

# Attack detection system based on network traffic analysis by means of fuzzy inference\*

Nataliia Petliak<sup>1,†</sup>, Yurii Klots<sup>1,\*,†</sup>, Vira Titova<sup>1,†</sup> and Abdel-Badeeh M. Salem<sup>2,†</sup>

<sup>1</sup> Khmelnytskyi National University, Institutska str.,11, Khmelnytskyi, 29000, Ukraine

<sup>2</sup> Ain Shams University, El-Khalyfa El-Mamoun Street Abbasya, Cairo, Egypt

## Abstract

This article presents an approach to analyzing network traffic using packet headers that provide information about the connection between network nodes. The bulk of the traffic is data, so the analysis is focused on headers that occupy a small part of the packet but contain important information about the connection structure. A method for selecting the most informative parameters is proposed, which allows an increase in the efficiency of the analysis and ensures the stable operation of the network. To implement the method, fuzzy inference tools are used, which allow uncertainty and blurred boundaries to be taken into account when classifying traffic. Based on expert opinions, trapezoidal membership functions were formed for each of the parameters, which allows the phasing of the input data and determining the degree of their belonging to specific terms. The proposed system is implemented on the basis of a hardware and software complex.

## Keywords

network traffic, fuzzy logic, signature analysis, traffic classification, network anomalies

## 1. Introduction

With the development of digital technologies and the growth of the global network infrastructure, cybersecurity threats have become one of the most serious problems in modern society. Every day, Internet users face numerous attacks on information systems that can lead to significant financial losses, data loss, and privacy violations. According to statistics, losses from cyber threats are growing every year, reaching billions of dollars [1-2]. Among the most common threats are network attacks such as DDoS, phishing attacks, malware, and brute force attacks [3-4]. In this regard, there is a need to develop new methods and technologies to detect and prevent network attacks. Traditional intrusion detection systems have a number of disadvantages, including limited ability to work under uncertainty and difficulty adapting to new types of threats. Modern approaches based on machine learning, fuzzy logic, and deep traffic analysis are becoming increasingly popular due to their ability to classify network traffic more efficiently and accurately.

The rapid development of network technologies contributes to the emergence of new types of attacks on computer networks. A variety of intrusion methods and their use in attacks threaten the effectiveness of existing security technologies in protecting data in corporate networks. This creates a constant need to improve technologies and tools to ensure reliable protection. The use of advanced information technologies is key to the effective management of various systems, and corporate computer networks remain indispensable tools for their successful operation. However, as networks


---

*AdvAIT-2024: 1st International Workshop on Advanced Applied Information Technologies, December 5, 2024, Khmelnytskyi, Ukraine - Zilina, Slovakia*

\* Corresponding author.

† These authors contributed equally.

✉ npetlyak@khnmu.edu.ua (N. Petliak); klots@khnmu.edu.ua (Y. Klots); titovav@khnmu.edu.ua (V. Titova); abmsalem@yahoo.com (Abdel-Badeeh M. Salem)

 0000-0001-5971-4428 (N. Petliak); 0000-0002-3914-0989 (Y. Klots); 0000-0001-8668-4834 (V. Titova); 0000-0003-0268-6539 (Abdel-Badeeh M. Salem);



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

grow, the number of users and the amount of information transmitted increases, which can lead to a deterioration in the quality of network services. This underscores the importance of improving network traffic monitoring and analysis tools to ensure the stability and quality of service.

## 2. Related works

Paper [5] describes a NIDS based on multidomain machine learning that analyzes the characteristics of traffic flows and payload data using two ML classifiers. One of them works with traffic flows, and the other with payload data. Both classifiers are based on a random forest algorithm, and their results are combined using a voting scheme. However, the described system works to detect attacks on network components.

Paper [6] presents a network intrusion detection system called SPAFIS, which uses fuzzy IF-THEN rules and software prototypes to analyze network traffic in real-time. The system is able to adapt to new data through self-development of its structure and parameters.

The authors of [7] propose a real-time network intrusion detection system. The main goal of the work is to test the effectiveness of the proposed model, which analyzes real packets with different types of attacks and safe traffic. An important part of the study is to improve signature generation methods for better detection of anomalies and malware.

In [8], the authors work on the development of an online system for detecting distributed denial of service (DDoS) attacks in a client-server environment. Their system consists of five modules that provide effective detection and response to suspicious activity using a dynamic thresholding algorithm based on Shannon's entropy and Chebyshev's theorem. The system is adaptive to changes in legitimate traffic.

The article [9] explores methods of detecting DDoS attacks in cloud computing. The authors proposed a combination of Mutual Information and Random Forest Feature Importance to select relevant features, which improves the accuracy of machine learning models, including Random Forest and Gradient Boosting. Using this approach on CICIDS 2017 and CICDDoS 2019 datasets showed a high level of accuracy in DDoS attack detection.

The paper [10] focuses on the development of an online system for detecting DDoS attacks using the entropy method with a dynamic threshold. Using network traffic analytics, the proposed approach allows you to automatically adapt to changes in the intensity of attacks, which increases accuracy and reduces the number of false positives.

The article [11] introduces a approach to defending against DDoS attacks. The RAD mechanism uses behavioral analysis and statistical modeling to detect anomalies and mitigate these attacks more effectively. The novelty of this research lies in its focus on leveraging dynamic traffic patterns and user behavior to distinguish between legitimate and malicious traffic, a significant improvement over traditional signature-based methods. This method offers a more adaptive and robust solution for real-time DDoS mitigation in cloud environments and large-scale networks.

The author of [12] describes the creation of a system based on fuzzy logic for classifying network traffic as malicious or harmless using weighting factors. The author explores the use of fuzzy systems to categorize network data into "good" and "bad" content. The system automates the process of analyzing and classifying traffic using a set of rules and can be integrated with other systems to improve the effectiveness of protection against cyber threats. However, the presented development offers a fuzzy inference system for classifying network packet types and detecting only TCP-SYN attacks.

S. R. Zahra and others [13] propose an intelligent system based on fuzzy logic and data mining, which is based on three layers, six segments, and 30 components that work synchronously with each other. This system is able to identify only phishing/malicious URL attacks with high accuracy.

In [14], a machine learning approach is proposed to detect DDoS attacks in software-defined networks. When detecting attacks, some flow characteristics are used to determine normal network traffic. The proposed approach is tested in four different machine learning algorithms.

The authors of [15] propose a model for detecting SSH-Brute Force attacks based on deep learning. The study showed that the CNN model outperforms traditional machine learning algorithms, such as naive Bayes, logistic regression, and others, in the ability to detect Brute Force attacks.

The authors of the article [16] propose the use of deep neural networks to detect network anomalies in IT infrastructures of the oil and gas industry. The proposed approach automates the process of feature selection from raw traffic, which improves the accuracy of models and reduces time spent on manual processing. This provides a cyber security solution in a specific industrial environment.

Article [17] is devoted to methods of detecting network attacks on cyber-physical systems (CPS) using neural networks based on logical rules. The main focus is on analyzing different approaches to data representation in CPS and evaluating their advantages and disadvantages. The authors propose a method for detecting attacks based on multivariate time series analyzed with the help of a logical neural network. This method allows for the prediction of the state of the system and the comparison of predicted and actual values to detect anomalies. An important step is the segmentation of the network to protect its various parts, which allows for more efficient detection of potential threats. Among the advantages of the proposed approach are high accuracy and the ability to detect short-term anomalies in the operation of the CPS. The authors point out a disadvantage - poor efficiency in detecting long-term anomalies that develop slowly. It should be noted that the method requires powerful computing resources, and the model needs to be re-trained when the system topology changes.

Paper [18] investigates machine learning methods for detecting network intrusions based on traffic flows. The focus is on methods based on decision trees, such as PART, J48, and random forest. Processing time is also taken into account. This is important for real-time, as fast processing of traffic flows allows for real-time detection and response to threats.

The article [19] examines multi-criteria methods of assessing the correctness of decision-making in the field of cyber security and information security. The authors pay attention to the analysis of the correctness of the decisions made in the context of the protection of specific objects from information weapons and vulnerabilities of computer technologies, which is an important factor in the development of new methods or the implementation of systems for detecting unauthorized actions or attacks.

The reviewed works explore different approaches to detecting network intrusions using machine learning, fuzzy logic, and deep learning methods. The main tools are the analysis of traffic flows, payloads, signatures, and attributes of network traffic using various classifiers. Attention is also focused on the importance of processing time to ensure effective real-time threat detection. However, the proposed solutions analyze incoming network traffic and are focused on certain types of attacks, such as phishing or DDoS attacks.

### **3. The system of fuzzy logical inference**

From the analysis, we can conclude that network traffic analysis allows us to detect attacks on the network with high reliability. For such analysis, machine learning and fuzzy logic methods are usually used. However, significant volumes of traffic to be analyzed require a significant increase in the capacity of the traffic analysis system or analysis of only a part of the parameters characterizing the packets. The use of informative traffic characteristics can significantly reduce the load on the analysis system and perform it in real time without significantly reducing network performance.

The bulk of the packet when two nodes communicate is data, so only headers are used to analyze traffic, which occupies a small part of the packet size and contains information about the connection. However, some headers are not used for traffic analysis. This is because these elements contain only service information, and their analysis can create an additional load on the system without providing important data about possible attacks. Using all the parameters would lead to an increased load on the network equipment, which, in turn, would reduce the efficiency of network traffic analysis and

slow down the data transfer rate. Therefore, the most informative parameters were selected to ensure the stable operation of the computer network during data exchange. Therefore, the packet signature is presented as follows [20]:

$$s = \{IPs, IPd, Ps, Pd, Pr, ITr, T, MAC, S\}, \quad (1)$$

where  $IPs$  is the source IP address that sends a request for connection and information exchange;  $IPd$  is the destination IP address, i.e., to which IP address requests are sent;  $Ps$  is the source port used to establish the connection;  $Pd$  is the destination port;  $Pr$  is the protocol used for data transmission;  $ITr$  is the traffic intensity determined by bits/s;  $T$  is the time the packet arrives for verification in the 24-hour format hh-mm-ss;  $MAC$  is the MAC address of the device that sends data from the network;  $S$  is the packet size.

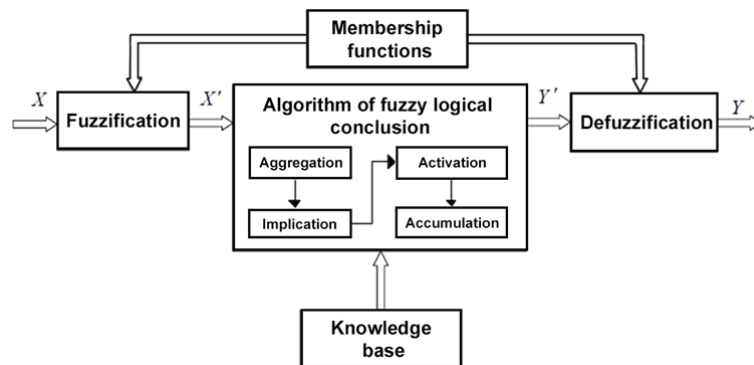
The  $IPs, Ps, MAC$  parameters are used to block the device that generates malicious traffic. The  $T$  parameter is used to store signatures in the dictionaries for a certain period of time after malicious data is received from the device, after the specified time period expires, the device data will be removed from the database of prohibited device signatures, thus the time to check prohibited connections will not increase over time. Since a linear search is used to check an item with all the items in the database, the time changes in proportion to the number of items in the database:

$$t = k * n, \quad (2)$$

where  $t$  is the total check time;  $k$  is a constant that depends on the speed of one check operation;  $n$  is the number of items in the database.

The proposed solution was implemented through a hardware and software complex, where a program code was created using fuzzy logic methods to check the signature for compliance with the type of network traffic. This approach allows for the taking into account of fuzzy criteria and blurred boundaries when classifying traffic.

The structure of the system for implementing fuzzy inference for classifying the features of signature elements using Matlab is shown in Fig. 1. It consists of the following blocks: phasing, membership functions, fuzzy inference algorithm, and defuzzification. In general, the sequence of system steps (Fig. 1) is as follows: input data  $X$  is fed to the fuzzification unit; the fuzzification stage converts the input data into fuzzy values  $X'$  using membership functions; the fuzzy inference algorithm performs implication, aggregation, activation and accumulation based on the fuzzy knowledge base to obtain fuzzy output values  $Y'$ ; defuzzification converts the fuzzy output values  $Y'$  into a clear output value  $Y$ ; the output value  $Y$  is the result of the system, which is used for decision-making.



**Figure 1:** Structure of the system for realizing fuzzy inference.

During fuzzification, the crisp input data  $X$  is transformed into fuzzy values  $X'$  using membership functions. This allows us to take into account the uncertainty and fuzziness in the input data. For example,  $ITr$  is equal to 10,000 bits per second, and the membership function converts this intensity to the fuzzy value “above average” at the phasing stage.

Membership functions determine the degree to which each input value belongs to certain terms. The linguistic variables required for network traffic analysis are  $IPd, Pd, Pr, ITr, S$ . Each input linguistic variable is defined by a term set of a certain number of values.

$IPd$  is represented by the following set:

$$IPd = \{IPd_{permit}, IPd_{unknown}, IPd_{prohibited}\}, \quad (3)$$

where  $IPd_{permit}$  is the set of allowed IP addresses, which includes IP addresses of social networks, search engines and other frequently used sites when connecting to public networks;  $IPd_{unknown}$  – a set of IP addresses whose affiliation is not defined as allowed or prohibited;  $IPd_{prohibited}$  – a set of prohibited IP addresses, which includes IP addresses from the blacklist of blocked addresses [21-22].

$Pd$  is defined by the following set:

$$Pd = \{Pd_{permit}, Pd_{unknown}, Pd_{prohibited}\}, \quad (4)$$

where  $Pd_{permit}$  – is the set of allowed ports; this set includes the frequently used 80 and 443 ports, ports used for popular social networks when using a browser or applications;  $Pd_{unknown}$  – the set of ports whose affiliation is not defined as allowed or prohibited;  $Pd_{prohibited}$  – the set of prohibited ports, which includes ports intended for remote access such as 21, 22 and 3389.

$Pr$  is defined by the following set:

$$Pr = \{Pr_{permit}, Pr_{unknown}, Pr_{prohibited}\}, \quad (5)$$

where  $Pr_{permit}$  – is the set of initially allowed protocols, including TCP and UDP;  $Pr_{unknown}$  – is the set of protocols whose affiliation is not initially determined as allowed or prohibited;  $Pr_{prohibited}$  – is the set of prohibited protocols, including RDP.

$ITr$  is defined by the following set:

$$ITr = \{ITr_{low}, ITr_{below\ average}, ITr_{average}, ITr_{above\ average}, ITr_{high}\}, \quad (6)$$

where  $ITr_{low}$  – is a set of traffic intensity values defined as low intensity;  $ITr_{below\ average}$  – is a set of traffic intensity values defined as below average;  $ITr_{average}$  – a set of traffic intensity values that are defined as an average intensity value;  $ITr_{above\ average}$  – a set of traffic intensity values that are defined as above average;  $ITr_{high}$  – a set of traffic intensity values that are defined as high intensity.

$S$  is defined by the following set:

$$S = \{S_{low}, S_{below\ average}, S_{average}, S_{above\ average}, S_{high}\}, \quad (7)$$

where  $S_{low}$  – is the set of data packet size values that are defined as small;  $S_{below\ average}$  – is the set of data packet size values that are defined as below average;  $S_{average}$  – is the set of data packet size values that are defined as average packet size;  $S_{above\ average}$  – is the set of data packet size values that are defined as above average;  $S_{high}$  is the set of data packet size values that are defined as large data packet size.

The membership function was chosen as a trapezoidal function because it can be used to model different states of network traffic, which allows for more detailed analysis and classification of traffic behavior, which is important for detecting anomalies and ensuring network security. The trapezoidal membership function is used to describe the membership of linguistic variables to certain terms. It has the shape of a trapezoid and is defined by four parameters:  $a, b, c$  and  $d$ , where  $a$  and  $d$  are the lower bases of the trapezoid,  $b$  and  $c$  are the upper bases. Formally, the trapezoidal membership function  $\mu(x)$  is defined as [23-25]:

$$\mu(x) = \begin{cases} 0, & \text{if } x \leq a \text{ or } x \geq d \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b \\ 1, & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{if } c \leq x \leq d \end{cases} \quad (8)$$

where  $a \leq b \leq c \leq d$ ;  $x$  is an input value compared with the defined limits for each term set. For each value of  $x$ , the trapezoidal membership function calculates the degree of membership of this value in a particular fuzzy set. This degree of membership ranges from 0 to 1, where 0 means no membership and 1 means full membership. The membership function calculates how much the value  $x$  belongs to a particular term. This allows you to phase the value of  $x$ , that is, determine the degree of its membership in each fuzzy set.

The parameters for each set of values are set based on static data. The data for all terms of linguistic variables are presented in Table 1.

**Table 1**

Data of the terms of linguistic variables

Linguistic variables	IPd	permit	[0 0.01 0.04 0.15]
		unknown	[0.05 0.13 0.77 0.94]
		prohibited	[0.71 0.93 0.99 1]
	Pd	permit	[0 0.01 0.1 0.36]
		unknown	[0.09 0.24 0.78 0.89]
		prohibited	[0.66 0.93 0.98 1]
	Pr	permit	[0 0.02 0.15 0.37]
		unknown	[0.17 0.31 0.73 0.84]
		prohibited	[0.63 0.86 0.99 1]
	ITr	low	[0 0.01 0.1 0.24]
		below average	[0.12 0.17 0.36 0.42]
		average	[0.34 0.44 0.57 0.65]
		above average	[0.57 0.66 0.83 0.91]
		high	[0.8 0.91 0.99 1]
	S	low	[0 0.02 0.1 0.23]
below average		[0.1 0.21 0.31 0.4]	
average		[0.29 0.41 0.58 0.7]	
above average		[0.6 0.66 0.77 0.91]	
		high	[0.8 0.88 0.98 1]

Phased median estimation is an important tool for analyzing and processing data in situations with a high level of uncertainty, providing a more accurate estimate of the mean. Since network traffic analysis belongs to this category of tasks, the phased median estimate is calculated in Table 2.

**Table 2**

Formed linguistic variables

Linguistic variables	Set of possible values	Phased median estimate
IPd	IPd={p=allowed IP addresses, unk=unspecified, pr=banned IP addresses}	IPd={p=0.05, unk=0.473, pr=0.91}
Pd	Pd={p=allowed ports, unk=unspecified, pr=prohibited ports}	Pd={p=0.118, unk=0.5, pr=0.893}
Pr	Pr={p=allowed protocols, unk=unspecified, pr=prohibited protocols}	Pr={p=0.135, unk=0.513, pr=0.87}

ITr	ITr={l=low traffic intensity, ba=below average traffic intensity, a=average traffic intensity, aa=above average traffic intensity, h=high traffic intensity}	ITr={l=0.088, ba=0.268, a=0.5, aa=0.743, h=0.925}
S	S={l=low packet size, ba=below average packet size, a=average packet size, aa=above average packet size, h=high packet size}	S={l=0.088, ba=0.255, a=0.495, aa=0.735, h=0.915}

Here is an example of a trapezoidal membership function for five terms of a linguistic variable *ITr*.

The trapezoidal membership function for a term from the set  $ITr_{low}$  of the linguistic variable *ITr* will be as follows:

$$\mu_{low}(x) = \begin{cases} 0, & \text{if } x \leq 0 \text{ or } x \geq 0.24 \\ \frac{x}{0.01}, & \text{if } 0 \leq x \leq 0.01 \\ 1, & \text{if } 0.01 \leq x \leq 0.1 \\ \frac{0.24 - x}{0.14}, & \text{if } 0.1 \leq x \leq 0.24 \end{cases} \quad (9)$$

The trapezoidal membership function for a term from the set  $ITr_{below\ average}$  of the linguistic variable *ITr* will be as follows:

$$\mu_{below\ average}(x) = \begin{cases} 0, & \text{if } x \leq 0.12 \text{ or } x \geq 0.42 \\ \frac{x - 0.12}{0.05}, & \text{if } 0.12 \leq x \leq 0.17 \\ 1, & \text{if } 0.17 \leq x \leq 0.36 \\ \frac{0.42 - x}{0.06}, & \text{if } 0.36 \leq x \leq 0.42 \end{cases} \quad (10)$$

The trapezoidal membership function for a term from the set  $ITr_{average}$  of the linguistic variable *ITr* will be as follows:

$$\mu_{average}(x) = \begin{cases} 0, & \text{if } x \leq 0.34 \text{ or } x \geq 0.65 \\ \frac{x - 0.34}{0.1}, & \text{if } 0.34 \leq x \leq 0.44 \\ 1, & \text{if } 0.44 \leq x \leq 0.57 \\ \frac{0.65 - x}{0.08}, & \text{if } 0.57 \leq x \leq 0.65 \end{cases} \quad (11)$$

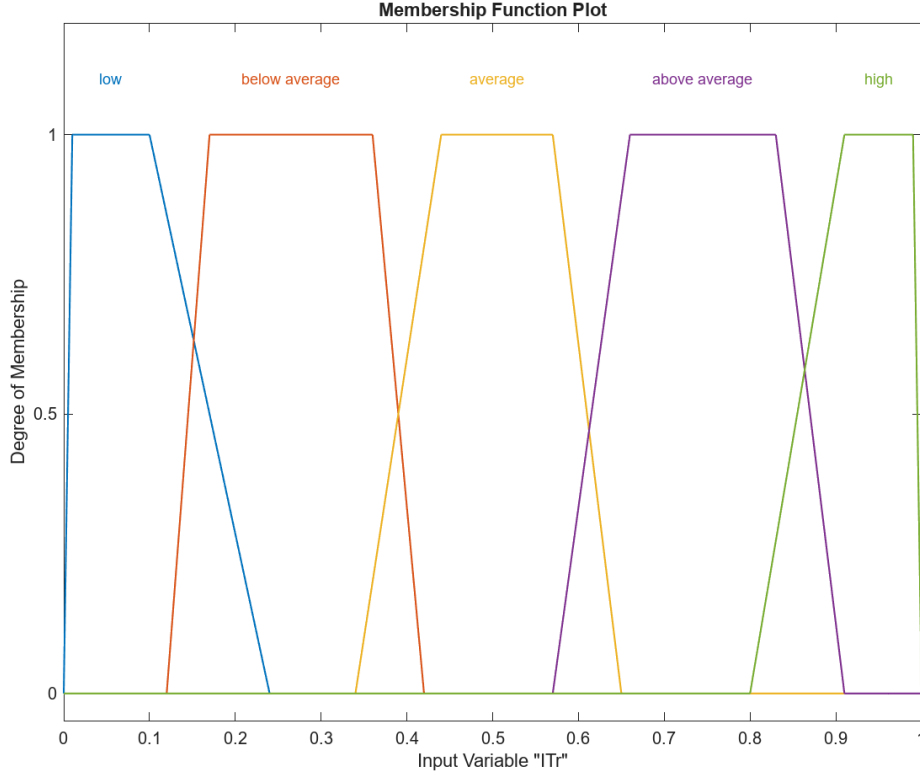
The trapezoidal membership function for a term from the set  $ITr_{above\ average}$  of the linguistic variable *ITr* will be as follows:

$$\mu_{above\ average}(x) = \begin{cases} 0, & \text{if } x \leq 0.57 \text{ or } x \geq 0.91 \\ \frac{x - 0.57}{0.09}, & \text{if } 0.57 \leq x \leq 0.66 \\ 1, & \text{if } 0.66 \leq x \leq 0.83 \\ \frac{0.91 - x}{0.08}, & \text{if } 0.83 \leq x \leq 0.91 \end{cases} \quad (12)$$

The trapezoidal membership function for a term from the set  $ITr_{high}$  of the linguistic variable *ITr* will be as follows:

$$\mu_{high}(x) = \begin{cases} 0, & \text{if } x \leq 0.8 \text{ or } x \geq 1 \\ \frac{x - 0.8}{0.11}, & \text{if } 0.8 \leq x \leq 0.91 \\ 1, & \text{if } 0.91 \leq x \leq 0.99 \\ \frac{1 - x}{0.01}, & \text{if } 0.99 \leq x \leq 1 \end{cases} \quad (13)$$

Graphically, these functions are shown in Fig. 3.



**Figure 3:** Membership functions for the linguistic variable ITr.

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

$$\mu_{M_{IPd}} \cap \mu_{M_{Pd}} \cap \mu_{M_{Pr}} \cap \mu_{M_{ITr}} \cap \mu_S = \min(\mu_{M_{IPd}}(x), \mu_{M_{Pd}}(x), \mu_{M_{Pr}}(x), \mu_{M_{ITr}}(x), \mu_S(x)), \quad (14)$$

where  $M_{IPd} = \{m_{IPd_i}\}_{i=0}^{4,3 \cdot 10^9}$  is the set of possible values of the destination IP address;  $M_{Pd} = \{m_{Pd_i}\}_{i=0}^{65535}$  is the set of possible values of the destination ports;  $M_{Pr} = \{m_{Pr_i}\}_{i=0}^{36}$  - the set of possible protocols;  $M_{ITr} = \{m_{ITr_i}\}_{i=0}^{10 \cdot 10^9}$  - the set of possible traffic intensity values;  $M_S = \{m_{S_i}\}_{i=0}^{65535}$  - the set of possible values of the 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".

The next step is to determine the set of rules for fuzzy classification. A total of 1125 rules were developed. Here are some examples.

If the destination IP address is prohibited, the connection will be blocked regardless of other parameters. This means that as soon as the system determines that the destination IP address is prohibited, no other parameter (protocol, traffic intensity, packet size) will be taken into account. A decision is automatically made based on the IP address being banned, and access to that address is blocked:

$$if (IPd \text{ is prohibited}) \Rightarrow (result \text{ is blocked}) \quad (15)$$



If the traffic volume is “high” and the packet size is “small”, the connection will be blocked. This is because high traffic intensity may indicate a potential threat (for example, a DDoS attack or excessive data flow), and a “small” packet size may indicate anomalous or malicious requests. In such cases, the system blocks the connection, regardless of protocol or IP address permissions:

$$\text{if } ((ITr \text{ is high}) \wedge (S \text{ is low})) \Rightarrow (\text{result is blocked}) \quad (16)$$

If the IPd allows the connection, the protocol being used is also allowed, and the traffic volume is average, the system assumes that it is a secure connection and allows it. In this case, there are no serious threats, and all parameters indicate a secure connection:

$$\text{if } ((IPd \text{ is permit}) \wedge (Pr \text{ is permit}) \wedge (ITr \text{ is average})) \Rightarrow (\text{result is allowed}) \quad (17)$$

If the protocol is allowed, the traffic volume is high, and the packet size is small, the connection will be blocked. Even if the protocol is allowed, a high traffic volume combined with a small packet size may indicate a potential threat or anomalous behavior. Therefore, the connection will be blocked to protect the system:

$$\text{if } ((Pr \text{ is permit}) \wedge (ITr \text{ is high}) \wedge (S \text{ is low})) \Rightarrow (\text{result is blocked}) \quad (18)$$

If the port and protocol are allowed and the traffic volume is below average, the connection is allowed. In this case, both the final decision and the protocol request are allowed, and the threat level (traffic intensity) is low. This means that there is no reason to block the connection, and it can be allowed:

$$\begin{aligned} \text{if } ((Pd \text{ is permit}) \wedge (Pr \text{ is permit}) \wedge (ITr \text{ is below average})) \\ \Rightarrow (\text{result is allowed}) \end{aligned} \quad (19)$$

If the destination IP address and port are not defined, the protocol is allowed, and the traffic intensity and packet size values are average, the connection will be allowed:

$$\begin{aligned} \text{if } (IPd \text{ is unknown}) \wedge (Pd \text{ is unknown}) \wedge (Pr \text{ is permit}) \wedge \\ \wedge (ITr \text{ is average}) \wedge (S \text{ is average}) \Rightarrow (\text{result is allowed}) \end{aligned} \quad (20)$$

If the IP address is allowed, the protocol is allowed, the traffic intensity is above average, and the packet size is small, then the connection will be allowed:

$$\begin{aligned} \text{if } (IPd \text{ is permit}) \wedge (Pr \text{ is permit}) \wedge (ITr \text{ is above average}) \wedge \\ \wedge (S \text{ is below average}) \Rightarrow (\text{result is allowed}) \end{aligned} \quad (21)$$

If the port is forbidden, the protocol is allowed, the traffic intensity is above average, and the packet size is small, then the connection will be blocked:

$$\begin{aligned} \text{if } (Pd \text{ is prohibited}) \wedge (Pr \text{ is permit}) \wedge (ITr \text{ is above average}) \wedge \\ \wedge (S \text{ is below average}) \Rightarrow (\text{result is blocked}) \end{aligned} \quad (22)$$

The fuzzy logic inference algorithm performs the process of obtaining a fuzzy logical inference based on fuzzy rules. It consists of four stages: implication, aggregation, activation and accumulation. Implication transforms input fuzzy values  $X'$  into output fuzzy sets. Aggregation combines the implication results from different rules for each output term. Activation applies the matching degree of each rule to the original fuzzy set using minimization methods to determine the degree of rule activation. Accumulation combines the original fuzzy sets into one fuzzy set  $Y'$  for each original term.

Defuzzification transforms the original fuzzy values of  $Y'$  into crisp original values of  $Y$ . This is done to obtain a specific decision based on the fuzzy inference. The result of the system's operation will be one result value, which takes one of two parameters: "allowed" [0, 0.5] or "forbidden" [0.5,1]. Therefore, if the signature received the result "allowed" upon completion of the system check, then data transfer is permitted, and the signature is recorded in the database of permitted connections. If the signature is defined as "forbidden", then the connection is blocked by the IP and MAC address of the sender, and the signature is added to the database of prohibited.

## 4. Testing

In this work, specialized data sets such as KDDCup99, NSL-KDD, UNSW-NB15, WSN-DS, and CICIoT2023 were used to test the system. They serve as a benchmark against which different models can be compared, allowing performance to be assessed based on real and simulated threat scenarios. These sets provide both legitimate and malicious traffic, which allows you to simulate real situations in networks and evaluate the accuracy, detection speed and resistance to false alarms of threat detection systems. With a variety of attack types and normal sessions, these datasets help adapt systems to today's threats and increase their reliability in a real network environment [26, 27].

The KDD99 dataset is one of the oldest and most well-known for intrusion detection analysis. It was created as part of the KDD Cup 1999 and includes data on various attacks such as DoS, R2L, and U2R.

The NSL-KDD dataset is an improved version of KDD'99 and was designed to eliminate problems such as excessive duplicate records. NSL-KDD contains data on various network intrusions and normal sessions, which can be used for training and testing attack detection systems. Its feature is a reduced volume of data without loss of quality, which facilitates processing and analysis. It covers attacks such as DoS, R2L, U2R and network scanning attempts.

The UNSW-NB15 dataset was collected in 2015 as part of a research project by the Australian Center for Cyber Security (ACCS). It combines normal traffic with today's sophisticated attacks, including DoS, intrusions, privilege abuse, backdoor attacks, and botnets. In total, the set contains 49 attributes for each session, which allows for detailed traffic analysis at various levels.

CICIDS-2017 is one of the most comprehensive datasets used for network intrusion detection analysis. It contains traffic collected under real-world conditions over five days and covers a wide range of attacks, including DoS, brute force and SQL injection.

The ISCXVPN2016 dataset was created to study traffic passing through VPNs and regular connections. It is useful for analyzing anomaly detection systems, as VPNs are often used to mask malicious activity. This set includes traffic from both legitimate activities (browsing web pages, videos) and potentially malicious activities performed over a VPN connection.

The SDN-Dataset focuses on software-defined networks, which are becoming increasingly popular due to their flexibility and centralized management. However, this structure also makes SDN vulnerable to specific attacks, such as traffic manipulation or attacks on the controller. The SDN-Dataset contains both normal and malicious traffic, which allows analyzing threats in such networks and developing methods for their protection.

A confusion matrix was used to assess the reliability of the developed system. A confusion matrix is a powerful tool used to evaluate the performance of intrusion detection systems and other classification systems, such as spam filters or anomaly detection systems. It allows you to quantify the classification accuracy, that is, how well the system can distinguish one class of objects from another. In the context of intrusion detection systems, a correspondence matrix is used to evaluate the system's ability to correctly identify malicious and normal network traffic flows. It consists of four main indicators:

- True Positive (TP) is the number of correctly identified malicious streams. In other words, the system detected an attack, and this attack did take place;
- True Negative (TN) is the number of correctly identified normal flows when the system did not detect an attack and the attack was really absent;
- False Positive (FP) is the number of false positives when the system classifies normal traffic as malicious, which can block legitimate activity and create unnecessary noise for analysts;
- False Negative (FN) is the number of misses when the system classifies malicious traffic as normal, allowing attacks to go undetected.

The results of testing with different data sets are shown in Table 3. It should be noted that to ensure the reliability of the results, the data from the sets were not used in full.

**Table 3**

Quality metrics

Data set	TP	TN	FP	FN
KDDCup99	24354	2344	401	872
NSL-KDD	29435	2956	756	935
UNSW-NB15	1995	394	86	73
CICIDS-2017	81517	2354	2274	2250
ISCXVPN2016	7652	1855	198	157
SDN-Dataset	1752	1543	88	82

The most common metrics include accuracy, precision, recall, specificity, and F-measure. Accuracy shows the total proportion of correctly classified samples and is calculated as the ratio of the sum of true positive and true negative predictions to the total number of samples:

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \quad (23)$$

Precision, in turn, determines what proportion of samples classified as positive by the model are actually positive and is calculated as the ratio of true positive predictions to the sum of true positive and false positive predictions:

$$Precision = TP/(TP + FP) \quad (24)$$

Recall reflects what proportion of all positive samples were correctly identified by the model and is calculated as the ratio of true positive predictions to the sum of true positive and false negative predictions:

$$Recall = TP/(TP + FN) \quad (25)$$

Specificity, on the other hand, shows what proportion of all negative samples were correctly classified and is calculated as the ratio of true negative predictions to the sum of true negative and false positive predictions:

$$Specificity = (FP + FN)/(TP + FP + TN + FN) \quad (26)$$

The F-measure is a harmonic mean of accuracy and completeness and is often used as a single measure of model quality because it takes into account both aspects: the ability of the model to correctly classify positive samples and the ability to avoid false positive classifications. The choice of a specific metric depends on the task at hand and the relative importance of different types of errors:

$$F - score = (2 \times Recall \times Precision)/(Recall + Precision) \quad (27)$$

Performance indicators of testing using KDDCup99, NSL-KDD, UNSW-NB15, WSN-DS, and CICIoT2023 data sets are given in Table 4.

**Table 4**

Performance indicators

Data set	Accuracy	Precision	Recall	Specificity	F-score
KDDCup99	0.95	0.98	0.97	0.05	0.97
NSL-KDD	0.95	0.97	0.97	0.05	0.97
UNSW-NB15	0.94	0.96	0.96	0.06	0.96
CICIDS-2017	0.95	0.97	0.97	0.05	0.97
ISCXVPN2016	0.96	0.97	0.98	0.04	0.98
SDN-Dataset	0.95	0.95	0.96	0.05	0.95

The analysis of the proposed system on different data sets shows the following regularities. Accuracy reached 0.95 on the KDDCup99, NSL-KDD, CICIDS-2017 and SDN-Dataset datasets. On the ISCXVPN2016 set, the system showed the highest accuracy of 0.96, while on the UNSW-NB15

set, the system showed the lowest accuracy of 0.94. Precision remains high on all sets, varying between 0.95 and 0.98. The system shows the highest precision on KDDCup99 (0.98), while on UNSW-NB15 and SDN-Dataset, it shows slightly lower values – 0.96 and 0.95, respectively. Completeness on all sets is also high, ranging from 0.96 to 0.98, with the highest on ISCXVPN2016 (0.98), while other sets, including KDDCup99, NSL-KDD, CICIDS-2017, show identical completeness at 0.97. The F-measure varies between 0.95 and 0.98. The highest F-measure is shown by ISCXVPN2016 (0.98), while the lowest F-measure is recorded by SDN-Dataset (0.95). In general, the system shows consistently high quality indicators that do not fluctuate significantly depending on the data sets.

## 5. Conclusions

The article is focused on the development and implementation of a system for network traffic analysis using fuzzy logic. The basic idea is to apply fuzzification and fuzzy inference techniques to classify network packets based on their signatures. Using only the most informative parameters allows you to reduce the load on the system and increase the speed of data processing, which is important for the stable operation of the network. The developed system effectively determines whether to allow or deny data transmission based on fuzzy criteria.

One of the disadvantages of the proposed system is the lack of ability to detect attacks stretched over time. The method of cumulative analysis of anomalies can improve the method of detecting network attacks, which will allow more effective detection of long-lasting attacks that may remain unnoticed if each anomaly is considered separately. However, this is not relevant in the context of this work because the duration of connection to public or campus networks is short-term, and it is not advisable to implement attacks of this type.

Further research involves expanding the knowledge base by adding new rules to the fuzzy knowledge base to cover a wider range of possible network traffic scenarios. The possibility of integrating this system with other solutions in the field of cyber security is being considered, which will allow the creation of a more comprehensive and reliable approach to protecting networks.

## Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check; DeepL Translate in order to: some phrases translation into English. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

- [1] Cost of a Data Breach Report 2024, 2024. URL: <https://www.ibm.com/reports/data-breach>
- [2] Cybercrime To Cost World \$9.5 trillion USD annually in 2024, 2024. URL: <https://cybersecurityventures.com/cybercrime-to-cost-the-world-9-trillion-annually-in-2024/>
- [3] DDoS threat report for 2023 Q4, 2024. URL: <https://blog.cloudflare.com/ddos-threat-report-2023-q4>
- [4] Biggest cyber threats for financial institutions in 2023, 2023. URL: <https://www.blazeinfosec.com/post/cyber-threats-for-finance-2023/>
- [5] A. Kiflay, A. Tsokanos, M. Fazlali, R. Kirner, Network intrusion detection leveraging multimodal features, *Array*, 2024, volume 22, doi: 10.1016/j.array.2024.100349.
- [6] X. Gu, G. Howells, H. Yuan, A soft prototype-based autonomous fuzzy inference system for network intrusion detection, *Information Sciences*, 2024, volume 677.
- [7] S. Kushal, B. Shanmugam, J. Sundaram et al, Self-healing hybrid intrusion detection system: an ensemble machine learning approach, *Discov Artif Intell* 4, 2024, volume 28.
- [8] L. D. Tsobdjou, S. Pierre and A. Quintero, An Online Entropy-Based DDoS Flooding Attack Detection System With Dynamic Threshold, *IEEE Transactions on Network and Service*

- Management, 2022, volume 19, no 2, pp. 1679-1689, doi: 10.1109/TNSM.2022.3142254.
- [9] M. Alduailij et al, Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method, *Symmetry*, 2022, volume 14.
- [10] Loïc D. Tsobdjou, Samuel Pierre, Alejandro Quintero, An Online Entropy-Based DDoS Flooding Attack Detection System With Dynamic Threshold, *IEEE Trans. on Netw. and Serv. Manag.*, 2022, volume 19, pp. 1679–1689, doi: 10.1109/TNSM.2022.3142254
- [11] M. Hajimaghsoodi, R. Jalili, RAD: A Statistical Mechanism Based on Behavioral Analysis for DDoS Attack Countermeasure, *IEEE Transactions on Information Forensics and Security*, 2022, volume 17, pp. 2732-2745, doi: 10.1109/TIFS.2022.3172598
- [12] A. Borisova, Network attack recognition using fuzzy logic, *ETR*, 2024, volume 2, pp. 55–60, doi: 10.17770/etr2024vol2.8054.
- [13] S. R. Zahra, M. A. Chishti, A. I. Baba, F. Wu, Detecting Covid-19 chaos-driven phishing/malicious URL attacks by a fuzzy logic and data mining-based intelligence system, *Egyptian Informatics Journal*, 2022, volume 23, issue 2, pp. 197-214, doi: 10.1016/j.eij.2021.12.003.
- [14] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, R. Kocaoğlu, Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking, *Electronics*, 2021, volume 10.
- [15] St. K. Wanjau, G. M. Wambugu, G. Nd. Kamau, SSH-Brute Force Attack Detection Model based on Deep Learning, *International Journal of Computer Applications Technology and Research*, 2021, volume 10, issue 01, pp. 42-50.
- [16] S. Naseer, Ali R. Faizan, PD. Dominic, Y. Saleem, Learning Representations of Network Traffic Using Deep Neural Networks for Network Anomaly Detection: A Perspective towards Oil and Gas IT Infrastructures, *Symmetry*, 2020, volume 12, doi: 10.3390/sym12111882
- [17] V. Titova, Y. Klots, V. Cheshun, N. Petliak. A.-B.M. Salem, Detection of network attacks in cyber-physical systems using a rule-based logical neural network, *CEUR Workshop Proceedings*, 2024, volume 3736, pp. 255–268.
- [18] M. Rodríguez, Á. Alesanco, L. Mehavilla, J. García, Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection, *Sensors*, 2022, volume 22, doi: 10.3390/s22239326
- [19] V. Khoroshko, M. Brailovskyi, M. Kapustian, Multi-criteria assessment of the correctness of decision-making in information security tasks, *Computer systems and information technologies*, 2023, volume 4, pp. 81-86, doi: doi.org/10.31891/csit-2023-4-11
- [20] N. Petliak, Y. Klots, V. Titova, V. Cheshun, A. Boyarchuk, Signature-based Approach to Detecting Malicious Outgoing Traffic, *CEUR Workshop Proceedings*, 2023, volume 3373, pp. 486-506.
- [21] Blacklist IP Addresses Live Database, 2024 URL: [https://myip.ms/browse/blacklist/Blacklist\\_IP\\_Blacklist\\_IP\\_Addresses\\_Live\\_Database\\_Real-time](https://myip.ms/browse/blacklist/Blacklist_IP_Blacklist_IP_Addresses_Live_Database_Real-time)
- [22] Export all blocked IPs, 2023. URL: <https://www.blocklist.de/en/export.html>
- [23] M. Voskoglou, *Fuzzy Sets, Fuzzy Logic and Their Applications*, 2021. 282 pages. doi: 10.3390/books978-3-0365-7375-5
- [24] L. Zadeh, R. Aliev, *Fuzzy Logic Theory and Applications*, World Scientific Publishing Company, 2018 URL: <https://www.perlego.com/book/858249/fuzzy-logic-theory-and-applications-part-i-and-part-ii-pdf>.
- [25] O. Savenko, S. Lysenko, A. Kryshchuk, Y. Klots. Proceedings of the 7-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Berlin, September 12–14, 2013. Berlin, 2013. Pp. 363–368.
- [26] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A. Nicheporuk. A Technique for detection of bots which are using polymorphic code. *Communications in Computer and Information Science*. 2014.Vol. 431. PP.265-276, ISSN: 1865-0929.
- [27] L.C. de Barros, R.C. Bassanezi, W.A. Lodwick, Notions of Fuzzy Logic. In: *A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics*, Studies in Fuzziness and Soft Computing, 2024, volume 432, doi: 10.1007/978-3-031-50492-1\_4