Research of the Neural Network Module for Detecting Anomalies in Network Traffic

Yurii Klots[a], Vira Titova [a], Natalia Petliak [a], Viktor Cheshun [a] and Abdel-Badeeh M. Salem[b]

[a] *Khmelnitsky National University, Institutska str.,11, Khmelnitsky, 29000, Ukraine*
[b] *Ain Shams University, El-Khalyfa El-Mamoun Street Abbasya, Cairo, Egypt*

Abstract

The aim of this article was to study, develop and apply neural network methods to analyze traffic of local area network for detection of the anomalies in network traffic. The theoretical significance of the presented material is coverage of the problem of research and development of methods of computational intelligence, namely artificial neural networks to detect intrusions in networks. In particular, a method of detecting anomalies was created on the base of a Kohonen self-organizing map, the introduction of which into the structure of the existing attack detection system allows to increase the accuracy of detecting anomalies in network traffic by 35-40% compared to other existing methods. The practical significance of the results is the possibility of applying the developed method to detect intrusions and anomalies in modern corporate networks and the possibility of building a real-time detection system based on it, as well as solving of other tasks of data mining such as classification or clustering of suspect events in network traffic.

Keywords

Network security, detection methods of network intrusion, data protection, intrusion detection systems, neural networks.

1. Introduction

The rapid development of computer networks and information technologies causes a number of problems related to the security of network resources, which require new approaches.

Currently, the issue of building intrusion detection systems is an important direction in the field of information technologies.

There are many researches on the detection and classification of intrusions using a variety of methods, which include traditional approaches based on compliance with signature patterns and adaptive models using data mining techniques. Most of these works have been done long ago, and some of them have a limited aspect in the covering form of only a specific subject area, namely, the detection of overuses or anomalies.

This work aims to develop a neural network module for detecting anomalies in network traffic, its implementation and verification of its operation on test and real data.

The article has the following structure. The second section provides a comparative analysis of intrusion detection methods. The third section substantiates the choice of an artificial neural network as a method of computational intelligence to solve the problems of detecting anomalies. The fourth section is devoted to the description of the structure and methods of functioning the popular intrusion detection system. The fifth section describes the neural network module for detecting anomalies in network traffic, which is an implementation of the neural network method.

IntelITSIS'2022: 3rd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 23–25, 2022, Khmelnytskyi, Ukraine
EMAIL: klots@khmnu.edu.ua (Y. Klots); titovav@khmnu.edu.ua (V. Titova); npetlyak@khmnu.edu.ua (N. Petliak); cheshunvictor@gmail.com (V. Cheshun); abmsalem@yahoo.com (A.-B. M. Salem)
ORCID: 0000-0002-3914-0989 (Y. Klots); 0000-0001-8668-4834 (V. Titova); 0000-0001-5971-4428 (N. Petliak); 0000-0002-3935-2068 (V. Cheshun); 0000-0003-0268-6539 (A.-B. M. Salem)

Sections number six and seven contain a description and results of the module operation on test and real data sets.

2. Comparative analysis of methods of detecting intrusions

The common classification of intrusion detection systems by intrusion detection methods includes anomaly detection systems and overuse detection systems [1, 2].

Figure 1 shows the scheme of network anomaly detection [3–5] based on network traffic indicators.

The general algorithm for detecting network anomalies can be described as follows. Network traffic, presented as a set of network packets generally fragmented at the IP level, is the data for analysis. The collected data serve as a source for the formation of the necessary information for further analysis.

Thus, the obtained data can be aggregated for a certain interval and normalized in order to set the feature attributes of the general form, which will be required when constructing the current activity profile.

The created set of features is compared with a set of characteristics of the normal operation of the object (user or system) by pattern of normal behavior. If there is a significant discrepancy between the compared parameters, the network anomaly is recorded.

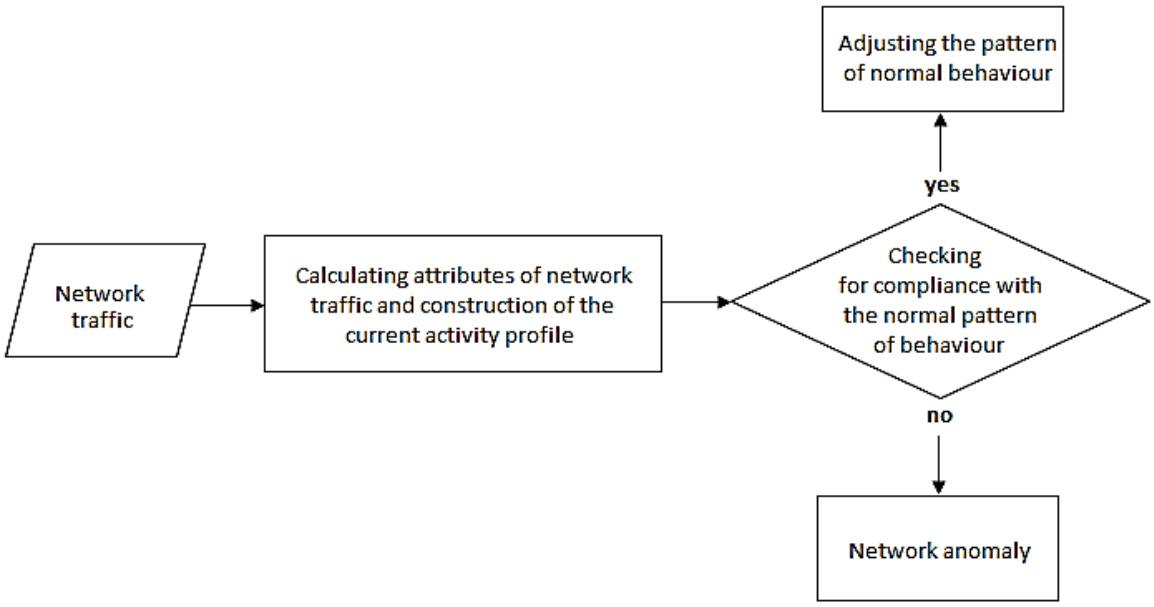

Figure 1: Detection scheme of network anomalies

The above-described algorithm may contain several variants for the realization of checking subsystem for the compliance of the pattern of normal behavior.

The procedure of comparison with the threshold value is the simplest of them, when the accumulated results describing the current network activity are compared with an expertly set numerical plank. In this approach, the case of exceeding the values of the parameters of the specified limit is a sign of a network anomaly.

It is worth noting that construction of a pattern of normal behavior is a time-consuming task and not always feasible. In practice it turns out that not every abnormal behavior is an intrusion. For example, a network administrator can use adjustment of utilities such as ping, traceroute, mtr to diagnose a network environment.

Actions of this kind do not pursue any illegal intent, but anomaly detection systems recognize this activity as anomalous network activity.

Detection of overuse allows you to identify unauthorized actions, if they are accurately reported in the form of intrusion patterns. Intrusion pattern is a set of actions, which explicitly describe a specific attack. Using them for the identified object we can receive a clear answer about affiliation of this pattern to this attack. As in the detection scheme of network anomalies, a network traffic is the primary data for analysis while detecting overuse.

Selected attributes and fields of network packets are transmitted to the module, which searches for and verifies compliance with the rules of input data and indicates a threat in the event of a positive interaction of one of the rules.

Figure 2 shows the scheme for detecting overuse in network traffic.

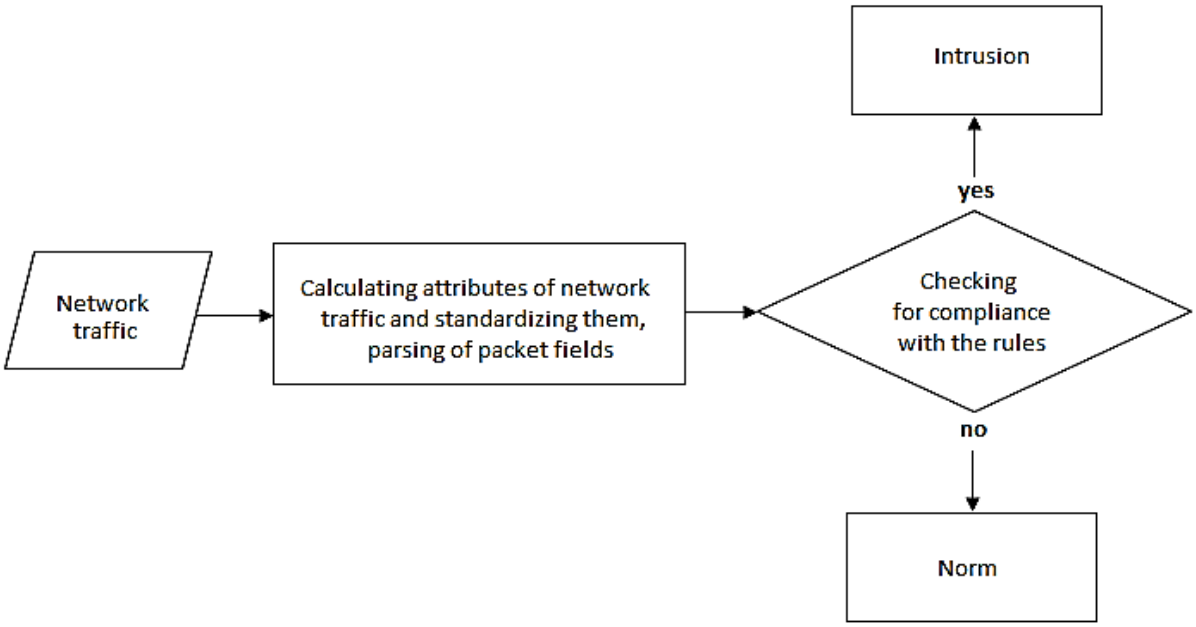

Figure 2: Detection scheme of network overuse

The key problem in creating any system of overuse detection is the issue of effective design of the rule setting mechanism. It is clear that the creation of an exhaustive base of rules for detecting various attacks is impossible due to several factors. One of these factors is that the description of different variations of attacking actions has a negative effect on system performance. And since even insignificant changes in the attack lead to the impossibility of its detection by methods based on overuse, the rules should be universal and cover as many known modifications of network intrusions.

Summing up, we should note that methods of overuse detecting are an effective tool for detecting known types of intrusions, but their applicability to new attacks, as well as to modifications of known attacks is ineffective.

“A Survey of Network Attack Detection Research” [6] of Abas Aboras, Mohammed Kamal Hadi is a classic work in the field of overuse detecting.

On the base of the analysis of modern methods of intrusion detecting, we can conclude that all of them are not perfect and do not provide high efficiency in counteracting intrusions in computer networks.

Anomaly detection methods are more effective in counteracting modified and previously unknown intrusions, but this group of methods has the following features:

- there are a large number of possible solutions, which requires significant time to solve the problem by a complete search of all available variants;
- input data can be changed in the process of solving the problem, and while changing at least one value, you need to sort all the available variants from the beginning;
- it is difficult to represent input data in the form of numerical data, and therefore the solution of the problem cannot be reduced to numerical calculations.

Thus, the problem of detection of anomalies is a difficult task.

The use of traditional mathematical methods for its solution is impractical, so based on the features of the above-mentioned problem it is necessary to develop intelligent methods to solve this important problem.

3. Substantiation of the intelligent method for solving the problem of anomaly detection

To choose the method, an analysis of a large list of works was carried out, including [7–25], which allowed to clarify the proposed taxonomies and schemes of known methods of detecting network intrusions.

Among all the methods, special attention, in our opinion, deserves the methods of computational intelligence, namely artificial neural networks.

An artificial neural network is a set of processing elements of neurons interconnected by synapses and converting a set of input values into a set of desired output values.

Neural networks are used in a wide range of applications: image recognition, management theory, cryptography, data compression.

Neural networks have the ability to learn from patterns and generalize from noisy and incomplete data. The coefficients associated with synaptic weights are adjusted in the learning process.

A brief overview of several models of neural networks is presented in this article, namely, multilayer feedforward networks, radial-basis networks, recurrent networks and self-organizing maps.

There are several methods for learning neural networks.

12 algorithms of their learning are presented in [7]. The method of backward propagated error is one of the best known and most widely used algorithms for learning multilayer feedforward neural networks. This algorithm is a gradient descent with minimization of mean squared error at each iteration of its execution.

A multilayer neural network with two hidden layers and an outgoing layer consisting of three neurons is used to detect attacks in [8, 9]. The classifier was learned to recognize two types of attacks and a normal connection. An algorithm of backward propagated error is used in both works to learn the neural network.

Another work that uses the same database is [10]. It shows the architecture of a multilevel neural network, in which each of the three levels is a separate multilayer perceptron, the distributing layer of which consists of 30 neurons. The classification of the connection is refined at each level. Whether a connection is an attack or a normal connection is determined at the first level. The second and third levels are responsible for classification according to the class and subclass of the attack. The ability to obtain the required degree of details in the classification of the connection is a feature of this approach.

A three-layer neural network was used in [11] as a binary classifier of network connections. The training set that is a network traffic, received by the network scanner, numbered about 10,000 connections, 3,000 of which were simulated attacks. Although the training took 26.13 hours, the results of the experiments showed a high level of recognition correctness.

The works [12, 13] are devoted to the detection of anomalies using neural networks based on data taken from the system audit log and log files of individual applications. Sets of the most common commands and frequency of their use were applied in [14] to set the profile of each user. [15] also used system information, including the amount of system resources, system time, etc.

Radial-basis neural networks are a class of neural networks based on the calculated distance from the entering vector to the centers of neurons of the hidden layer. Radial-basis neural networks require fewer computing resources and time for learning because they have a simpler structure compared to multilayer neural networks, that why they are ideal for tasks with a large sample size. The work [16] gives a review of works with application of radial-basis neural networks to problems of detection of attacks.

Self-organizing maps, or Kohonen maps, are single-layer feedforward neural networks whose outgoing layer is an n-dimensional lattice (usually $n = 2$ or $n = 3$). After training such networks group entering vectors with similar characteristics into separate clusters.

[17, 18] propose to use self-organizing maps to detect anomalies. For this purpose, data was collected describing the legitimate behaviour of users and including the characteristics of system calls within the computer network. Self-organizing maps are used in [19-20] to process and cluster data about network traffic.

In [21-23] new approaches for DNS tunneling botnet detection, which considers all the features and architectural characteristics of botnets are presented.

The described methods of working with self-organized neural networks show a great increase in the accuracy of detecting anomalies in the analysis of computer data networks, they also help in compiling the topology of the computer network and finding errors in existing networks.

The main difference of the method of self-organized maps is that it eliminates the need to learn the neural network on a pre-prepared data set [26].

This method is suitable for solving the tasks in the work. So, to implement an intelligent method of detecting anomalies, we decided to take a self-organized Kohonen map.

The implementation of the maps is a neural network module connected in real time to the Snort intrusion detection system.

4. The structure and methods of operation of the intrusion detection system

Snort is a network intrusion detection (IDS) and intrusion prevention (IPS) system with open-source code that can perform packet logging and real-time traffic analysis on IP networks, combining signature matching capabilities, protocol inspection tools, and anomaly detection mechanisms. Snort was created by Martin Roesch in 1998 and quickly gained popularity as a free intrusion detection system that allows you independently and effortlessly to write rules for detecting attacks. In essence, the Snort signature description language has become de facto standard for many intrusion detection systems that have used it in their functioning.

According to the monitoring method, the Snort intrusion detection system can be host-based and network-based, depending on the configuration parameters. It usually protects a certain segment of the local network from external attacks from the Internet. The Snort system provides protocol analysis, content search, and it is widely used for active blocking or passive detecting a range of attacks and soundings.

Snort is based on five modes:

- Sniffer packets: this mode is responsible for capturing data transmitted over network to the decoder. This is done using the DAQ (Data AcQuisition) library. This sniffer can be inline, passive or read network data from a pre-prepared file;
- Packet decoder: this mode parses headers of captured packets, parses packets themselves, searches for anomalies and deviations from RFC, analyzes TCP flags, excludes certain protocols from further analysis and so on. This decoder focuses on TCP/IP stack;
- Preprocessors: if a decoder parses the traffic at the 2nd and 3rd levels of the reference model, the preprocessors are designed for more detailed analysis and normalization of protocols at the 3rd, 4th and 7th levels. Among the most popular preprocessors there are frag3 (work with fragmented traffic), stream5 (reconstruction of TCP streams), http_inspect (normalization of HTTP traffic), DCE / RPC2, sfPortscan (used to detect port scans) and various decoders for Telnet, FTP, SMTP, SIP, SSL, SSH, IMAP, etc. Some developers write their own preprocessors (for example, for industrial protocols) and add them to their own intrusion detection systems (IDS), built with Snort;
- Intrusion detection mode: this mode consists of two parts. The rule constructor collects many different decision rules (attack signatures) into a single set, optimized for further use by the inspection subsystem of captured and processed traffic in search of certain violations;
- Output mode: Snort can generate (write or display) a corresponding message upon detection of an attack in various formats: file, syslog, ASCII, PCAP, Unified2 (binary format for accelerated and easy processing).

It should be noted that despite the planned updates of the Snort system, the functionality of the program does not change and is based on two main principles. Elementary analysis – comparison of parameters of incoming network packets, arranged on the rules of Snort system. Signature analysis –

parsing of packets coming to the listening socket of the system and finding a certain sequence of bytes.

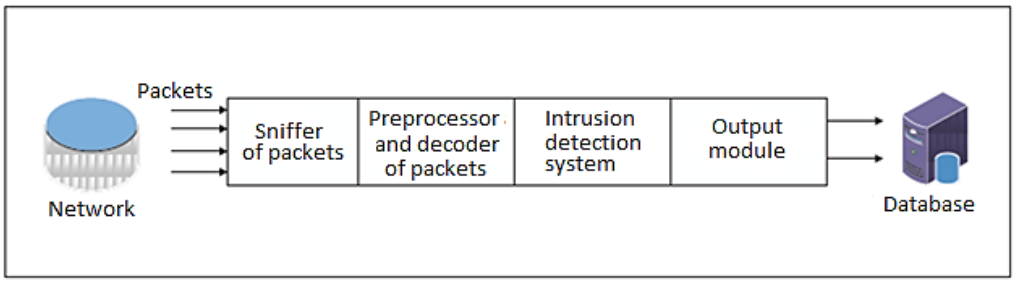

Figure 3: Snort system structure

Elementary analysis is based on the parsing and verification of the header of incoming network packets, analysis of data included in this network packet. This method is less laborious in terms of network resources of the Snort system.

According to the data obtained by parsing the header of the network packet (IP address of sender/ recipient, port of sender/recipient) and comparing them with Snort's own base of rules, the system is able to detect an attempt to attack the network.

Signature analysis means the analysis of data contained in the network packet. This method is a more effective way to identify hazards, but at the same time more laborious in terms of computing resources.

However, none of the analyses is highly effective in detecting anomalies. That`s why, it seems promising to create new methods of detecting attacks that eliminate the shortcomings of these methods. It is also necessary to develop a system that combines the advantages of new techniques that will allow more effective detection of attacks in computer networks, ensuring high accuracy of attack detection and low computational complexity of the algorithms used to analyze network traffic data.

5. Description of the neural network module

The main goal is to change the existing Snort structure, shown in Fig. 3, and integrate an additional module into it, which will lead to the new structure, shown in Fig. 4. The proposed adaptive module works in parallel with the Snort rule set. Substantiation of integration of adaptive module in parallel with the rule set is that Snort rule set detects only known destructive traffic.

The module may detect an unknown variant of destructive traffic or reduce the number of false operations. This will reduce the number of false-positive alarms, and improve detection accuracy.

The preprocessor will transmit network traffic into the neural network module and the Snort rule set, and they will both work in parallel for more accurate detection of destructive traffic.

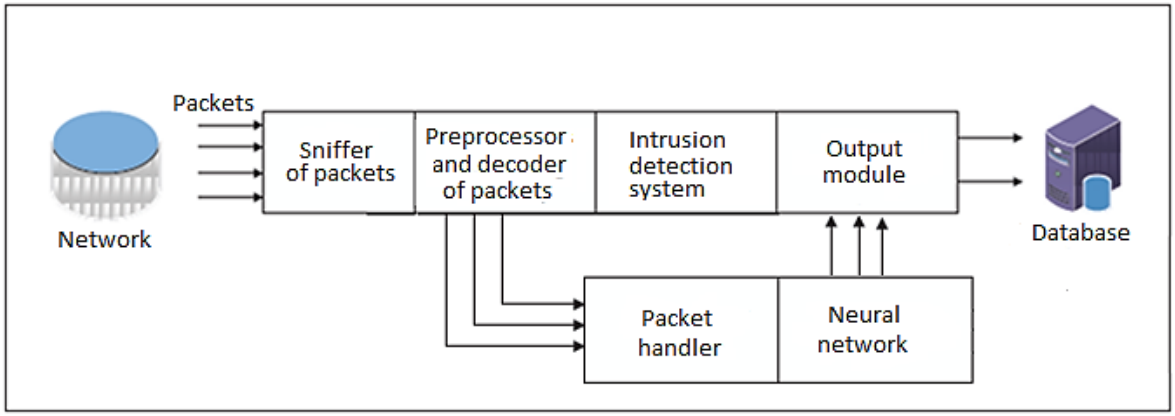

Figure 4: Snort system structure with developed neural network module

The most important process of working with the models of machine learning is obtaining reliable data. Obtaining such data is itself a serious problem, as the availability of data sets is very low. On the one hand, many datasets are internal and cannot be used openly due to privacy issues, and on the other hand, datasets are highly anonymous and do not reflect current network trends or do not have certain statistical parameters, so the perfect dataset does not yet exist. Thus, researchers must use data sets that are often nonoptimal. As behavior and modes of the network change, as well as development of intrusion systems, there is a need to move from static and disposable datasets to more dynamic datasets that not only reflect traffic structure but are changeable, extensible, and reproducible.

Intrusions are divided into three main groups for classification:

- DOS: denial of service;
- R2L: unauthorized access from a remote machine;
- U2R: : unauthorized access to local root privileges.

Input parameters must be prepared before they can be used in the machine learning algorithm. Some items can be easily found, others need to be found through experimentation and testing. Using all the features of the data set does not necessarily guarantee the best performance. This can increase computing costs as well as the frequency of errors in the system. This is due to the fact that some functions are redundant or useless for distinction of different classes of attacks.

The main advantage of this training method is the introduction of attributes proposed by the expert, which help to understand the behavior of different types of attacks, including the basic characteristics of attack detection. Here is the main list of input parameters for learning (Table 1).

Table 1

List of input network parameters

Name	Description
duration	Duration in seconds
protocol_type	Type of protocol (TCP, UDP etc.)
service	Type of service (telnet, http etc.)
flag	Connection flag: error or norm
scr_bytes	Number of bytes from source to receiver
dst_bytes	Number of bytes from receiver to source
land	1, if connection is on the same host or port
wrong_fragments	Number of incorrect fragments
urgent	Number of urgent packages
count	Number of connections to the host in the last two seconds
srv_count	Number of connections to the service in the last two seconds
serror_rate	Percentage of connections to SYN with errors
diff_srv_rate	Percentage of connections to various services
srv_diff_hast_rate	Percentage of connections to different hosts
dst_host_srv_count	Number of connections to the local host through remote access and the same service

The basic principle of learning the neural network module: the network is learned for some time in the normal operation of the data transmission network. Some clusters are created meanwhile which fully characterize the network's normal operation when properly configured. The training utilizes competitive learning. When a training example is fed to the network, its Euclidean distance to all weight vectors is computed. The neuron whose weight vector is most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM grid are adjusted towards the input vector. The magnitude of the change decreases with time and with the grid-distance from the BMU. The update formula for a neuron v with weight vector $W_v(s)$ is:

$$W_v(s+1) = W_v(s) + \theta(u, v, s) \cdot \alpha(s) \cdot \big(D(t) - W_v(s)\big), \quad (1)$$

where

$D(t)$ – is a target input data vector;

W_v – is the current weight vector of node;

$\theta(u, v, s) -$ is a restraint due to distance from BMU, usually called the neighborhood function;
$\alpha(s) -$ is a learning restraint due to iteration progress.
There are variants of detection of anomalies:

- the network is learned on test data sets, contains a number of clusters. In the case of data containing an anomaly, the structure and number of clusters will begin to change;
- transfer of new data for testing the trained network and its comparison, whether they suit for one of the formed clusters.

In these cases, the anomalies can be identified by changing the number of existing clusters in the training model, the dynamics of change in the creation and coordination of neurons and the connections between them. It is also possible to detect anomalies in the deviation of the value of new measurements from the average value of existing neurons on which the network conducted training.

It is quite problematic to perform dynamic counting methods in the methods of network activities. Neural network can change the number of neurons and the connections between them at each iteration. In addition, some algorithms collect and remove a fixed number of neurons at each iteration.

Also, change of the number of clusters is not a reliable indicator of threat detection. The change of the deviations from the values of the existing neurons, on which the network has allowed training, is one reliable option. An adaptive adjustment, which is similar to semi-automatic learning with a teacher, can be used in the given method as can be seen from the block-scheme. The algorithm from the block-scheme uses not one deviation but the average deviation between the best neurons for more complete detection. Also, the arithmetical mean of all deviations included in this cluster of neurons is calculated for each cluster. The nearest neuron from the neurons formed during the training is found for the new data sets submitted for the neural network module test and the distance between them is calculated. Next, this distance is compared with the standard deviation for the cluster, which includes this neuron. If the distance between the neuron and the data submitted for verification is greater than the average deviation of the cluster, this data set is considered abnormal.

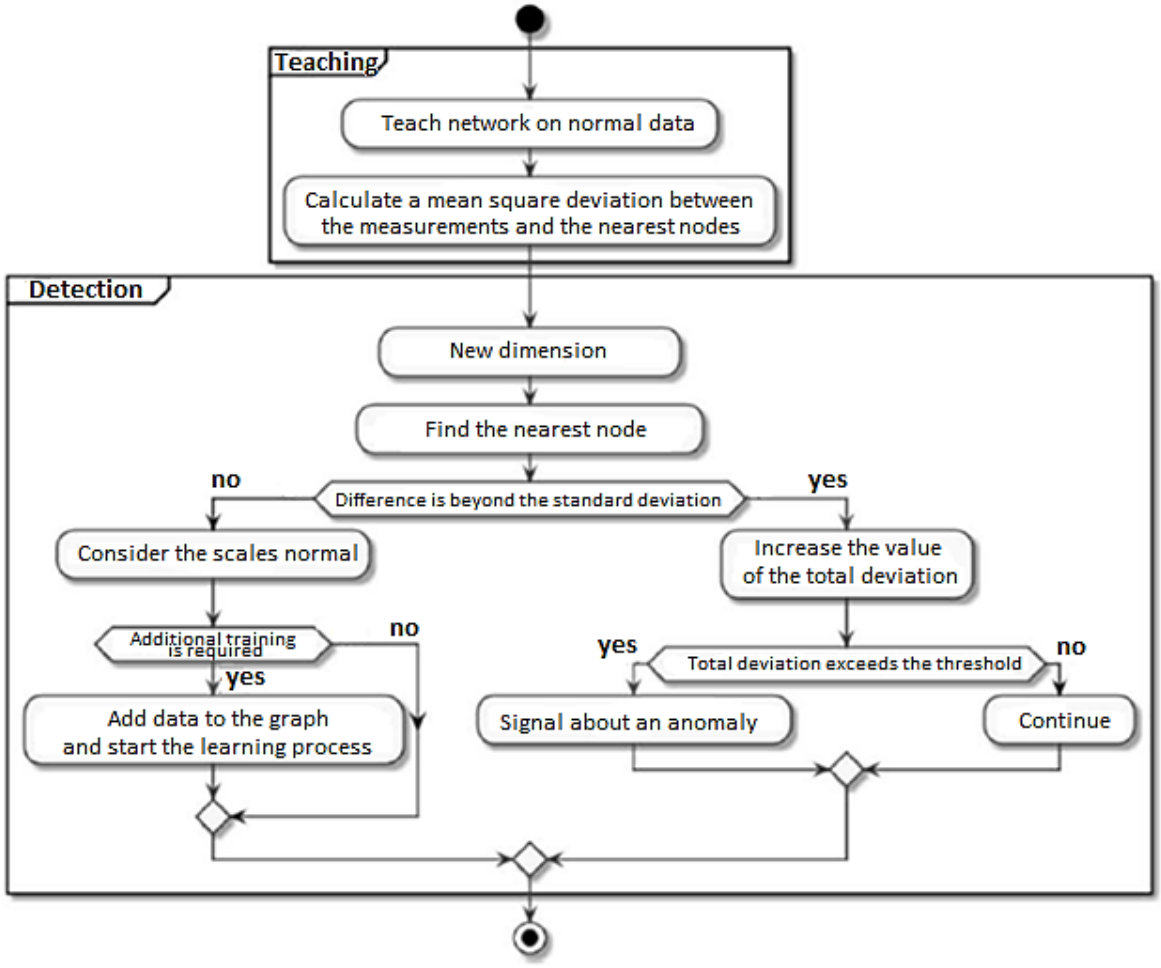

Figure 5: Block-scheme of the clustering method based on the self-organized Kohonen map.

6. Description and results of work on test data

To check the capacity of the developed neural network module, it was tested on the data set CSE-CIC-IDS2018 of the Canadian Institute of Cybersecurity. The module conducts training on a training sample from this set, which contains 1011 records. Then that set is re-submitted to the neural network module to check the correct training.

The trained module was applied to complete data from the same set which was used to construct the training sample. The results are shown in table 2.

After all the tests and making sure that the training was correct, the module was used to analyze real samples. Python with a neural network library called Scikit-Learn was used as the implementing language for the neural network module.

There are a large number of machine learning algorithms based on the learning algorithms with and without teacher.

This library does not set input data, so the NumPy library downloads the input data. The Scikit-Learn library specializes in neural network analysis algorithms for clustering, dimensional reduction, and anomaly detection.

The NetworkX library is used for visualization, if necessary. This library is designed to work with network structures and graphs.

Data for the neural network module is processed from the Snort preprocessor module using the standard csv module.

The data is transmitted as a non-normalized NumPy array. With the help of the Scikit-Learn library the array data is converted to a normalized form. Further work is carried out only with normalized data.

An undirected graph for visualization is built on the base of the read data. Also empty graph will be built, in which neurons and the connection between them will be added. Then the clustering method is applied repeatedly, where one of the values is transmitted, which is a set of coordinates of a point in multidimensional space.

Obtaining data from this point, the clustering method selects the closest neurons of the internal graph and, if they are not satisfactory, creates new neurons and connections.

Table 2

The result of the neural network module on the test data set

Variable	Value	Description
l_time	1869.00	Training time in seconds
te_l_time	39.5	Test time in seconds on the set that formed a training sample
te_t_time	125.61	Validation time in seconds on the complete data set of test sample
g_l_perc	69.5	Percentage of detected anomalies for the set which formed the training sample
g_t_perc	78.4	Percentage of detected anomalies for the complete data set of test sample
f_l_perc	0.00	Percentage of false operations for the set which formed the training sample
f_t_perc	6.9	Percentage of false operations for the complete data set of test sample

The clustering method periodically saves the image with visualization, which is used twice: first to display the data, then to display the neural network over the data. After training, all saved images are united in gif.

Specified parameters for test data:

- number of learning steps for SOM – 7000;
- number of normal records in the training sample – 516;
- number of anomalous records in the training sample – 495:

- number of normal records in the test sample – 2152;
- number of anomalous records in the test sample – 9698;
- full size of the test sample – 11 850.

7. Description and results of work on real data

To check the capacity of the neural network module, a traffic dump was recorded using software of the traffic analysis for Snort computer networks. The recording was based on real data for 60 seconds from the Internet and made a test sample of 936,840 lines.

This test sample was divided into 2 parts. The first 200,000 lines of the test sample were selected for training. The remaining 736,840 records were selected for testing the neural network module.

With the help of the visualization module graphic representations of self-organized map were constructed after 46,000 steps and after 109,000 learning steps, which are presented in Figure 6.

The self-organizing maps were built from 200,000 lines of network traffic that were routed to the neural network module through the Snort system preprocessor module.

The basic structure of the self-organized map according to the graphic display was formed on 46,000 steps.

And the main differences after 46,000 and after 109,000 iterations were not observed.

The disconnected location from the main plane in this case is justified by the presence of packets in the test set of network traffic that are not typical for the total mass.

Test data is submitted through the Snort system preprocessor module. Then, in the normal operation mode, the conclusions of the neural network are compared with the values obtained as a result of the operation of the intrusion detection module of Snort system.

The number of detected anomalies of the neural network module and the number of detected anomalies by the intrusion detection module of Snort system are shown in Table 3 after building a self-organized map.

Checking was performed on a test set consisting of 736,840 lines.

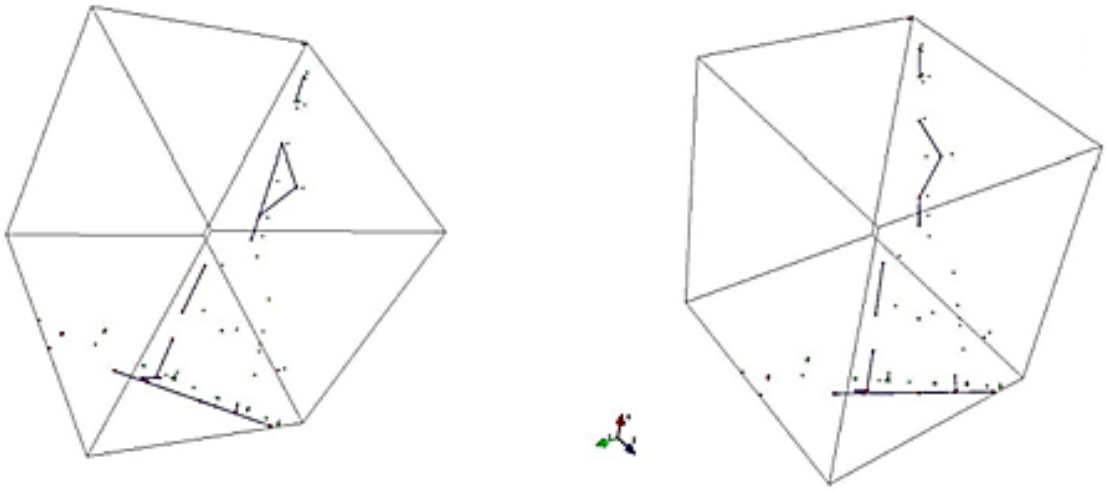

Figure 6: Self-organized map after 46,000 and after 109,000 learning steps

Table 3
The results of the neural network module on real data

Description	IDS Snort	Neural network module
Number of processed records	736 840	736 840
The number of detected anomalies	14	19

8. Conclusions

This article analyzes the intrusion detection systems. The general processing mechanism of network events is considered on the example of the Snort detection system. The purpose of the analysis was to clarify the main problems of these systems and the possibilities of solving them through the use of neural network technologies. An adaptive module has been proposed that works in parallel with the Snort system rule set. The integration of such a module in parallel with the set of rules of the Snort system is due to the fact that the set of rules of the Snort system detects only known destructive traffic from its data base. The neural network module has the ability to detect unknown or altered variants of destructive traffic, which, in its turn, improves detection accuracy. After learning modern systems of neural network technologies, it was decided to detect anomalies using clustering methods of "map-network activity".

Within the neural network technologies, the clustering problem is solved with the use of self-organizing Kohonen maps. The algorithm of clustering of the self-organized Kohonen map was described and the working method of operation of the neural network module according to this algorithm was offered.

Experiments with the developed model of the neural network module showed the ability to detect attempts to attack the network. To check the capacity of the module on real traffic the comparison of the results of the neural network module with the intrusion detection module of Snort system was performed.

The scientific novelty of the work:

- A clustering method based on self-organized neural networks has been developed, which increases the accuracy of detecting anomalies in network traffic by 35–40%.
- The structure of the intrusion detection system was improved on the base of the developed method, which allows to increase accuracy compared to other systems, to recognize previously unknown attacks, and to update the attack database for additional training and configuration of detection modules.

9. References

[1] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7.

[2] Sharma et al., Network Attacks and Intrusion Detection System. 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 280-283, 2019.

[3] Veeramreddy Jyothsna and Koneti Munivara Prasad J. Cohen (Ed.). Anomaly-Based Intrusion Detection System. Computer and Network Security, 2018. DOI: 10.5772/intechopen.82287

[4] Shikha Agrawal, Jitendra Agrawal. Survey on Anomaly Detection using Data Mining Techniques. Procedia Computer Science. Volume 60, 2015, Pages 708-713. https://doi.org/10.1016/j.procs.2015.08.220.

[5] Mohiuddin Ahmed, Abdun Naser Mahmood, Jiankun Hu. A survey of network anomaly detection techniques. Journal of Network and Computer Applications. Volume 60, January 2016, Pages 19-31. https://doi.org/10.1016/j.jnca.2015.11.016.

[6] Abas Aboras, Mohammed Kamal Hadi. A Survey of Network Attack Detection Research. International Journal of engineering Research & Technology (IJERT), Volume 10, Issue 08, 2021.

[7] Mohammed Maithem, Dr.Ghadaa A. Al-sultany. Network intrusion detection system using deep neural networks. ICMAICT 2020. doi:10.1088/1742-6596/1804/1/012138

[8] Prajoy Podder, Subrato Bharati, M. Rubaiyat Hossain Mondal, Pinto Kumar Paul, Utku Kose. Artificial Neural Network for Cybersecurity: A Comprehensive Review. Journal of Information Assurance and Security, Volume: 16, Issue: 1, 2021, pp.010-023.

[9] Sarker, I.H. Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. SN COMPUT. SCI. 2, 154 (2021). https://doi.org/10.1007/s42979-021-00535-6.

[10] Yousef Abuadlla, Omran Ben Taher, Hesham Elzentani. Flow Based Intrusion Detection System Using Multistage Neural Network. 2018.
[11] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis, Robert Atkinson. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. 2017. https://doi.org/10.48550/arXiv.1701.02145.
[12] Sergio Hidalgo-Espinoza, Kevin Chamorro-Cupuerán and Oscar Chang-Tortolero. Intrusion detection in computer systems by using artificial neural networks with Deep Learning approaches. 10th International Conference on Advances in Computing and Information Technology (ACITY 2020), November 28~29, 2020, London, United Kingdom Volume Editors : David C. Wyld, Dhinaharan Nagamalai (Eds). DOI: 10.5121/csit.2020.101501.
[13] Halenar, Igor & Juhásová, Bohuslava & Juhás, Martin & Martin, Nesticky. (2014). Application of Neural Networks in Computer Security. Procedia Engineering. 69. 1209-1215. 10.1016/j.proeng.2014.03.111.
[14] Alia AbuGhazleh, Muder Almiani, Basel Magableh, and Abdul Razaque. Intelligent intrusion detection using radial basis function neural network. 2019 Sixth International Conference on Software Defined Systems (SDS). PP. 200-208.
[15] Liu, Xuejun et al. Improved RBF Network Intrusion Detection Model Based on Edge Computing with Multi-algorithm Fusion. International Journal of Computers Communications & Control, [S.l.], v. 16, n. 4, july 2021. ISSN 1841-9844.
[16] Sheth H., Shah B., Yagnik S. A survey on RBF Neural Network for Intrusion Detection System. Int. Journal of Engineering Research and Applications. 2014. vol. 4. Issue 12. pp. 17–22.
[17] Y. Dong, R. Wang and J. He, Real-Time Network Intrusion Detection System Based on Deep Learning, 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), 2019, pp. 1-4, doi: 10.1109/ICSESS47205.2019.9040718.
[18] Farzan, Ali. Intrusion Detection System Using Self Organizing Map Algorithms. 2014. 3. 585.
[19] Vita Santa Barletta, Danilo Caivano , Antonella Nannavecchia and Michele Scalera. A Kohonen SOM Architecture for Intrusion Detection on In-Vehicle Communication Networks. Appl. Sci. 2020, 10(15), 5062; https://doi.org/10.3390/app10155062.
[20] Subarna Shakya, Bisho Raj Kaphle. Intrusion Detection System Using Back Propagation Algorithm and Compare its Performance with Self Organizing Map. Journal of Advanced College of Engineering and Management, Vol. 1, 2015. DOI:10.3126/jacem.v1i0.14930.
[21] A. Nicheporuk, Y. Klots, O. Yashyna, S. Mostovyi, Y. Nicheporuk. Prediction of entering processes into the deadlock state. Indonesian Journal of Electrical Engineering and Computer Science, 2019, 14(3), pp. 1484–1492.
[22] O. Savenko, S. Lysenko, A. Kryshchuk, Y. Klots. Botnet detection technique for corporate area network. Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS 2013, 2013, 1, pp. 363–368, 6662707.
[23] Yu.P. Klyots, Yu.G. Savchenko, V.N. Cheshun. Trouble-shooting without dictionary: A new approach to diagnosis of digital devices. Upravlyayushchie Sistemy i Mashiny, 2001, (3), pp. 36–42.
[24] Sergii Lysenko, Kira Bobrovnikova, Oleg Savenko, Roman Shchuka. A Cyberattacks Detection Technique Based on Evolutionary Algorithms. 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2020. Vol.1. pp. 127-132.
[25] B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky. Detection DNS Tunneling Botnets. Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021, Cracow, Poland, September 22-25, 2021.
[26] Shangytbayeva, G.A., Akhmetov, B.S., Karpinski, M.P., Beysembekova, R.N., Ospanov, E.A. Research distributed attacks in computer networks. Biosciences Biotechnology Research Asia, 2015, 12(1), pp. 737–744.