Distributed System for Predicting Malicious Activity in Computer Networks

Denys Liubinetskyi[a], Antonina Kashtalian[a], Tomas Sochor[b], Andrii Selskyi[a] and Olexandr Klein[a]

[a] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*
[b] *Prigo University, Havirov, Czech Republic*

Abstract

This article introduce a self-organized that based on deep learning (DL) algorithms. A new self-organized incremental neural network FG-SOINN presented in the Python programming language. This self-organized neural network (SOINN) algorithm generates packets of nodes and edges that are cleaned through fixed, user-defined time intervals. This leads to sudden changes in the network structure. This is due to the fact that large regions of the network are removed with a fixed period before the resumption of the training. Such "oblivion" conflicts with a more gradual oblivion on a large period of time, which is usually observed in natural cognitive systems. In a self-organized incremental algorithm, the removal of nodes and edges is determined by two parameters. These settings should be improved for each existing program through cross-checking. This article proves that FG-SOINN significantly eliminates this shortcoming, considering removal of nodes and edges as a necessary part of the process of study. Three concepts have been developed and implemented to form FG: Time of idle nodes and edges, reliability and utility of a particular node or edges. It is using these concepts that this algorithm and system will remove nodes and fins. The FG technique or node wear technique determines when and after which node of the period it will be removed. But if they are not updated in the specified time, they are simply deleted. These characteristics make it attractive for the dynamics of the environment, which requires long-term training. To ensure system scalability, the network will be guided by the n parameter. The algorithm starts its work and initiates the network with empty sets of nodes.Next, the nodes are added to the weight vectors. After that, the winner vector (nearest node) and the second winner vector will be searched. Distance measurement formulas will be used to determine the distance.
The structure and results of the system demonstrate that this proposition can be adapted to the dynamic profile of network data for both normal and attack classes. The algorithm uses less resources, is faster, and has a higher detection rate than the teaching-teacher method than the traditional SOINN.

Keywords

DS, traffic analysis, DL, neural networks, harmful/malicious activity, self-organized system

1. Introduction

With the continuous growth of threats and attacks, it is quite a difficult task to accurately and timely detect malicious activity in computer networks. To date, many principles, methods, and systems for detecting network intrusions have been proposed. However, they face critical challenges due to the constant increase in new threats that current systems do not understand. Network activity refers to the interaction of different computers to achieve certain goals.

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: kiberplayer@gmail.com (D. Liubinetskyi); yantonina@ukr.net (A. Kashtalian); tomas.sochor@osu.cz; (T. Sochor); andriy.saa@gmail.com (A. Selskyi); oleksanderklein@gmail.com (O. Klein);
ORCID: 0000-0003-1575-4457 (D. Liubinetskyi); 0000-0002-4925-9713 (A. Kashtalian); 0000-0002-1704-1883 (T. Sochor); 0000-0002-7373-0472 (A. Selskyi); 0000-0002-1896-943X (O. Klein)

Internet traffic prediction is very important for tasks such as resource allocation, network planning, and detection of network anomalies caused by attacks [1]. An accurate prediction model can be used to detect security attacks in computer networks by comparing predicted traffic with actual traffic. [2] Additionally, predicting future traffic on a computer network based on current traffic allows the network manager to take action against attacks, congestion, disconnections, or downtime. Similar predictions can be made by modeling network input and output traffic as time series. Currently, there are several studies in this field [3, 4].

Studies of benign activity help establish a baseline [5] or characterize its growth. Unfortunately, it is difficult to understand the extent of these potential threats due to the decentralization of the Internet, so detecting malicious activity in computer networks becomes a rather important task. In addition, due to frequent cyberattacks, one can observe a tendency that they become more and more qualitative and skilled. Failure to prevent or detect such intrusions can have serious consequences for users of such a network. Preventing exposure to malicious activity requires a system that will recognize network connectivity patterns to classify known and unknown intrusions, but also requires periodic retraining to maintain high performance.

2. Statement of the problem in a general form and its connection with important scientific or practical tasks

Malicious cyber-attacks are creating serious security challenges that require a new, flexible and more robust intrusion detection system (IDS). An IDS is a proactive intrusion detection tool that is used to automatically detect and classify intrusions, attacks, or violations of security policies. Most of the methods for detecting malicious actions proposed in the literature are rule-based methods (signature matching) and predictive modeling methods (anomaly detection) [6], [7]. Rule-based methods typically use known malicious behavior as a baseline to compare against new behaviors known to indicate security breaches [6]. This is usually achieved by embedding heuristics to search for known patterns (signatures) in the network and/or audit data [8-11]. However, developing malicious activity scenarios that cover all patterns and/or invisible patterns (i.e., zero-day attacks) is quite difficult. This task is complicated by the presence of polymorphism techniques [12]. In addition, attackers may be aware of detection heuristics already in use by the engine and attempt to avoid them. Therefore, there is a need for more reliable methods that can be adapted during operation to prevent malicious activity.

In order to ensure the protection of the system's network, three important problems related to network security must be solved.

1. The first problem is related to the rapid increase in the amount of network data. This growth is due to the use of the Internet of Things (IoT) [13], cloud services and the rapid growth of the network of devices. Improving the methods of data analysis includes increasing the speed and reliability of the analysis process.

2. The second problem is that more accurate tracking and interpretation significantly increases the quality of the findings. NIDS (Network Instruction Detection System) analysis requires more context-specific observations that emphasize more abstract and higher-level observations. All behavior changes should be traceable to a specific user, operating system version, or application specific protocols.

3. The third threat of modern networks is the variety of protocols and massive data transfer in modern networks. In this case, there is an extremely high level of complexity when trying to distinguish between abnormal and normal behavior. This increases the likelihood of unreliable and inconsistent data and increases the potential for exposure to zero-day vulnerabilities.

Intrusion detection systems can be classified based on the used detection mechanism. Detection methodologies are divided into two types: Signature-based and anomaly-based. Signature-based detection uses pre-defined templates associated with known attacks and is distributed as a set of signatures. The following signatures are compared to network traffic patterns to detect potential attacks. Despite its effectiveness against known threats, this method cannot detect or prevent unknown attacks, nor can it maintain or update signatures for known or recently discovered threats.

You can decide that anomaly-based detection sets the baseline/normal level using statistically significant traffic.

Research on improving classification methods for intrusion detection systems has focused on evaluating alternative solutions to fundamental analysis, including neural networks [14], fuzzy logic [9, 15], genetic algorithms [16] and vector support machines [17]. As pointed out by previous studies, SOINN and incremental learning are indeed very effective approaches to the problem of malicious activity detection. The development of intrusion detection systems has shown that hybrid detection methods are more effective due to their ability to distinguish different types of network attacks. If the detection engine determines that the traffic is malicious or part of an attack, the packet may be logged or rejected and only partially forwarded to the intended recipient.

3. Statement of the problem

As the amount and size of data increases, learning algorithms are needed to efficiently handle large amounts of signals. Furthermore, a much more challenging task for unsupervised learning is to efficiently and robustly learn data from distributions in which noisy data exist. The main difficulty is that the learning algorithms have no prior knowledge about the distribution of the data as a whole. Thus, when the first few data of a particular distribution are received, the amount of data is insufficient to represent the entire distribution. Currently, learning algorithms cannot determine whether this data is noisy or normal. Therefore, for each iteration, existing methods (such as self-organizing map (SOM) [19], neural gas (NG) [20], etc.) must respond to new data and update the weight vectors of the corresponding neurons, which usually causes a critical deviation of the resulting topology mapping. Another problem is that in most self-organizing "growing-type" neural networks, such as the growing neural gas (GNG) [21], the number of neurons will constantly increase due to the growth of the strategy for identifying them. A large number of neurons increases the computational cost of finding the winning neurons at each iteration, which makes the training procedure inefficient. Therefore, solving such problems requires an algorithm that will avoid these problems, which will greatly improve the efficiency of the self-organized system.

A Self-Organizing and Incremental Neural Network (SOINN) is an unsupervised learning mechanism for unlabeled data. SOINN has already been used in other studies as a clustering method that processes controlled data [22]. SOINN offers unsupervised training for an incremental clustering method with relatively high processing speed at low computational cost. In addition, the complexity and size of the SOINN network is controlled and stabilized through "garbage" or the discovery of unnecessary nodes (neurons).

The technique of the collector or the method of wear of the node, determines when and after which node of the period it will be removed. But if they are not updated at the specified time, they are simply deleted. This property makes it attractive for the dynamics of any environment where continuous learning is required.

The task of the research is to create a distributed self-organized system for predicting malicious activity, namely an intrusion detection system based on artificial intelligence and a module based on the modified SOINN algorithm. Such a module should use a "garbage-forgetting" architecture based on the concepts of downtime, reliability, and utility.

Incremental learning algorithms allow the classifier to improve and expand its capabilities over time (as it processes more data), unlike an autonomous or batch learning algorithm, which provides that the classifier is subject to one group of input data. The dynamics of network data are constantly changing, and using static models will negatively affect detection, as in the case of teacher-trained algorithms.

Implementation of the set task will allow to determine, design and implement the system of prediction of malicious activity by using neural networks and SOINN methods.

4. The architecture of the self-organized prediction system

For the correct functioning of the work system aimed at detecting malicious activity, it is necessary to determine its architecture.

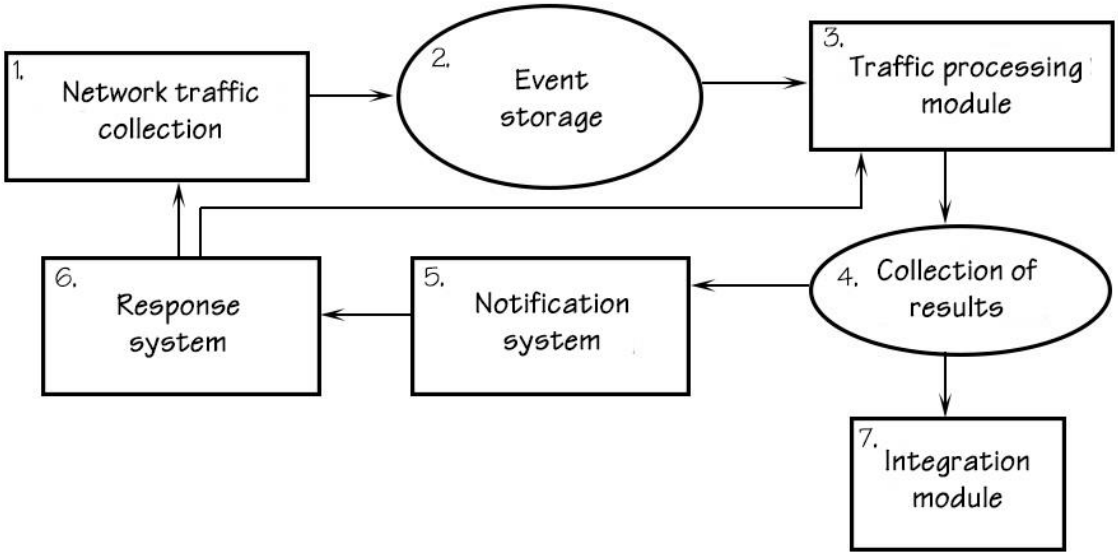

Figure 1: Architecture of the IDS system based on FG-SOINN

This system will consist of the following components:

1. Collection of network traffic, which is necessary to use all available information about devices used in the network. In addition, it will perform the function of converting the original network traffic in the required form (for calculating the necessary parameters) and recording data in the event store.

2. Event storage this system defines as a place to save information, which is then analyzed by the system for the presence of malicious traffic and network attacks. Each point of records in the storage will store information about data flows according to the parameters required by the traffic processing module.

3. The traffic processing module is needed as a system working on the FG-SOINN algorithm (forgetting garbage of self-organized incremental neural networks). This module will perform traffic analysis of all flow records that will be stored in the output event database using algorithms that are based on machine learning methods. At the end, for each record, the module determines the type of connection: normal or malicious, as well as the type of attack if it is detected.

4. The results of the data analysis will be recorded in the results database. The location of the results collection is presented as a separate database, which is used to save the anomalies detected.

5. The data should be used by the notification system as well as the compatibility module to extract and further use the analysis results.

6. The response system will act as a "garbage collector". Each record that was determined by the system as malicious will be destroyed, undefined records, that is, such records that are neither normal nor malicious, will be returned to the traffic processing module. This information will then be used for further reprocessing. After that, the system returns to the network traffic collection module, which continues the work cycle.

7. The integration module is an API for the ability to integrate with response systems, an interface for system interaction using http requests.

5. The main part

Self-organized incremental neural network (SOINN) is a mechanism of uncontrolled learning (or teaching without a teacher) for unmarked data. Unsupervised training (without a teacher) has two main objectives: clustering and studying of data topology. The purpose of clustering is to divide this dataset into multiple clusters, where each data pair in one cluster is more similar than two different clusters [23]. On the other hand, data topology learning can be described as follows: given a high-dimensional data distribution, it is necessary to project the input data into a topological structure, in which the topological contiguous units of the data in the input space are projected. Recently this

technique is widely used for intellectual analysis of data, vector quantization, image recognition, computer vision and many other related industries [24].

SOINN initiates a network with an empty set of nodes, then adds the first two nodes to the list using vector weights as two input vectors [25]. After the initializations, the neural network finds the winning or nearest node and the same second closest winning node. The distance S1 and S2 from each input to each node is measured by means of the equations (1) and (2). This formula is a general formula for measuring the distance between the layers.

$$s_1 = argmin_{C \in A} dist(x, w_c) \quad (1)$$

$$s_2 = argmin_{C \in A - \{s_1\}} dist(x, w_c) \quad (2)$$

If the input vector corresponds to the same cluster as the winning nodes, SOINN updates the weight vector of the node and its neighbors with the input weight vector and connects it to the rebromine node. If no matching vector is found in the input vector definition, a new node is added to the network.

The original SOINN is most often used for unsupervised learning. SOINN is used to learn the topological structure of the input data, it is able to grow gradually and take into account input patterns of non-stationary data distribution. It can separate low-density overlapping classes and detect the underlying structure of clusters that are contaminated with noise. It automatically learns the number of nodes and network structure, reports the number of clusters, and provides typical prototypes of each cluster.

Self-organizing incremental neural networks include a family of neural networks, common for which is that they find a topological reflection of incoming data in the network structure by means of competitive training [26].

That understand that SOINN reflects p - measurable input $x = x_{1,}x_2 \dots x_p$ where $x_i \, is \, i^{-M}$ meaning of signs for a separate node in a non-oriented graph. The display corresponds to the point in p - three-dimensional space of sign. Training in SOINN means adaptation of the topology map: Nodes can move, join with other nodes, remain single or be removed, and the edges between nodes can be created or deleted. The node should be considered as a microcluster of incoming cases, which are about one to one. Edges can be considered as consolidated links between associated nodes, such as nodes belonging to one (macro)cluster.

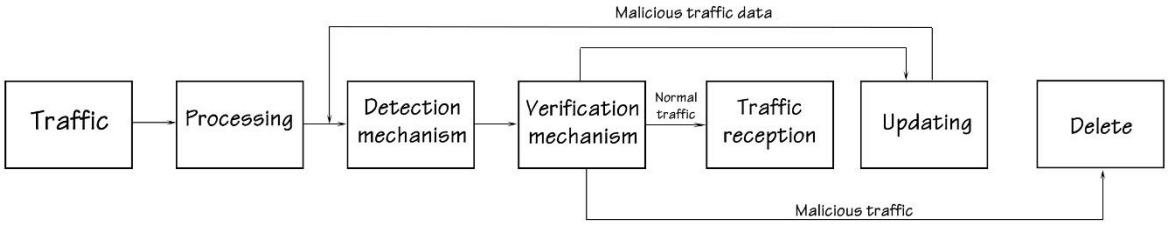

Figure 2: System information flows based on FG-SOINN

Figure 2 shows how information flows are processed in the system, where traffic is checked for classification in the system. The system starts with the fact that the incoming traffic collected by the system during the work cycle is fed to the pre-processing module. It is this module that captures and processes incoming traffic in real time. The defined and identified connections are processed to extract and further use the functions. These functions will be the input vector of the detection mechanism - FG-SOINN. The NSL-KDD dataset was performed for the study, and all attribute information is available in [27]. The information is then fed to the malicious activity detection engine, which, after processing, transmits the traffic information to the data validation engine module. The validation module then determines how the system is improved and improved by confirming the expected label. It is at this stage that traffic is divided into harmful and normal traffic. After confirming that the traffic is normal, it is transferred to the user for further work. However, if the system has confirmed that a malicious traffic has been detected, the data validation module will forward forecasts to the update module and remove the traffic from the data set. The system update

module operates in two phases: the active phase and the update phase. In the real-time phase, he makes decisions based on what his capabilities were at the time. In the active phase, the module makes decisions using the data it possesses at the time. In the system update phase, the module updates data with failed predictions that will improve its capabilities in the future. The phases will be executed simultaneously, when necessary, or alternate according to the data of the network traffic.

The basic concept of the proposed algorithm is the gradual creation of the mechanism of network protection. The initial stage of network training uses a relatively small sample containing attributes that are necessary for further correct detection of attacks by the system. Further, after the first cycles of work, the system is re-plenished with definite, undefined and unknown data. Thanks to the last two, the system can be re-learned and improved to improve the security mechanism. To enhance its capabilities, the underlying detection engine must be able to classify network data into multiple classes, not only as to whether it is associated with an attack or not, but also as to the type of attack.

The main part of the system consists of Clustering and Classifier. The clustering block contains a pair of nSOINN [28], which are understood and used by the algorithm classes to compress the data received by the preprocessing module. The classification part takes the output of n-SOINNs, creates the input data for the SVM classifier for each class of the classification pre-run.

In SOINN, node and edge removal is defined by two parameters that must be optimized for each available program using cross-validation or similar resampling approaches. FG-SOINN overcomes this drawback by treating node and edge removal as an integral part of the learning process. Unlike simple SOINN, FG-SOINN has a unique function that tries to bind the nodes that are likely to represent the signal rather than the noise. Binding depends on the reliability of the node.

6. Experimental research

The evaluation of the received structure was carried out on NSL-KDD data set [29], which is an improved version of the well-known data set KDD'99. Despite its age, the data set is still a de facto alternative to methods and tools of comparative analysis aimed at providing effective systems of intrusion detection. The training of the system begins with the study of 125,972 traffic instances of the NSL-KDD data set. As can be seen in the figure, the time of initial training and first acquisition of knowledge of the neural network, which will use the FG-SOINN algorithm, is 18 min 57 sec. The number of nodes defined by the network is 819, the number of edges is 2308. With these data, the system will continue to work.

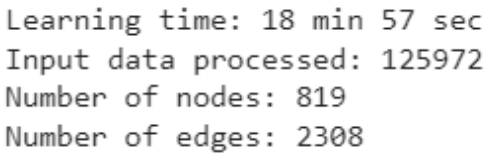

Figure 3: First training phase

Due to the fact that the input data set includes a large sample of data, five binary classes were created four for attacks and one for normal traffic:

- Normal traffic.
- DoS attacks - denial of service attack.
- Probing attack is a starting phase of attack on web resources and web applications. During this attack, the attacker collects information about the structural features of the web application (pages, settings, etc.) and an additional supporting infrastructure (operating system, databases, etc.).
- Remote to user (R2L) — an R2L attack occurs when an attacker tries to send packets to a machine over a network that does not have an account [30].
- User to Root (U2R): The main attack in U2R is a buffer overflow, which copies too much data to a static buffer without checking that the data is written exactly to the program [31, 32].

Several system assessment metrics were used to further compare the results of the study:

- True TPR or Detection Ratio (DR): The ratio between the number of correctly predicted attacks and the total number of attacks, also called Detection Ratio (DR) (3).

$$TPR = TP / (TP + FN) \quad (3)$$

- Level of false positive results (FPR): Ratio between the number of common cases misclassified as attacks and the total number of common cases (4).

$$FPR = FP / (FP + TN) \quad (4)$$

- Negative indicator (FNR): The anomaly could not be identified and classified as normal traffic (5).

$$FNR = FN / (FN + TP) \quad (5)$$

- Positive predictive value (PPV): Probability of intrusion detection if IDS gives alarm (6).

$$PPV = TP / (TP + FP) \quad (6)$$

- Negative predicted value (NPV): Probability of no invasion when IDS does not give alarm (7).

$$NPV = TN / (TN + FN) \quad (7)$$

- The classification coefficient (CR) or accuracy: The percentage of all these correctly predicted cases to all cases, also known as accuracy (8).

$$CR = (TP + TN) / (TP + TN + FP + FN) \quad (8)$$

- Base rate (B): The probability that input data is an attack (9).

$$B = (TP + FN) / (TP + TN + FP + FN) \quad (9)$$

- Intrusion Detection capability (CID): The ratio of common information between input and output and input entropy (10).

$$CID = (H(X) - H(X \mid Y)) / H(X) \quad (10)$$

- Complicated formula for defining the entropy using PPV and NPV (11).

$$h_{x_y} = -b * (1 - fnr) * math.\log(ppv) - b * fnr * math.log(1 - npv) - (1 - b) * (1 - fpr) * math.log(npv) - (1 - b) * fpr * math.log(1 - ppv) \quad (11)$$

The first phase of the update showed the following results (fig. 4). Accuracy 96.44%, detection rate 97.7%, classification as attack 4.65%, error rate 2.3%, possibility of invasion detection 78%. For comparison, also pay attention to the eighth update (fig. 5) and the twelfth update (fig. 6).

```
update_phase(model=s, data=train_1, labels=y_train_1)
```

```
Time for the upgrade phase: 1 min  9 sec
Accuracy (percentage of correctly predicted cases): 96.45%
Detection percentage (TPR): 97.7%
Ability to detect intrusions (CID): 78.31 %
False positive rate (FPR - normally classified as seizures): 4.65%
False negative rate (FNR) - attacks are classified as normal): 2.3%
```

Figure 4: First update phase

```
update_phase(model=s, data=train_8, labels=y_train_8)
```

```
Time for the upgrade phase: 1 min 20 sec
Accuracy (percentage of correctly predicted cases): 97.5 %
Detection percentage (TPR): 97.93%
Ability to detect intrusions (CID): 83.23%
False positive rate (FPR - normally classified as seizures): 2.88%
False negative rate (FNR) - attacks are classified as normal): 2.07%
```

Figure 5: Eighth update phase

```
update_phase(model=s, data=train_12, labels=y_train_12)
```

```
Time for the upgrade phase: 1 min 19 sec
Accuracy (percentage of correctly predicted cases): 98.23%
Detection percentage (TPR): 98.31%
Ability to detect intrusions (CID): 87.15%
False positive rate (FPR - normally classified as seizures): 1.83%
False negative rate (FNR) - attacks are classified as normal): 1.69%
```

Figure 6: The Phase of the twelfth update

Table 1. Results of testing the results of the implemented system based on FG-SOINN

№	Time, sec	Accuracy,%	TPR,%	FPR,%	FNR,%	CID,%
1	69	96.45	97.7	4.65	2.3	78.31
2	72	96.73	97.4	4.6	2.26	79.02
3	150	96.83	97.47	3.72	2.53	79.88
4	75	97.66	98.1	2.72	2.49	84.09
5	80	97.67	97.55	2.23	2.45	84.1
6	79	97.7	97.61	2.71	2.39	82.83
7	77	97.67	97.62	1.91	2.81	83.99
8	80	97.5	97.93	2.88	2.07	83.23
9	84	97.52	97.96	2.87	2.24	83.77
10	87	97.92	98.24	2.35	1.76	85.47
11	80	98.06	97.65	1.57	2.15	86.19
12	79	98.23	98.31	1.83	1.69	87.15

If compare the data from the first, eighth and twelfth updates, the difference in the improvement of attack detection becomes clear. In the first, unknown threats were 2.3%, while in the twelfth they became 1.69%. The results of all twelve updates are shown in Table 1. According to Table 1, the CID intrusion detection result has been continuously increased during twelve update cycles, and the error level has been constantly changed and the system has achieved the best results for twelve updates.

It should be noted that after the initial training for each round of renewal the subset is checked on the training FG-SOINN, and only soon forecasts return to the system. For visualization the data received after the training were presented on the linear graph, which is shown in Figure 7. The blue color represents the percentage value of the system results, which is learned with the FG-SOINN algorithm, and the red one is SOINN.

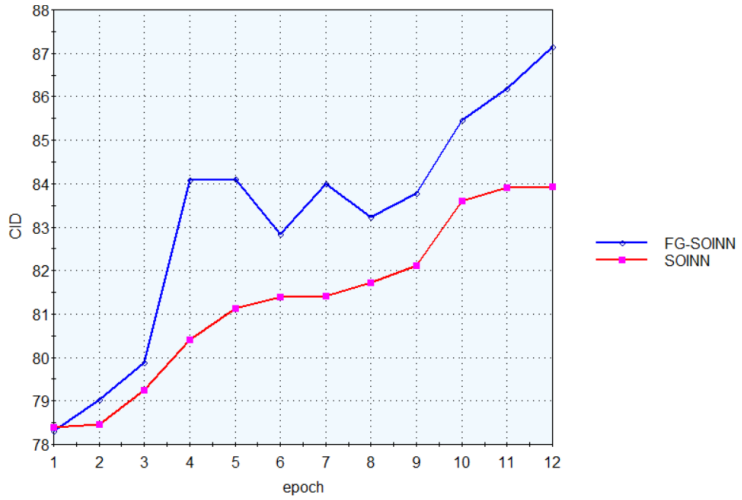

Figure 7: Comparison of systems based on algorithms FG-SOINN and SOINN

These indicators show that the accuracy of the FG-SOINN algorithm is greater than SOINN. The quality of intrusion detection by IDS is improved with each new phase, which confirms the efficiency of the proposed algorithm modification.

7. Conclusion

The paper proposes a self-organized system for predicting malicious activity based on incremental learning. It is performed with the help of an intrusion detection system that learns with the help of neural networks. The structure of the developed system and its results show that this proposal is confidently adaptable with the nature of the dynamic network data profile. This system uses fewer resources, runs faster and has a higher detection level than teacher training or simple SOINN. Analysis of the proposed FG-SOINN algorithm, which will be used in the distributed intrusion detection system, showed that the efficiency of each subsequent phase of updating and developing the system increases by 1-4%, which gives a good result on huge volumes of data. If we provide this system with many unsuccessful predictions, we not only achieve incremental learning with great and promising accuracy, but also create an effective structure. It will simply not be necessary to add a full set of data to the system. By providing the system with failed predictions, we not only achieve incremental learning with promising accuracy, but also an efficient framework, i.e., instead of feeding it the full dataset. Although system learning time increases with the increase in update input, system update and operating modes can either work concurrently (simultaneously), or update mode can switch when the operational phase is inactive (i.e. no input for detection). In this way, it acquires abilities by learning new input data or failed classifications.

8. References

[1] W. Jiang and L. Jiayun Graph neural network for traffic forecasting: A survey. Expert Systems with Applications (2022) 117921.

[2] J. Bao, H. Bechir W. Weng-Keen Iot device type identification using hybrid deep learning approach for increased iot security 2020 International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2020.

[3] A.H. Gandomi, C. Fang and A. Laith Machine learning technologies for big data analytics Electronics 11.3 (2022) 421.

[4] S.S. Lin, S. Shui-Long and Z. Annan Real-time analysis and prediction of shield cutterhead torque using optimized gated recurrent unit neural network. Journal of Rock Mechanics and Geotechnical Engineering 14.4 (2022) 1232-1240.

[5] Q. Gong, and G. Chenwei A Baseline modeling algorithm for internet port scanning radiation flows. 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP). IEEE, 2021.

[6] J. Snehi et al. Global intrusion detection environments and platform for anomaly-based intrusion detection systems //Proceedings of Second International Conference on Computing, Communications, and Cyber-Security: IC4S 2020. – Springer Singapore (2021) 817-831.

[7] M. Alsoufi, et al. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. Applied sciences 11.18 (2021) 8383.

[8] S. Lysenko, O. Savenko, K. Bobrovnikova, A. Kryshchuk, B. Savenko. Information technology for botnets detection based on their behaviour in the corporate area network, Communications in Computer and Information Science 718 (2017) 166–181

[9] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic, Communications in Computer and Information Science 370 (2013) 243-254

[10] 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), Cracow, Poland, September 22-25.- 2021. pp. 64-69

[11] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A. Nicheporuk, A Technique for detection of bots which are using polymorphic code, Communications in Computer and Information Science 431 (2014) 265-276.
[12] O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk, Metamorphic Viruses Detection Technique based on the the Modified Emulators. CEUR-WS 1614 (2016) 375-383
[13] Y. Otoum, N.Amiya, As-ids: Anomaly and signature based ids for the internet of things. Journal of Network and Systems Management 29 (2021) 1-26.
[14] MA Khan, HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. Processes 9.5 (2021) 834.
[15] M. Almseidin, A.-S. Jamil and A. Mouhammd Anomaly-based intrusion detection system using fuzzy logic. 2021 International Conference on Information Technology (ICIT). IEEE, 2021.
[16] Z. Halim, et al. An effective genetic algorithm-based feature selection method for intrusion detection systems. Computers & Security 110 (2021) 102448.
[17] M. Mohammadi, et al. A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. Journal of Network and Computer Applications 178 (2021) 102983.
[18] Z. Wang, et al. An efficient network intrusion detection approach based on deep learning. Wireless Networks (2021) 1-14.
[19] X. Qu, et al. A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile networks and applications 26 (2021) 808-829.
[20] MS Al-Daweri, A. Salwani and AZ Khairul, A homogeneous ensemble based dynamic artificial neural network for solving the intrusion detection problem. International Journal of Critical Infrastructure Protection 34 (2021) 100449.
[21] F. Ardilla, AS Azhar and K. Naoyuki Batch Learning Growing Neural Gas for Sequential Point Cloud Processing. 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2022.
[22] M Zhu, et al. Attention-based federated incremental learning for traffic classification in the Internet of Things. Computer Communications 185 (2022) 168-175.
[23] S Comert, et al, Hopfield neural network based on clustering algorithms for solving green vehicle routing problem. International Journal of Industrial Engineering Computations 13.4 (2022) 573-586.
[24] A. Protogerou, et al. A graph neural network method for distributed anomaly detection in IoT. Evolving Systems 12 (2021) 19-36.
[25] RW Ng, et al. An improved self-organizing incremental neural network model for short-term time-series load prediction. Applied Energy 292 (2021) 116912.
[26] J.D Nunes, et al. Spiking neural networks: A survey. IEEE Access 10 (2022) 60738-60764.
[27] F. Masoodi, Machine learning for classification analysis of intrusion detection on NSL-KDD dataset. Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.10 (2021) 2286-2293.
[28] Z. Chiba, et al. A Deep Study of Novel Intrusion Detection Systems and Intrusion Prevention Systems for Internet of Things Networks. Procedia Computer Science 210 (2022) 94-103.
[29] A. Devarakonda, et al. Network intrusion detection: a comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. Journal of Physics: Conference Series. 2161 1. (2022).
[30] Beulah J. R. et al. Enhancing Detection of R2L Attacks by Multistage Clustering Based Outlier Detection //Wireless Personal Communications. – 2022. – Т. 124. – №. 3. – С. 2637-2659.
[31] A. Sachenko, V. Kochan, V. Turchenko, V. Tymchyshyn and N. Vasylkiv, "Intelligent nodes for distributed sensor network," IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No.99CH36309), Venice, Italy, 1999, pp. 1479-1484 vol.3, doi: 10.1109/IMTC.1999.776072
[32] V. Sreerag, et al. Reinforce NIDS Using GAN to Detect U2R and R2L Attacks. Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021. Springer Singapore, 2022.