traffic in public networks," 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece 2023, pp. 1–5, doi: 10.1109/DESSERT61349.2023.10416502.
[23] R. J. Alzahrani, A. Alzahrani, A novel multi algorithm approach to identify network anomalies in the iot using fog computing and a model to distinguish between iot and non-iot devices, J. Sens. Actuator Netw. 12.2 (2023) 19. doi:10.3390/jsan12020019.
[24] B. A. Alabsi, M. Anbar, S. D. A. Rihan, Conditional tabular generative adversarial based intrusion detection system for detecting ddos and dos attacks on the internet of things networks, Sensors 23.12 (2023) 5644. doi:10.3390/s23125644.
[25] F. Safarov, M. Basak, R. Nasimov, A. Abdusalomov, Y. I. Cho, Explainable lightweight block attention module framework for network-based iot attack detection, 15.9 (2023) 297. doi:10.3390/fi15090297.
[26] B. A. Narayanavadivoo Gopinathan, V. Sarveshwaran, V. Ravi, R. Chaganti, LPCOCN: A layered paddy crop optimization-based capsule network approach for anomaly detection at iot edge, Information 13.12 (2022) 587. doi:10.3390/info13120587.
[27] R. A. Manzano Sanchez, M. Zaman, N. Goel, K. Naik, R. Joshi, Towards developing a robust intrusion detection model using hadoop–spark and data augmentation for iot networks, Sensors 22.20 (2022) 7726. doi:10.3390/s22207726.
[28] I. Syamsuddin, O. M. Barukab, SUKRY: suricata IDS with enhanced knn algorithm on raspberry pi for classifying iot botnet attacks, Electronics 11.5 (2022) 737. doi:10.3390/electronics11050737.
[29] N. G. Anoh, T. Kone, J. C. Adepo, J. F. M'Moh, M. Babri, IoT intrusion detection system based on machine learning algorithms usingthe UNSW-NB15 dataset, Int. J. Adv. Sci. Res. Eng. 10.01 (2024) 16–28. doi:10.31695/ijasre.2024.1.3.
[30] B. A. Alabsi, M. Anbar, S. D. A. Rihan, CNN-CNN: dual convolutional neural network approach for feature selection and attack detection on internet of things networks, Sensors 23.14 (2023) 6507. doi:10.3390/s23146507.
[31] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, R. Braun, GNN-Based network traffic analysis for the detection of sequential attacks in iot, Electronics 13.12 (2024) 2274. doi:10.3390/electronics13122274.

[32] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, A. A. Atayero, SMOTE-DRNN: A deep learning algorithm for botnet detection in the internet-of-things networks, Sensors 21.9 (2021) 2985. doi:10.3390/s21092985.
[33] Petliak N., Klots Y., Titova V., Salem A.-B.M. Attack detection system based on network traffic analysis by means of fuzzy inference. CEUR Workshop Proceedings, (2024), 3899, pp. 201 – 213