A Fuzzy Neural Network Approach for Intelligent Environmental Monitoring and Anomaly Detection in IoT Networks

Majidov Anvarkhon[1][0009-0009-3797-1533]

[1] National University of Uzbekistan named after Mirzo Ulugbek, Tashkent, Uzbekistan
anvarxonmajidov@gmail.com

Abstract. The rapid deployment of Internet of Things (IoT) technologies has significantly expanded the capabilities of large-scale environmental monitoring systems. How-ever, the heterogeneity of sensor devices and the openness of communication channels increase the risk of anomalous data transmission and malicious interference. This paper proposes an intelligent environmental monitoring approach based on a fuzzy neural network for anomaly detection in IoT networks. The proposed method combines fuzzy logic and a multilayer neural network to analyze sensor data packets using nine network traffic features. Adaptive adjustment of membership functions during neural network training enables probabilistic evaluation of normal and anomalous behavior. The approach is integrated into a software-defined networking (SDN) architecture, allowing real-time traffic analysis, node validation, and dynamic synthesis of packet filtering rules. Simulation results demonstrate high anomaly detection accuracy with moderate processing overhead, confirming the effectiveness of the proposed solution for secure and scalable environmental monitoring.

Keywords: Environmental monitoring, Internet of Things, fuzzy logic, neural networks, anomaly detection, software-defined networking.

1 Inroduction

The rapid adoption of IoT technologies in various domains has significantly increased machine-to-machine interactions, leading to the collection and exchange of large volumes of data from spatially distributed sensors and devices. While this has enabled intelligent environmental monitoring and improved operational efficiency, it has also introduced new challenges in information security. The growing number of IoT devices, coupled with their connectivity over distributed telecommunication channels, has escalated the risk of cyber-attacks and anomalous behavior in IoT networks [1, 2]. Among the most critical threats in such networks are flooding and Distributed Denial of Service (DDoS) attacks, which generate excessive requests to network nodes, potentially disrupting the normal operation of critical Environmental Monitoring systems [3].

To address these challenges, intelligent Environmental Monitoring systems based on fuzzy logic and neural networks have been proposed. These systems are capable of analyzing, recognizing, and classifying network traffic patterns to detect anomalies in real time. In critical infrastructures, such as road transport systems, environmental monitoring involves the collection, processing, and storage of data from multiple sources, including photo radar complexes, video surveillance cameras, and other IoT sensors [4, 5]. Such data provides information about traffic conditions, accidents, and other environmental events, which is essential for ensuring public safety under concepts like “Safe City” and “Safe Roads.”

Recent advances in SDN have introduced new possibilities for intelligent Environmental Monitoring and anomaly detection in IoT networks [6, 7]. SDN provides centralized control of network switches, separating the logical management layer from the data forwarding layer, which facilitates dynamic reconfiguration, traffic analysis, and threat mitigation [8–10]. However, the centralized nature of SDN introduces limitations, particularly in terms of scalability and responsiveness during distributed attacks. In the case of large-scale DDoS attacks, the SDN controller may struggle to process incoming requests and update flow table rules in a timely manner [11, 12]. To overcome these limitations, SDN architectures are increasingly integrated with machine learning, fuzzy logic, neural networks, and statistical analysis techniques for proactive detection and mitigation of network anomalies [2, 13].

Several studies have explored combining SDN with advanced computational methods for IoT security. For example, platforms have been proposed for real-time detection of DDoS attacks using SDN in corporate networks [14], while others integrate SDN, blockchain, and network function virtualization to enhance IoT network security [15]. Such approaches address critical challenges, including low detection efficiency, lack of standardization, and high latency during large-scale attacks. Additionally, SDN-based solutions for low-power sensor networks enable real-time anomaly detection with minimal computational overhead [16]. Other works have explored Environmental Monitoring flooding attacks in SDN controllers’ data caches [17] and securing heterogeneous networks using blockchain-based distributed ledgers for data integrity and authentication [18].

Building on these insights, this article presents a secure and intelligent IoT Environmental Monitoring system that integrates SDN technology, fuzzy logic, and a multilayer neural network for real-time environmental monitoring and anomaly detection. The proposed system leverages distributed ledger technology for secure data storage, smart contracts for validating network nodes, synthesis of flow table rules, and comprehensive traffic analysis and filtering. This approach enables robust detection of network anomalies, mitigates cyber threats, and ensures reliable operation of critical IoT Environmental Monitoring infrastructures.

2 Materials and methods

The main function of the subsystem for protecting operations on collecting and transmitting network traffic from distributed nodes in the Environmental Monitoring

system is to analyze data packets from authorized nodes in order to detect atypical packets with possible malicious content. The architecture of the developed traffic analysis subsystem includes a parser, an analyzer, a clustering module with a neural network, and a logging module (Fig. 1).

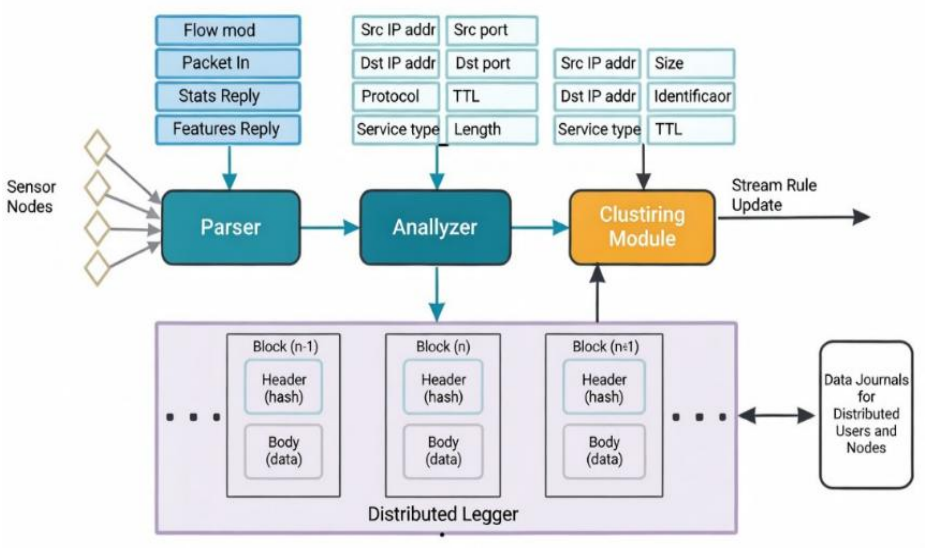

Fig. 1. Subsystem for analysis and detection of atypical data packets

Parsing and intelligent analysis modules are required to detect data packets with characteristics different from those usually transmitted by authorized nodes. The analysis procedure is implemented in the SDN network controller and allows identifying atypical behavior of sensor nodes and user mobile devices. Packets are selected by the parser and transmitted to the analyzer. The analysis module extracts packet characteristics from the headers of protocol data units. The characteristics are used to cluster data units in the feature space in order to determine the degree of similarity with the centroids of clusters of typical and atypical packets formed during the training of the neural network. Clustering allows detecting facts of modification of data packet headers and compromise of sensor nodes. For sets of characteristics of atypical packets, the controller creates filtering rules in flow tables, and for normal packets – retransmission rules.

The packet clustering method is implemented in a combined manner using fuzzy logic and a neural network. An example of such an approach is the intrusion detection technology in the ANFIS (Adaptive Network-based Fuzzy Inference System) [19]. To synthesize the feature space and perform clustering, it is proposed to use nine characteristics extracted from the headers of IP packets and TCP segments, namely the

source IP address (IP_src), destination IP address (IP_Dst), header length (Head_Lng), total packet length (Total_Lng), fragment identifier (Frag_Id), packet lifetime (TTL), service type (Srv_type), sender (Port_Srv) and recipient (Port_Dst) protocol ports. The initial training of the neural network is performed based on the sets of characteristics of packets from unauthorized nodes, and the identified sets of characteristics of atypical packets are subsequently used for training. The parameters of the function of packet membership in clusters are configured using a neural network training algorithm, and the conclusion about membership and the degree of similarity with cluster centroids is formed using a fuzzy logic apparatus. The fuzzy clustering method is based on the fuzzy c-means algorithm [20] and includes the following steps:

1. The number of packet clusters M is set, which is then adjusted during the training process, the degree of fuzziness of the objective function $m > 1$ is selected.

2. The input sets of packet characteristics are feature vectors $X_j (j = 1, \dots, N)$. The vector defines a point in space that can belong to clusters with centroids $C^{(k)} (k = 1, \dots, M)$ and a probability membership function ${\mu_{X_j}}^{(k)}$, where $0 < {\mu_{X_j}}^{(k)} < 1$, $\sum_{k=1}^{M} {\mu_{X_j}}^{(k)} = 1$, which acts as the degree of similarity to the centroid and is defined as the distance ${D_{X_j}}^{(k)}$.

3. The points are randomly distributed among the clusters. The distribution of points is determined by the matrix of degrees of similarity with centroids in the feature space with nine coordinates $(x_{i1}, x_{i2} x_{i3} x_{i4} x_{i5} x_{i6} x_{i7} x_{i8} x_{i9})$.

4. The coordinates of the centroids of clusters $C^{(k)} (k = 1, \dots, M)$ are calculated by calculating the average proximity of cluster points:

$$C_k = \frac{\sum_{k=1}^{M} ({\mu_{X_j}}^{(k)} X_j)}{\sum_{k=1}^{M} ({\mu_{X_j}}^{(k)})}. \quad (1)$$

5. The distances between the points and the cluster centroids are determined:

$${D_{X_j}}^{(k)} = \sqrt{\| X_j - C^{(k)} \|^2}. \quad (2)$$

6. The degrees of membership of the points in the clusters are recalculated and the distribution matrix of the points is updated:

$$\mu_{X_j}{}^{(k)} = \frac{1}{\left(\sum_{k=1}^{M} \frac{D_{X_i}{}^{(k)}}{D_{X_j}{}^{(k)}}\right)^{2/m-1}}), \quad (3)$$

where $m > 1$ is the clustering fuzziness coefficient.

7. To stop the iteration process, the parameter $\varepsilon > 0$ is set. If the condition $\{|\mu_{X_j}{}^{(k)} - \mu_{X_j}{}^{(k-1)}|\} < \varepsilon$ is not met, then the transition to point 5 is performed.

The clustering method training procedure is implemented using a neural network, which is a five-layer structure without feedback with weight coefficients and activation functions (Fig. 2). The adaptive Takagi-Sugeno-Kang (TSK) model is chosen as a basis [21]. The output signal is determined by the aggregation function for M rules and N variables (in our case, there are $N = 9$ packet characteristics at the network input):

$$y(x) = \frac{\sum_{k=1}^{M}(\mathcal{W}_k * y_k(x))}{\sum_{k=1}^{M}(\mathcal{W}_k)}, \quad (4)$$

where $y_k(x) = z_{k0}\sum_{j=1}^{N}(z_{kj}x_j)$is the i-th polynomial component of the approximation, weights ($\mathcal{W}_i$ represent the degree of fulfillment of the conditions of the rule ($\mathcal{W}_k = \mu_A{}^{(k)}(x_j)$.

The membership function or fuzzification $\mu_A{}^{(k)}$ for the variable x_j is represented by the Gaussian function

$$(\mathcal{W}_k = \mu_A{}^{(k)}(x_j) = \prod_{j=1}^{N}\left[\frac{1}{1+\left(\frac{\left(x_j - c_j{}^{(k)}\right)}{\sigma_j{}^{(k)}}\right)^{2b_j{}^{(k)}}}\right], \quad (5)$$

where k - is the number of membership functions ($k = 1 \ldots M$); j - is the number of variables ($N = 9$); $c_j{}^{(k)}, \sigma_j{}^{(k)}, b_j{}^{(k)}$ - parameters of the Gaussian function that determine its center, width and shape of the k-th membership function of the j-th variable.

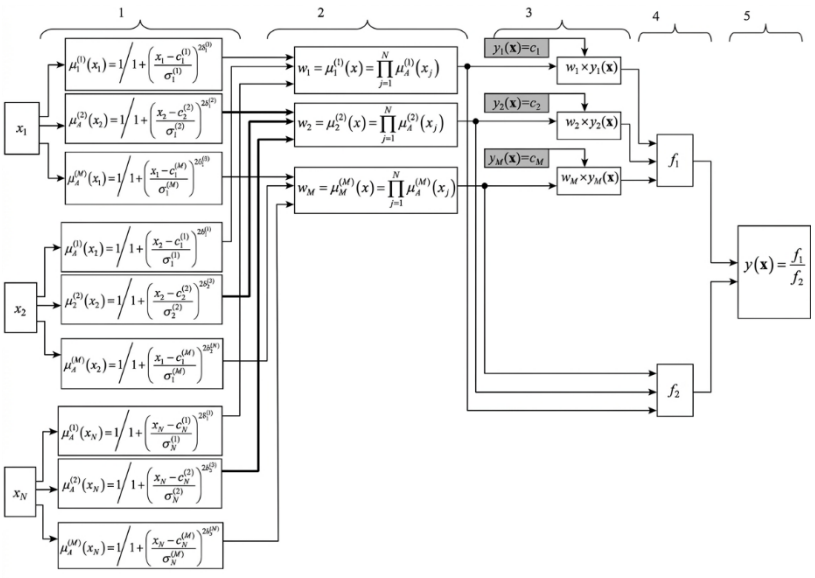

Fig. 2. Architecture of fuzzy neural network for clustering network traffic packets

The rules for deriving output variables $Y = (y_1, y_2, \dots, y_M)$ for a set of variables $X = (x_1, x_2, \dots, x_9)$, taking on a set of values $A_j{}^{(k)}$, represent a matrix of membership function values of size $9 \times M$:

$R1$: if $x_1^{(1)} \in A_1^{(1)}$ and $x_2^{(1)} \in A_2^{(1)}$and, …, and $x_9^{(1)} \in A_9^{(1)}$, then $y_1(x) = z_{10} \sum_{j=1}^{9} (z_{1j} x_j)$, … … … … (6)

R_M: if $x_1^{(M)} \in A_1^{(M)}$ and $x_2^{(M)} \in A_2^{(M)}$and, …, and $x_9^{(M)} \in A_9^{(M)}$, then $y_M(x) = z_{M0} \sum_{j=1}^{9} (z_M x_j)$.

To reduce the computational complexity, we assume that the number of rules

coincides with the number of membership functions, although they may differ.

In the first layer, fuzzification is performed according to formula (5) for each variable x_i. In this case, for each rule R_j, the values of the membership function $\mu_A{}^{(k)}(x_i)$are determined:

$$\mu_A{}^{(k)}(x_j) = \frac{1}{1+\left(\frac{\left(x_j - c_i{}^{(k)}\right)}{\sigma_i{}^{(k)}}\right)^{2b_i{}^{(k)}}}. \quad (7)$$

In the second layer, the coefficients ($\mathcal{W}_k = \mu_A{}^{(k)}(x)$ are determined by aggregating the values of the variables x_i. The ($\mathcal{W}_k$ coefficients are passed to the 3rd layer, where they are multiplied by the $y_i(x)$ values, and to the fourth layer to calculate the sum of the weights.

The third layer calculates the $y_i(x) = z_{k0} \sum_{j=1}^{N}(z_{kj} * x_j)$, values, which are multiplied by the $\mathcal{W}_k$ weight coefficients. The linear parameters z_{k0} and z_{kj} are functions of the consequences of the rules, and z_{k0} is considered as the center of the membership function.

The fourth layer is represented by two neurons (f_1 and f_2) performing result aggregation:

$$f_1 = \sum_{k=1}^{M} \mathcal{W}_k y_k(x) = \sum_{k=1}^{M}\left[\left(\prod_{j=1}^{N} \mu_A{}^{(k)}(x_j)\right)c_k\right], \quad (8)$$

$$f_2 = \sum_{k=1}^{M} \mathcal{W}_k = \sum_{k=1}^{M}\left[\prod_{j=1}^{N} \mu_A{}^{(k)}(x_j)\right]. \quad (9)$$

The fifth normalizing layer is represented by one neuron, where the weights are normalized and the output function is calculated:

$$y(x) = \frac{f_1}{f_2} = \frac{\sum_{k=1}^{M} \mathcal{W}_k y_k(x)}{\sum_{k=1}^{M} \mathcal{W}_k} = \frac{\sum_{k=1}^{M}\left[\left(\prod_{j=1}^{N} \mu_A{}^{(k)}(x_j)\right)c_k\right]}{\sum_{k=1}^{M}\left[\prod_{j=1}^{N} \mu_A{}^{(k)}(x_j)\right]}. \quad (10)$$

The parameter values for the first and third layers are selected at the training stage. The parameters of the first layer $c_j{}^{(k)}, \sigma_j{}^{(k)}, b_j{}^{(k)}$ are considered nonlinear, and the parameters of the third layer z_{kj} are considered linear. Training is performed in two steps. In the first step, the parameters of the membership functions of the third layer are selected by fixing individual parameter values and solving a system of linear equations

$$y(x) = \sum_{k=1}^{M} \mathcal{W}_k \, (z_{k0} + \sum_{j=1}^{N}(z_{kj} x_j)). \quad (11)$$

The output variables are replaced by reference values d_p (P - is the number of training samples). The system of equations can be written in matrix form: $D_p = W * Z$. The solution to the system of equations is found by means of the pseudo-inverse matrix $W^+ : Z = W^+ D_p$. Next, after fixing the values of the linear parameters z_{kj}, the vector Y of the actual output variables is calculated and the error vector $E = Y - D_p$ is determined.

At the second step, the errors are directed in the opposite direction to the first layer, where the parameters of the gradient vector of the target membership function are calculated relative to the parameters $c_j{}^{(k)}, \sigma_j{}^{(k)}, b_j{}^{(k)}$. Then, the parameters of the membership functions are adjusted using the fast gradient descent method:

$$c_j{}^{(k)}(n+1) = c_j{}^{(k)}(n) - \eta \frac{\partial E(n)}{\partial c_j{}^{(k)}}, \quad (12)$$

$$\sigma_j{}^{(k)}(n+1) = \sigma_j{}^{(k)}(n) - \eta \frac{\partial E(n)}{\partial \sigma_j{}^{(k)}}, \quad (13)$$

$$b_j{}^{(k)}(n+1) = b_j{}^{(k)}(n) - \eta \frac{\partial E(n)}{\partial b_j{}^{(k)}}, \quad (14)$$

where n - is the iteration number; η - is the learning rate parameter.

After specifying the nonlinear parameters, the process of adapting the linear and nonlinear parameters is started again. The iterative process is repeated until all process parameters are stabilized.

To estimate the probabilistic membership of a data packet to the clusters of
normal and atypical packets, the membership degree distribution is applied over three intervals:

a) 0–33 (%), b) 34–66 (%), c) 67–100 (%).

The decision to syn the size filtering rules is made when a packet falls into the third interval. Such a packet is considered atypical, the source is identified as a compromised node and its address is added to the exclusion rule of the flow table. If a packet falls into the first interval, it is considered normal. A rule is created for retransmitting packets with similar sets of characteristics and the source address is added to the white list. If a packet falls into the second interval, its degree of proximity to the clusters of normal and atypical packets is questionable. The point corresponding to the feature vector of such a packet becomes the centroid of a new cluster. The number of clusters increases by one, and the clustering problem is solved again. The result of intelligent analysis of data packets is the synthesis of rules for their filtering/retransmission in switches, inclusion of source addresses in white or black lists, adding records to the security event log, which include the identifier of the compromised node, IP address, attack detection time, event recording time, transaction type, etc.

3 Results

A subsystem of intelligent analysis and clustering has been developed, which includes modules for validating network nodes, filtering traffic, logging, a parser and packet analyzer based on fuzzy logic and a neural network, which is part of a system for secure Environmental Monitoring of critical events in the road transport infrastructure (Fig. 3).

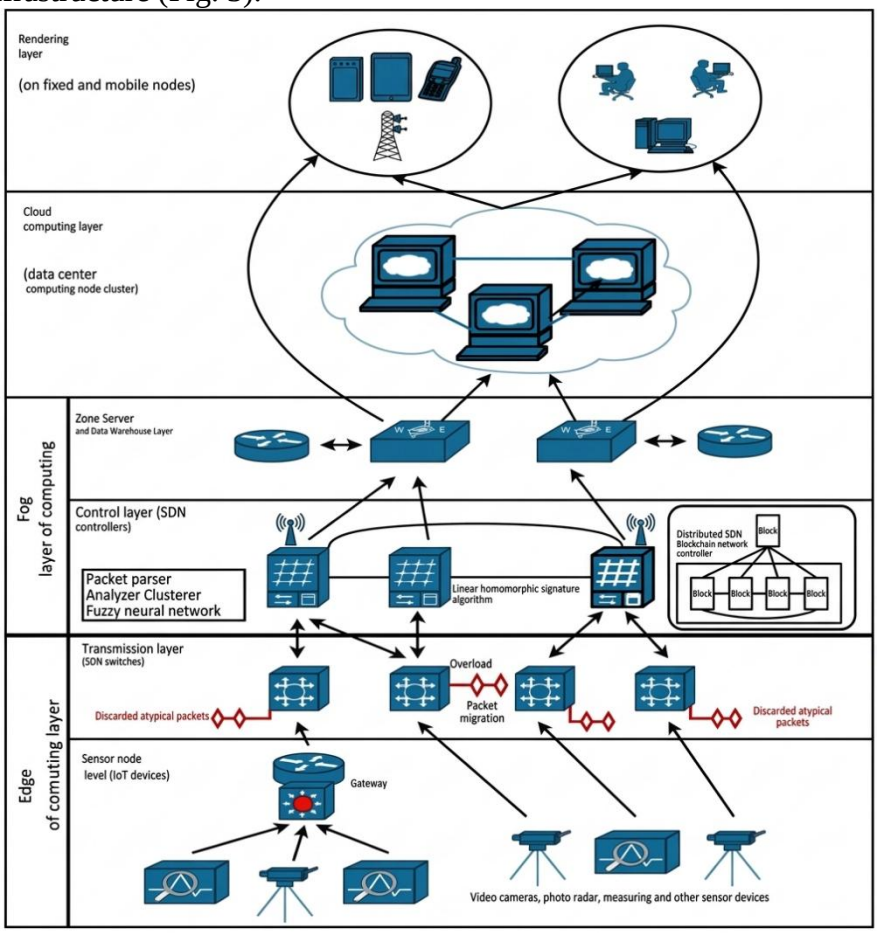

Fig. 3. Architecture of a secure Environmental Monitoring system

The system includes four layers with six data management levels:

1) edge computing layer, including the level of end sensor nodes, routers, sensor network coordinators and the level of data in SDN switches. The layer implements the processes of collecting, processing

and storing telemetry indicators of equipment operation and sensory data, and the function of packet relaying/filtering in SDN switches according to the rules of flow tables;

2) fog computing layer, including the level of zonal servers and the control level in SDN controllers. The layer implements the procedures for synthesizing flow table rules, validating the digital signature of network nodes, analyzing and filtering network traffic, detecting atypical packets based on header analysis and clustering, training a neural network, and logging. The zonal servers perform the functions of data consolidation, aggregation and storage based on the distributed hash table (DHT) technology, since the bulk of the data is photographs or videos from the scene of the incident, which solves the problem of storage scalability;

3) a cloud computing layer on the data center servers, which implements the main functionality of the Environmental Monitoring system for processing and predictive analysis of data, centralized storage of results;

4) a visualization layer is designed to present the Environmental Monitoring results on mobile and stationary user devices.

The modeling of the network architecture of the Environmental Monitoring system was performed in the Network Simulator version 3 (NS3) system [22, 23]. To evaluate the efficiency of packet analysis, several unauthorized nodes were created that generate atypical packets. The result of the modeling is estimates of the architecture performance indicators, such as transmission delay, response time, throughput and packet recognition accuracy. The growth of the transmission delay was estimated depending on the increase in the number of sensor nodes and the number of processed packets (Fig. 4).

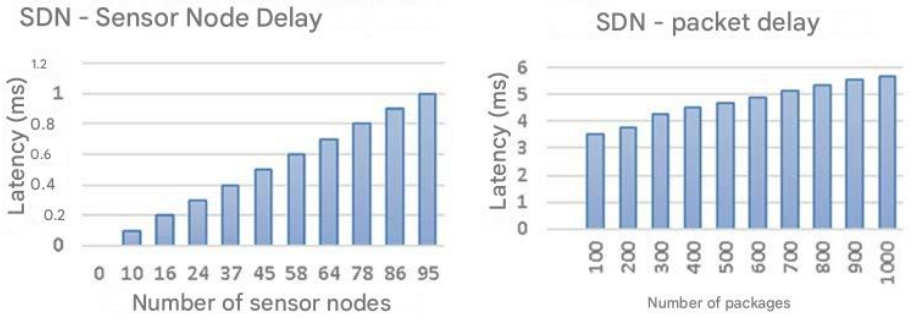

Fig. 4. Delay plots on the number of sensor nodes and the number of packets

The graphs show that the delay increases as the number of devices increases, with changes of several milliseconds after the packet is

transmitted from the end node. However, as the number of nodes increases, the delay becomes significant, as a linear dependence is observed.

The response time is determined based on requests received from end nodes to the gateway, which are transmitted to switches, where they are retransmitted or filtered after packet analysis. Comparison of the response time in the proposed architecture with the response time in the traditional architecture without analysis and clustering showed an increase in delay by an average of 10.5% (Fig. 5).

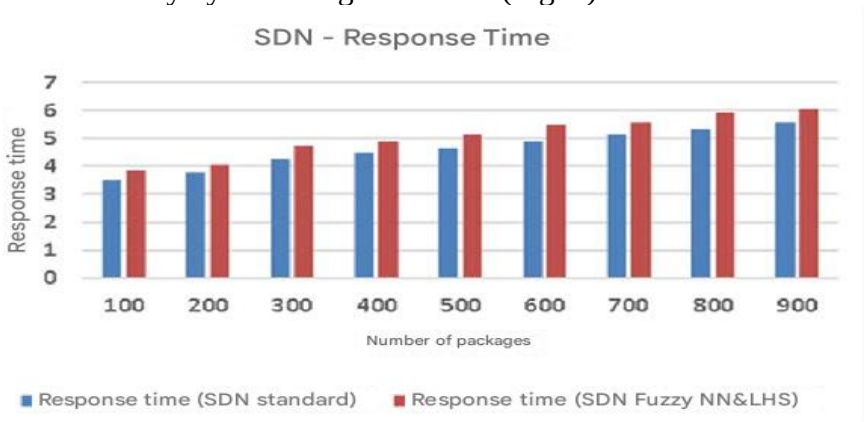

Fig. 5. Comparison of the response time of a network with a traditional SDN architecture and a network with an analysis and clustering subsystem

Fig. 6 shows diagrams of changes in throughput depending on the number of requests to network nodes. The reduction in throughput by an average of 16% in the proposed architecture occurs due to the time spent on analysis, recognition and clustering of packets.

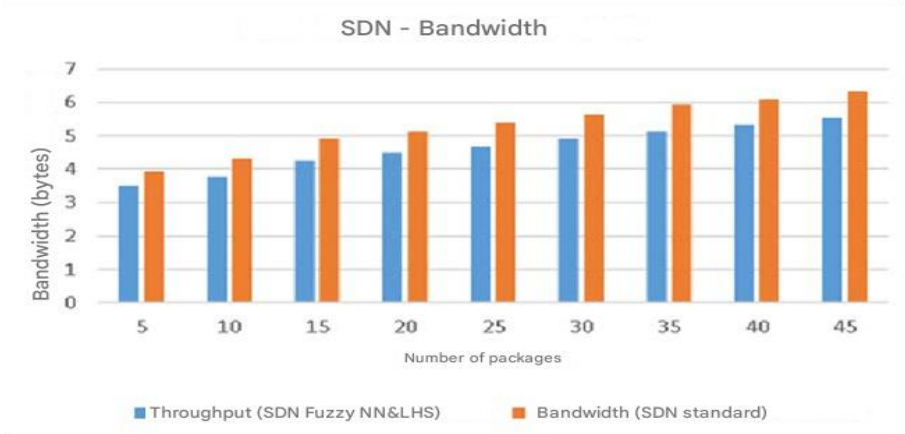

Fig. 6. Comparison of throughput in networks with traditional and proposed architecture

To evaluate the effectiveness of the proposed approach, an assessment of the accuracy of recognizing atypical packets is used. Table 1 shows the accuracy of detecting atypical packets depending on the rate of arrival of switches. The accuracy of detection is quite high, but decreases as the rate of arrival of packets increases, which is due to the time of packet analysis and synthesis of flow table rules.

Table 1. Accuracy of detecting atypical packets.

Packets received for processing (packets/sec)	**Total number of packages**		**Number of packets as a result of clustering**		**Recognition accuracy (%)**
	Normal packages	Atypical packages	Normal packages	Atypical packages	
20	400	75	399	76	**98,68**
40	800	150	797	153	**98,03**
60	1200	225	1195	230	**97,82**
80	1600	300	1590	310	**96,77**
100	**2000**	**375**	**1980**	**395**	**94,94**

4 Conclusion

During the research, a secure architecture of the road traffic Environmental Monitoring system was synthesized using SDN network technology, distributed registry, analysis and clustering of data packets using a fuzzy neural network. The paper presents a number of results:

a) SDN architecture of the system with four device layers and six data management levels;

b) a method for detecting cyberattacks using a fuzzy neural network and a clustering algorithm for evaluating packets by nine characteristics;

c) a set of data packet characteristics for recognizing and probabilistically evaluating malicious packets.

Modeling and evaluation of the effectiveness of the proposed approach showed a fairly high speed of traffic processing in the Environmental Monitoring system.

To reduce the computational load, a three-stage analysis technology is implemented. At the first stage, invalid traffic is filtered by port numbers. At the second stage, the authenticity of data sources and access devices is confirmed and traffic from invalid nodes is filtered. At the third stage, machine learning procedures and intelligent analysis of filtered packets are performed.

References

1. de Campos Souza, P.V., Guimarães, A.J., Rezende, T.S., Silva Araujo, V.J., Araujo, V.S.: Detection of Anomalies in Large-Scale Cyberattacks Using Fuzzy Neural Networks. AI 1(1), 92–116 (2020).
2. Moshkin, V., Kurilo, D., Yarushkina, N.: Integration of Fuzzy Ontologies and Neural Networks in the Detection of Time Series Anomalies. Mathematics 11(5), 1204 (2023).
3. Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M.R., Tarmizi, S., Rodrigues, J.J.P.C.: Anomaly Detection Using Deep Neural Network for IoT Architecture. Applied Sciences 11(15), 7050 (2021).
4. Souza, P.V., et al.: Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. Journal of Parallel and Distributed Computing 73(6), 790–806 (2013).
5. Lee, W., Maidanov, K.: Fog-Assisted Anomaly Detection in IoT-WSN Ecosystems Using Hybrid Deep Learning Models. Journal of Wireless Sensor Networks and IoT 3(1), 1–9 (2025).
6. Ahmad, Z., et al.: Deep neural network-based anomaly detection for IoT network flows. Applied Sciences 11, 7050 (2021).
7. Hussein, S.F., Dallalbashi, Z.E., Mohammed, A.B.: Anomaly Detection in Internet of Medical Things with Artificial Intelligence. Eastern-European Journal of Enterprise Technologies (2023).
8. Atassi, R.: Anomaly Detection in IoT Networks: Machine Learning Approaches for Intrusion Detection. Journal of Intelligent Systems and Internet of Things 13(1), 126–134 (2023).
9. Mardenov, Y., Adamova, A., Zhukabayeva, T., Othman, M.: Enhancing Fault Detection in Wireless Sensor Networks Through Support Vector Machines. Journal of Robotics and Control 4(6) (2024).
10. Xiaofeng, W.: Federated deep learning for anomaly detection in the Internet of Things. Computers & Electrical Engineering (2023).
11. Karthikeyan, M., Manimegalai, D., RajaGopal, K.: Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. Scientific Reports 14, Article 231 (2024).
12. Bashar, A., Hanif, M., Alshammari, N., Alqahtani, A., Alshahrani, M.: A novel hybrid technique using fuzzy logic, neural networks and genetic algorithm for intrusion detection system. Measurement: Sensors 30, 100933 (2023).
13. Abirami, S., Chitra, P.: A novel enhanced neural network for anomaly detection in the IoT environment. Computers & Electrical Engineering 129(B), 110833 (2026).
14. Hamed, B., Ali, A., Mohamed, E.: A fuzzy neural network approach for online fault detection in waste water treatment process. Computers & Electrical Engineering 40(7), 2216–2226 (2014).
15. Caville, E., Lo, W.W., Layeghy, S., Portmann, M.: Anomal-E: A Self-Supervised Network Intrusion Detection System based on Graph Neural Networks. arXiv (2022).
16. Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R.: DÏoT: A Federated Self-learning Anomaly Detection System for IoT. arXiv (2018).
17. Lo, W.W., Layeghy, S., Sarhan, M., Gallagher, M., Portmann, M.: E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. arXiv (2021).
18. Sáez-de-Cámara, X., Flores, J.L., Arellano, C., Urbieta, A., Zurutuza, U.: Clustered Federated Learning Architecture for Network Anomaly Detection in Large Scale Heterogeneous IoT Networks. arXiv (2023).

19. Kasabov, N.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. MIT Press, Cambridge (1996).
20. Kasabov, N.: Evolving Connectionist Systems: The Knowledge Engineering Approach. Springer, London (2007).
21. Kasabov, N.: Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence. Springer, Berlin (2019).
22. Halder, A., Konar, A., Mandal, R., Chakraborty, A., Bhowmik, P.: General and Interval Type-2 Fuzzy Face-Space Approach to Emotion Recognition. IEEE Transactions on Systems, Man, and Cybernetics (2013).
23. Saha, A., Konar, A., Nagar, A.K.: EEG Analysis for Cognitive Failure Detection in Driving Using Type-2 Fuzzy Classifiers. IEEE Transactions on Emerging Topics in Computational Intelligence (2017).