Environment. Technology. Resources. Rezekne, Latvia
Proceedings of the 15th International Scientific and Practical Conference. Volume II, 55-60

Network attack recognition using fuzzy logic

Angela R. Borisova
Computer Systems and Technologies Department
National Military University
"Vasil Levski", Faculty "Artillery, Air Defense and CIS"
Shumen, Bulgaria
arborisova@nvu.bg

Abstract. **The following research paper presents a fuzzy logic system model related to classifying network traffic as malicious or normal. The relevance of the problem stems from the increasingly widespread worldwide problem, namely cyber threats against various companies, organizations, individuals, etc. and at the same time the use of artificial intelligence systems as a means of detecting and preventing various types of cybercrime. To accomplish the task, several basic work methods are followed: first, the development goal is defined - building a fuzzy logic system that supports and automates decision-making about the type of network traffic (malicious or normal traffic), second, appropriate software is selected to perform the task, in this case MATLAB and specifically the Fuzzy Logic Designer toolbox, third, the actual system is built consisting of pre-obtained network traffic inputs that are taken from a pre-collected and compiled .pcap file (the data in it are captured and modified to contain only some network information fields from the set of packets necessary for the experiment to run successfully), the system itself consists of nine input linguistic variables, one output linguistic variable and a knowledge base (the core of the project, namely if-then rules). The studied system is compared with other similar fuzzy output systems of other researchers. Based on this, it is concluded that the approach proposed in the present work to categorize network traffic, based on pre-selected network information fields, in collaboration with other means of cyber protection gives very good results in the context of cyber security.**
The present project proposes a fuzzy inference system to classify network packet types and detect TCP-SYN attack.
The fully built fuzzy source system provides a different perspective to solve the present problem by defining the abstract solution and facilitating the work of specialized personnel charged with such tasks by automating the process of providing an adequate solution regarding the legitimacy of network traffic.
Keywords: artificial intelligence, fuzzy logic system, network traffic analysis, TCP-SYN flood

I. INTRODUCTION

The urgency of the problem lies in the ever-growing cyberthreats, both with regard to large Internet giants and commercial, non-profit and governmental organizations, as well as with regard to the average user. The need for a quick and adequate response to protect their data from bad actors forces the systems used to provide protection to rely on the increasing power of artificial intelligence [9].

Security comes with additional question, stressing out cybersecurity as Internet and other media is on the infrastructure core. Attacks by hackers against various organizations and individuals in general are becoming more massive and widespread, and potential attackers can be both organized criminals and amateurs who have decided to play a prank on someone [2]. A few examples can be given in this regard: Continuously sending the same messages to multiple email addresses at random (this action is known as social engineering [3]), with the first goal being to determine whether a real email address exists among the selected, and secondly, whether its owner is susceptible to fraud (most often the message sent contains a malicious attachment, with the help of which, when clicked, users' devices are infected, and information such as personal data, passwords, etc. becomes available to ill-wishers). Another example is direct attempts to hack accounts in various social networks, as well as sending messages of the type: "You need to change your password at bank X", suggesting to customers that this is a preventive measure as part of a bank to protect clients' data and finances, and more. To deal with these and other similar problems, software protection tools such as anti-virus systems, firewalls and, for some time, artificial intelligence systems are used. Working together, protection tools seem successful in a decent manner. But to determine what data arrives over the network to users, the data must be separated into good data and bad labels.

It is appropriate to use software that makes such an assessment of the continuous flow of network packets. There are many possible ways to analyse, evaluate and predict the relevant network traffic, but in this research scenario, one particular system is considered - a fuzzy logic system (FLS - doubtful logic [10]), based on a fuzzy source system, which is designed to classify different types of

Print ISSN 1691-5402
Online ISSN 2256-070X
https://doi.org/10.17770/etr2024vol2.8054

attacks in good and bad (this system uses a ready-made data set called KDD CUP 99) [7] , and the current project aims to create a fuzzy logic system that supports the automation process of classifying a given traffic as normal or malicious traffic, as the data set used for logging is taken from pre-recorded network traffic in a .pcap file and pre-processed using an artificial intelligence algorithm. In this line of thought, fuzzy logic systems are used to develop various systems that aim to provide precise estimation of real-time signal processing [1]. Artificial neural networks and fuzzy systems for cybersecurity are used to gather information and enrich data about new threats and vulnerabilities, which is of upmost importance for cyber systems to adapt to ever-changing conditions.

II. MATERIALS AND METHODS

For the needs of the present research project, a system called "NETWORK TRAFFIC_sample" was created in MATLAB using the Fuzzy Logic Designer tool, consisting of 9 input linguistic variables, one output and 10 Mamdani-type rules at the beginning of the experiment. (subsequently the rules were increased to 51) to support the aims of the experiment. The input linguistic variables are: IP source port, IP destination port, IP destination port buffer, TCP source port, TCP destination port, TCP sequence number, TCP synchronize flag, TCP counter, Timestamp. The output linguistic variable label is “Malicious traffic”. The system works with weight coefficients ranging from 0 to 1, similar to neural networks, where, depending on the activation function, the output is most often in the same range. Rule is defined: 1 meaning the given network traffic is certainly malicious and 0 meaning that the given network traffic is certainly normal (in the particular case, the membership functions for the output linguistic variable "Malicious traffic" are low risk and high risk). If the weight coefficients have values from 0 to 0.4, they will again belong to the set of normal traffic, i.e. low risk, otherwise if values exceed 0.4, i.e. become 0.5 and greater up to and including 1, they will be categorized in the malicious traffic set i.e. high risk.

Each input linguistic variable in turn consists of a different number of membership functions that satisfy the needs of the system. What they have in common is that they are of the same type called the Generalized Bell-shaped MF (MF - Membership Function), also called the Cauchy MF. This model was chosen because of its many advantages, allowing for more accurate final results and as few similarities as possible between the elements in the individual membership functions. Thus, allowing them to be classified into the correct "group", i.e. their falling into the exact belonging function amongst the predefined ones. Cauchy MF allows changing the shape of the function and the size of the function thanks to the three main parameters that make it up – *a* and *b*, responsible for the width and *c*, responsible for the center point of the membership function. Generalized bell MF is a combination of Gaussian and trapezoidal function. On fig. 1 a graph composed of the membership functions of the linguistic variable *IP source port* is shown, and on Fig. 1.1 the automatically generated MATLAB source code of the same variable is shown.

Based on these two databases, it can be summarized that the membership functions are named using four example IP addresses, concrete "192.168.6.2", "192.168.6.3", 192.168.6.4, "192.168.6.5". As the membership function of the first IP address "192.168.6.2" is located in the range from 0 to 1, the membership function of the second IP address "192.168.6.3" is located in the range from 1 to 2, the membership function of the third IP packet "192.168.6.4" is located in the range 2 to 3, ... and the membership function of the fourth IP address "192.168.6.5" is located in the range 3 to 4.

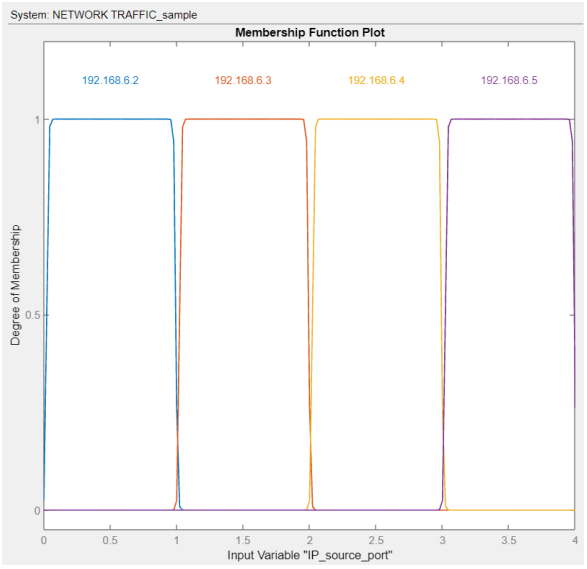

Fig. 1. IP source port Membership function.

```
[Input1]
Name='IP_source_port'
Range=[0 4]
NumMFs=4
MF1='192.168.6.2':'gbellmf',[0.485899 41.4327 0.50796]
MF2='192.168.6.3':'gbellmf',[0.485899 41.4327 1.50796]
MF3='192.168.6.4':'gbellmf',[0.485899 41.4327 2.50796]
MF4='192.168.6.5':'gbellmf',[0.485899 41.4327 3.50796]
```

Fig. 1.1. Listing of IP source port Membership - source code in MATLAB.

IP source port is only one component of the selected total of 9 elements of a single packet involved in network traffic.

Like *IP_source_port*, *IP_dest_port* membership functions (since they are again 4 IP addresses) are in range from 0 to 4. MFs are: "192.168.6.254", "192.168.10.10", "192.168.10.11", "192.168.10.12". *IP_dest_port_buffer* accordingly has two MFs: "Unknown_IP" – occupying the range from 0 to 1 and "Known_IP" – from 1 to 2. *TCP_source_port* consists of 5 membership function: "1024", "1025", "1026", "1060", "2000", similarly from 0 to 5. *TCP_dest_port* consists of 4 MFs and they are: "10", "80", "1023", "8080" ranging from 0 to 4. The next *TCP_Sequence number* consists of 10 MFs – "1", "2"…, "10", in a range from 0 to 10. Then comes *TCP_syn_flag* with 2 MFs – "0" (it means false, the flag is down), "1" (it means true, the flag is raised) in a range from 0 to 2. The next *TCP_counter* with 2 MFs – "under _10", "10_and_more" from 0 to 2 again. And the last one is *TIME* (timestamp) with 3 MFs - "under_1_second", "1_second", "over_1_second", from 0 to 3 similarly.

The designed system considers these 9 packet information fields as basic control data. What is monitored

is how many times the TCP-SYN flag (TCP synchronize flag) is raised, i.e. how many times the IP source port (user *X*) communicated (a session had been created) with the IP destination port (legitimate server *Y*). Rules are built in that if the number of sessions (TCP_counter) reaches 10 or more from a specific client to a specific monitored server within a 1.00 second time-range (TIME), then malicious intent is present (a network attack called TCP syn flood), i.e. user X purposefully sends the same request to server Y in order to achieve Denial of Service (DoS) for example. In such case, the role of IP destination port buffer is to save the recipient's IP address if the TCP syn flag has been detected (the flag is raised, it is 1) and being sent to server from the same actor least twice in less than 1.00 second time-range. TCP sequence number reports the sequence number of the session, and TCP counter counts the total number of packets passed through the system when establishing a connection between the client-server model users. To determine whether a packet is safe or harmful, sample data for the above-mentioned information is provided. To demystify the FLS idea, 10 if-then rules are indicated on fig. 2 and fig. 2.1.

	Rule	Weight	Name
1	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	1	rule1
2	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.9	rule2
3	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.8	rule3
4	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.7	rule4
5	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.6	rule5
6	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.5	rule6
7	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.5	rule7
8	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.6	rule8
9	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	0.7	rule9
10	If IP_source_port is 192.168.6.2 and IP_dest_port is 192....	1	rule10

Fig.2. Network traffic rules.

If IP_source_port is 192.168.6.2 and IP_dest_port is 192.168.6.254 and IP_dest_port_buffer is Known_IP and TCP_source_port is 1060 and TCP_dest_port is 80 and TCP_Sequence_number is 10 and TCP_syn_flag is 1 and TCP_counter is 10 and more and TIME is 1_second then MALICIOUS_traffic is high_risk

Fig.2.1. Rule classifying a high risk with a weight of 1, generated by MATLAB.

It can be seen from fig. 2. that all rules are set with weight factors between 0.5 and 1, ensuring that the selected set of rules belongs to the desired output. And fig. 2.1. demonstrates what a rule should look like such as certainly presenting the traffic to be malicious. Obvious point is having a severity factor of 1, the TCP-SYN flag is raised 10 times according to the TCP sequence number, and the IP address of the recipient and sender respectively are the same.

These 10 rules are far from providing all the possibilities for the state of the system, but they are a basic example of how it should work. For example, a rule like:

If (IP_source_port is 192.168.6.3) and (IP_dest_port is 192.168.10.10) and (IP_dest_port_buffer is Known_IP) and (TCP_source_port is 1024) and (TCP_dest_port is 8080) and (TCP_sequence_number is 10) and (TCP_syn_flag is 1) and (TCP_counter is 10 and more) and (TIME is 1_second) then (Malicious_traffic is high_risk) (1),

will also work correctly with a weight factor of 1, since the above requirement of malicious traffic - high risk, is met and the weight factor is in the correct range from 0.5 to 1 inclusive.

A similar rule, now with a weighting factor of 0, i.e. guaranteeing that the given set of rules certainly does not belong to the desired output, looks like this:

If (IP_source_port is 192.168.6.5) and (IP_dest_port is 192.168.10.10) and (IP_dest_port_buffer is Unknown_IP) and (TCP_source_port is 1025) and (TCP_dest_port is 1023) and (TCP_sequence_number is 1) and (TCP_syn_flag is 1) and (TCP_counter is 1 and more) and (TIME is over_1_second) then (Malicious_traffic is high_risk) (0)

Here, the role of the weighting factor in the construction of these rules is extremely important. In the particular case, a weight set to 0 ensures the situation that the probability of the traffic being malicious is low. The same result can be achieved if the membership function is changed from high risk to low risk and the weight factor becomes 1.

In order to obtain more accurate results, a few more experimental setups are made, and the number of rules in the knowledge base becomes 51.

Fuzzy output systems are abstract, unorthodox [5] and can be modified according to the understandings and needs of artificial intelligence through fuzzy reasoning [8].

On fig. 3 the constructed output is presented, consisting of two membership functions, with the help of which the traffic is classified as *good* or *bad*, respectively. Fig. 3.1. presents the automatically generated source code of the output linguistic variable from the MATLAB software.

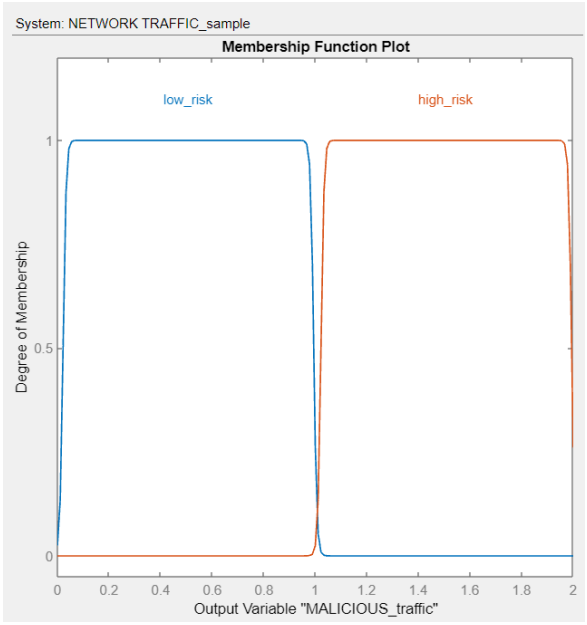

Fig. 3. Degree of membership for 1 linguistic output variable.

```
[Output1]
Name='MALICIOUS_traffic'
Range=[0 2]
NumMFs=2
MF1='low_risk':'gbellmf',[0.485899 41.4327 0.50796]
MF2='high_risk':'gbellmf',[0.485899 41.4327 1.50796]
```

Fig. 3.1. Output - source code in MATLAB.

Judging by fig. 3. and fig. 3.1. it can be summarized that the output of the system can be only two options low risk - harmless traffic or high risk - bad traffic, with the membership function "no risk" extending in the range from 0 to 0.99, and the membership function "high risk" covers the range from 1.0 to 2.

III. Results and Discussion

As a result of the conducted experiment (with 10 if-then rules), a simulation of a system with fuzzy output was built, giving information about the type of specifically selected packets of network traffic (whether there is cause for alarm in them, i.e. the presence of a malicious action or not). When testing the already built fuzzy logic system, sample values were selected for the inputs which are defined in the following data array: [0.2321;0.5893;1.903;3.606;1.646;9.69;1.673;1.743;1.606]. This array represents the membership functions of each of the 9 pre-selected network packet information data for monitoring, with the resulting score having a value of 1.51 and indicating that the traffic in this case is malicious, as the membership function of this point is ' high risk' (because MF 'high risk' ranges from 1 to 2). The result of this can be seen on fig. 4.

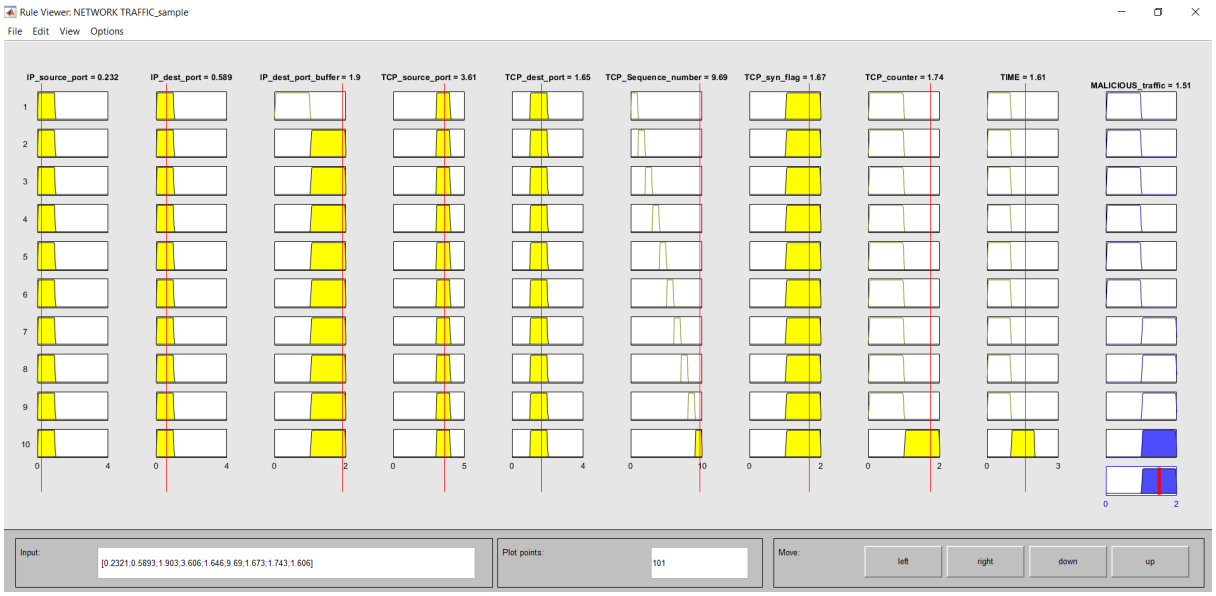

Fig. 4. The result, from the built 10 rules, showing the outcome desired by the expert - with weight coefficients between 0.5 and 1.

In order for the reach to be more comprehensive, 7 more attempts were made, and the if-then rules were supplemented to 51. The results of the experiment can be seen in table 1 and table 2. As table 2 presents the interpreted results of table 1.

Table 1 Multiple Testing Of Fuzzy Output – With 51 If-Then Rules

IP_source_port	IP_dest_port	IP_dest_port_buffer	TCP_source_port	TCP_dest_port	TCP_Sequence_number	TCP_syn_flag	TCP_counter	Timestamp	Malicious_traffic
1.482	1.661	0.5575	3.695	3.699	0.04425	1.584	0.00885	2.456	0.662
2.696	2.544	0.5575	0.1096	0.4696	1.36	1.296	1.798	1.396	0.903
3.391	3.526	1.904	0.2851	0.9913	10	2	2	1.213	1.24
1.617	3	1.765	0.943	1.861	9.079	1.661	1.833	0.7696	1.07
2.591	1.105	0.7217	0.4167	0.01739	3.904	1.365	1.061	2.596	0.961
2.522	2.93	1.452	3.311	2.417	7.412	1.435	1.325	0.326	1.04
2.625	2.589	1.779	0.5973	3.133	7.478	2	1.336	0.1726	1.05

TABLE 2 INTERPRETATION OF TABLE 1

IP source port	IP dest port	IP dest port buffer	TCP source port	TCP dest port	TCP Sequence number	TCP syn flag	TCP counter	TIME (Timestamp)	Malicious traffic
192.168.6.3	192.168.10.10	Uknown_IP	1060	8080	1	1	under_10	over_1_second	low_risk
192.168.6.4	192.168.10.11	Uknown_IP	1024	10	2	1	10_and_more	1_second	low_risk
192.168.6.5	192.168.10.12	Known_IP	1024	10	10	1	10_and_more	1_second	high_risk
192.168.6.3	192.168.10.12	Known_IP	1024	80	10	1	10_and_more	under_1_second	high_risk
192.168.6.4	192.168.10.10	Uknown_IP	1024	10	4	1	10_and_more	over_1_second	low_risk
192.168.6.4	192.168.10.11	Known_IP	1060	1023	8	1	10_and_more	under_1_second	high_risk
192.168.6.4	192.168.10.11	Uknown_IP	1024	8080	8	1	10_and_more	under_1_second	high_risk

The "Materials and Methods" section describes the ranges in which the respective membership functions of each input linguistic variable and the output linguistic variable lie, and table 2 is constructed based on them. Between the first column and the first row of table 1, the numeric value 1.482 for IP_source_address is presented, which means that the second IP address is specified (since it is in the range 1 to 2), which is "192.168.6.3". Between the second column and the first row of table 1 is the numeric value 1.661 for IP_dest_address, which again shows the second IP address, but this time to the recipient and it is "192.168.10.10". Between the third column and the first row of table 1 is the value 0.5575 for IP_dest_buffer, which means that the value falls in the range 0-1 and the membership function is "Uknown_IP". The intersection of the fourth column and the first row of table 1 expresses the value 3.695 of the input linguistic variable TCP_source_port, i.e. the membership function is "1060" (since it is in the range 3-4). The intersection of the fifth column and the first row of table 1 represents the value 3.699 of TCP_dest_port, so the MF is "8080" (falls in range 3-4). Between the sixth column and the sixth row of table 1, the resulting value 0.04425 is presented, which refers to the TCP_Sequence_number and falls in the range 0-1, which means that the membership function in this situation is "1". The value between the seventh column and the first row of table 1 of the linguistic variable TCP_syn_flag is "1.584" respectively, the resulting value falls in the range 1-2, i.e. MF is "1", i.e. the session is complete. The value between the eighth column and the first row of table 1 of TCP_counter is 0.00885, i.e. MF is "under_10". Between column number 9 and row 1 of table 1 is represented the value 2.456 for TIME (timestamp) - with a total of three membership functions, this value falls in the range 2-3, i.e. "over_1_second". Between the last 10 column and the first row of table 1, the result of the search is concluded, namely whether the traffic is malicious or not. At a value of 0.662 MF is "low_traffic". The interpretation of the remaining 6 lines is analogous.

A detailed examination of Tables 1 and 2 shows that the results obtained are quite good, i.e. the created system works correctly.

A brief comparison between the problem at hand and two other similar problems follows.

Shanmugavadivu's [7] fuzzy system uses 34 input linguistic variables representing a part of the KDD CUP 99 data set, based on which multiple if-then rules are built, of which a part of them determined by a filter is finally used in forming the final exit. Overall, the system is reliable, but from the point of view of having some unused input linguistic variables and ultimately unused if-rules, this can lead to not so good efficiency and not so good speed. On the other hand, Slavyanov's [1] fuzzy system has the required number of rules and the required number of inputs to function properly. In general, the proposed system in this paper uses pre-collected and pre-processed input data that are as accurate as necessary to properly build the algorithm. The rules are properly defined, initially the system was tested with only 10 rules, but subsequently with 51 rules. As the number of rules increases, the accuracy of the algorithm increases.

Striving to take the best of the two systems with which the current one is being compared, and accordingly to avoid approaches that are not well suited to the current problem, an optimal system was created.

Analysing the obtained results from fig. 4, table 1, table 2 and in view of the comparison made with other two similar systems, it can be concluded that the proposed system is efficient, stable [6] and accurate.

IV. CONCLUSIONS

The project that is built in this paper answers the question of how a fuzzy inference system, can be built to successfully classify output as malicious or non-malicious by using weighting factors.

In the context of cybersecurity discussed up to this point, it can be summarized that FLS are a powerful tool for building systems showing the principle of separating data passing through the network into good content and bad content i.e. fuzzy systems are an essential element in an overall system supporting the cyber defense of various organizations and individuals.

The future development of artificial intelligence algorithms is inevitable to continue the evolution in cybersecurity. The introduction of fuzzy output algorithms in the context of network security has demonstrated an

increase in its accuracy and its effectiveness over time [4] and the improvement of security measures in various systems.

The end result is automating the process of deciding on the type of network traffic, analysing and correctly classifying network data by creating a set of rules.

The proposed system can be used in various areas of network security, including optimization of network resources, detection of malicious actions and protection against cyberattacks, and it can be integrated with other systems for even more reliable protection and improvement of response time by specialists in this field.

ACKNOWLEDGEMENTS

This paper is created in favor of Bulgarian National Scientific program "Security and Defense", Ministry Council decision No 731/21.10.2021, Agreement No Д01-74/19.05.2022.

REFERENCES

[1] K. O. Slavyanov, "Fuzzy logic procedure for drawing up a psychological profile of learners for better perception in courses," in Proceedings of the 12th International Scientific and Practical Conference, July 20-26, 2019, Rezekne, Latvia: Rezekne Academy of technologies, 2019. Available: http://dx.doi.org/10.17770/etr2019vol2.4073, [Accessed: January, 5, 2024]

[2] L. G. Nikolov and K. O. Slavyanov, "On the contemporary cybersecurity threats", International scientific journal "Security & future", vol. 1, pp. 111-113, 2017. Available: https://stumejournals.com/journals/confsec/2017/3/111.full.pdf, [Accessed: January, 10, 2024]

[3] L. G. Nikolov, "Social engineering as a high cybersecurity threat", International scientific journal "Security & future", vol. 3, pp. 106-108, 2019. Available: https://stumejournals.com/journals/confsec/2019/3/106.full.pdf. [Accessed: January, 11, 2024].

[4] M. I. Mihailescu, S. L. Nita, M. Rogobete and V. Marascu, "Unveiling Threats: Leveraging User Behavior Analysis for Enhanced Cybersecurity," in Proceedings of 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), June 29-30, 2023, Bucharest, Romania: Institute of Electrical and Electronics Engineers, 2023. Available: https://doi.org/10.1109/ECAI58194.2023.10194039. [Accessed: January, 11, 2024].

[5] B. Singh and A. K. Mishra, "Fuzzy logic control system and its applications", International Research Journal of Engineering and Technology (IRJET), vol. 02, pp. 742-746, 2015. Available: https://www.irjet.net/archives/V2/i8/IRJET-V2I8104.pdf. [Accessed: January, 12, 2024].

[6] G. Sharma, V. Raju, H. Dhall, P. Sudan, B Reddy, I. Alpackaya, "Fuzzy Logic-Based Energy Management in Smart Grids for Renewable Integration," International Conference on "Advanced Materials for Green Chemistry and Sustainable Environmen.t" (AMGSE-2024), vol. 511, pp. 1-14, 2024, Available: https://www.e3s-conferences.org/articles/e3sconf/pdf/2024/41/e3sconf_amgse2024_01013.pdf. [Accessed: January. 12, 2024].

[7] R. Shanmugavadivu and N. Nagarajan, "Network Intrusion Detection System using Fuzzy Logic", Indian Journal of Computer Science and Engineering (IJCSE), vol.2, no.1, pp. 101-111, 2015, Available: https://ijcse.com/docs/IJCSE11-02-01-034.pdf, [Accessed: February, 1, 2024].

[8] L. C. Barros , R. C. Bassanezi , W. A. Lodwick, A First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics: *Notions of Fuzzy Logic,* Vol. 432. Cham: Springer, 2024, pp. 53-78. https://doi.org/10.1007/978-3-031-50492-1_4.

[9] P. Vähäkainu, M. Lehto, Artificial Intelligence and Cybersecurity: *Use of Artificial Intelligence in a Cybersecurity Environment*. Cham: Springer, 2022, pp. 3-27. https://doi.org/10.1007/978-3-031-15030-2_1.

[10] M. Soltanifar, H. Sharafi, F. H. Lotfi, W. Pedrycz, T. Allahviranloo, Preferential Voting and Applications: Approaches Based on Data Envelopment Analysis. Studies in Systems, Decision and Control: *Introduction to Fuzzy Logic*., vol 471. Cham: Springer, 2023, pp.31-45. https://doi.org/10.1007/978-3-031-30403-3_3.