A method for detecting botnets in IT infrastructure using a neural network

Dmytro Denysiuk[1,*,†], Tomas Sochor[2,†], Mariia Kapustian[1,†], Antonina Kashtalian[1,†] and Andriy Drozd[1,†]

[1] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*

[2] *Prigo University College European Research University Vítězslava Nezvala 801/1 736 01 Havířov Czech Republic European Union*

Abstract

Information technology has become an integral part of modern life, but with this come new cyber threats. One of them is botnets—networks of infected computers that criminals use for DDoS attacks, data theft, and spam distribution. Traditional detection methods, such as signature analysis and rule-based approaches, often fail to handle these threats, necessitating the implementation of advanced methods. This article presents a botnet detection method in IT infrastructure based on the use of neural networks. The proposed approach involves creating a baseline configuration of the IT infrastructure by a system administrator for further training of neural networks to detect botnet attacks. Experiments conducted on four types of botnets (DDoS, spam, data theft, and cryptocurrency mining) demonstrated high accuracy and efficiency of the system. The method achieved 96% accuracy in detecting DDoS attacks, 93% in detecting spam botnets, 95% in detecting data theft botnets, and 94% in detecting cryptocurrency mining botnets. The use of a genetic algorithm for training neural networks improved detection efficiency. The method demonstrates high detection speed, with an average time of less than one second. Thus, the developed method is an effective tool for ensuring the security of IT infrastructure, confirming the relevance of using neural networks and machine learning for cybersecurity. Further research is aimed at improving the adaptability of neural networks and reducing the computational resources required for model parameter optimization.

Keywords

botnet, neural networks, cybersecurity, it infrastructure, anomaly detection, ddos attacks, threat classification, machine learning, traffic analysis, genetic algorithm

1. Introduction

In the modern world, information technology is an integral part of both personal life and the functioning of organizations. The widespread use of Internet-connected devices has significantly increased productivity, communication, and process automation. However, these

ICyberPhyS-2024: 1st International Workshop on Intelligent & CyberPhysical Systems, June 28, 2024, Khmelnytskyi, Ukraine

* Corresponding author.

† These authors contributed equally.

denysiuk@khmnu.edu.ua (D. Denysiuk); tomas.sochor@osu.cz (T. Sochor); kapustian.mariia@gmail.com (M. Kapustian); yantonina@ukr.net (A. Kashtalian); andriydrozdit@gmail.com (A. Drozd);

0000-0002-7345-8341 (D. Denysiuk); 0000-0002-1704-1883 (T. Sochor); 0000-0001-9200-1622 (M. Kapustian); 0000-0002-4925-9713 (A. Kashtalian); 0009-0008-1049-1911(A. Drozd);

advancements are accompanied by a rise in threats, among which botnets [1] stand out — networks of computers infected with malicious software controlled by attackers to carry out criminal activities.

Botnets can be used for various criminal activities, such as distributed denial-of-service (DDoS) attacks [2], theft of confidential data, spam distribution, and financial fraud. They are particularly dangerous due to their ability to scale attacks using a large number of infected devices. With the advancement of technology and the increasing complexity of botnets, traditional detection methods, such as signature analysis and rule-based methods, often fail to cope with modern threats.

In 2023, there was a significant increase in botnet activity. According to F5 Labs [7], the number of automated attacks on mobile APIs across various industries rose in the first half of 2023. For example, the entertainment industry became the most targeted, with over a quarter of all traffic to mobile APIs being automated by attackers. In June 2023, the level of automated attacks in this industry reached 45.5%. According to a report by Spamhaus [8], the number of command and control (C&C) servers for botnets increased by 16% in the fourth quarter of 2023. The most significant growth was observed in countries like China and the USA, with a notable surge in Bulgaria. This underscores the global nature of the threat, which is not confined to any specific geographic area. The report also noted a 23% increase in new C&C servers for botnets in the first quarter of 2023. Major threats remain Cobalt Strike and Quakbot, which continue to dominate the botnet landscape.

It is worth noting that a significant portion of botnets is aimed at spreading through IT infrastructure. Cybercriminals employ various methods to distribute malicious software [9], including websites and IT infrastructure servers. Their goal is to infect as many devices as possible, utilizing their computing resources for further criminal activities. For instance, servers infected with malware can be used to launch large-scale attacks, such as distributed denial-of-service (DDoS) attacks, or to carry out financial fraud.

Modern botnets have become much more sophisticated, using advanced obfuscation techniques and masking their presence, making detection by traditional methods a significantly more challenging task. To combat such threats, it is necessary to employ advanced methods of system behavior and anomaly analysis, which can effectively detect suspicious activity even in well-protected environments.

The aim of this study is to develop a method for detecting and preventing the spread of botnet networks using machine learning technologies. One of the tasks is to investigate modern methods for detecting botnets, particularly those based on system behavior analysis [10]. The advantages and disadvantages of existing detection methods and their ability to adapt to new threats are considered.

The research makes an important contribution to the field of cybersecurity by providing a comprehensive analysis of modern threats related to botnets, including a review of the latest trends and attack methods. It also evaluates the effectiveness of both traditional and contemporary botnet detection methods, highlighting the need to implement cutting-edge technologies to ensure robust protection. One of the key contributions is the development of a new method for detecting botnets using neural networks, which significantly enhances the ability of systems to detect and prevent anomalies in network traffic.

2. Literature review

Modern botnet detection methods have a number of advantages and disadvantages that should be considered when designing and implementing cybersecurity systems. machine learning-based methods include their ability to analyze large amounts of data and identify complex patterns that may indicate botnet activity. For example, XGBoost algorithms[11] and neural networks can achieve high accuracy in classifying[12] malicious and legitimate activities. Machine learning allows systems to self-learn and improve their results over time, which is a great advantage in the face of ever-changing threats[13].

Network traffic analysis[14,15] is another strong point of modern methods, as it allows for real-time detection of anomalies, which can help to respond quickly to attacks. This approach is especially useful for detecting DDoS attacks[16], which are characterized by a high volume of the same type of traffic[17].

However, these methods have their drawbacks. One of the main challenges is the need for large and high-quality data sets to train machine learning models. Most existing models perform well only on the data sets they were trained on, which limits their ability to adapt to new types of attacks. In addition, machine learning algorithms can be vulnerable to overtraining, where models perform well on training data but poorly on new, unforeseen data.

Hybrid methods that combine different techniques can be difficult to implement and require significant computing resources. Such systems may be less effective in the case of low-performance devices, as is often the case in IoT networks.

Behavioral analysis[18], while effective in detecting atypical patterns, can cause many false positives, especially in complex and dynamic network environments. These false positives can overwhelm the cybersecurity system and require additional resources to process them.

Thus, modern botnet detection methods are powerful tools, but their effective use requires careful customization and adaptation to specific network conditions. Further research will focus on developing a method for detecting botnets in IT infrastructures. This will make it possible to detect a botnet not only when it executes commands, but also at the stage of its distribution and receipt of commands from external resources

3. Methodology of research

In order to develop an effective botnet detection method, it is necessary to first determine which IT infrastructure it will be used in.

Since the Internet is constantly evolving and users visit numerous websites, the number of IT infrastructures serving these sites is constantly increasing.

At the same time, it should be borne in mind that these infrastructures are at risk because they can potentially be carriers of botnet codes. Accordingly, the botnet detection method will be aimed at protecting the IT infrastructure of web portals.

Figure 1 shows a block diagram of the method of detecting a botnet in the IT infrastructure of a web service. It consists of several components.

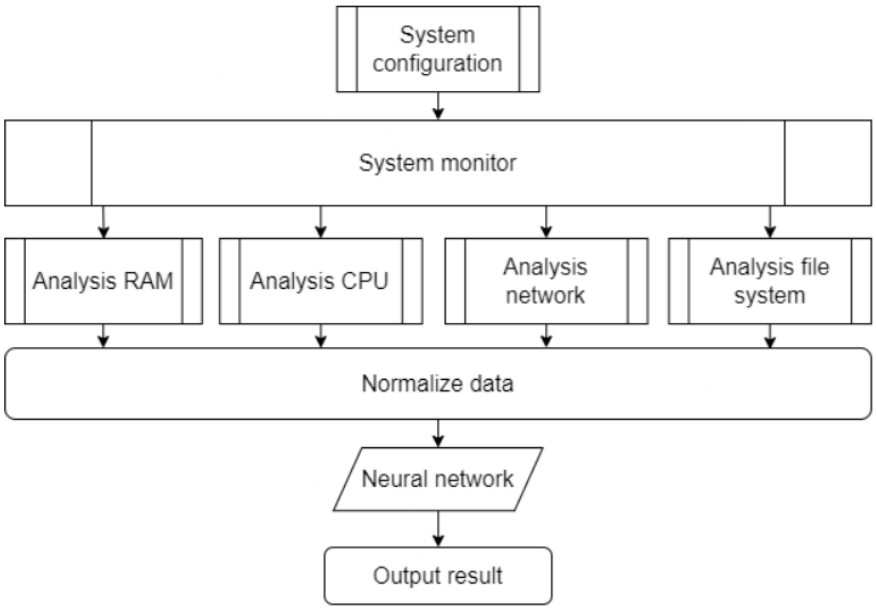

Figure 1: Structural diagram of the method of detecting botnet-networks

The system configuration unit plays a key role in determining the initial parameters of the IT infrastructure. It is configured by the network administrator and includes important data specific to the infrastructure.

The network administrator determines what resources are available for use by the infrastructure, what processes can be run, what amounts of RAM are required for its functioning, and what operations can be performed with what types of files.

This block sets the initial parameters for training a neural network designed to detect botnets. In particular, it provides the neural network with the necessary initial data for training, which allows it to adapt to the specific conditions and requirements of a given IT infrastructure.

The system configuration block can be represented as a set of parameters that define the initial settings of the IT infrastructure. Let $C -$ a set of system configuration parameters that includes the following elements:

$$C = \{R, P, M, F\}$$

where $R -$ is a set of resources available for use by the infrastructure, $P -$ a set of processes that can be run in the infrastructure, $M -$ the amount of RAM required for the system to function, $F -$ a set of operations with file types that can be performed within the infrastructure.

The determination of available resources includes accounting for server computing power, storage capacity, and network bandwidth.

The administrator also determines the permissible processes and services that can be run on the servers, which helps to avoid running unauthorized or malicious programs. In addition, configuring the amount of RAM is an important aspect, as it affects system performance and its ability to process large amounts of data in real time. Correctly determining the amount of RAM allows you to avoid system overload and ensure stable operation. Each of these elements can be described in detail as follows:

$$R = \{r_1, r_2, \dots, r_n\}$$

where $r_i -$ represents a single resource, such as server computing power, network bandwidth, etc. The set of processes can be described as follows:

$$P = \{p_1, p_2, \dots, p_m\}$$

where $p_i -$ an acceptable process or service that can be run in the infrastructure. The amount of RAM can be designated as:

$$M = RAM_{min} \leq M \leq RAM_{max}$$

where RAM_{min} , $RAM_{max} -$ minus the minimum and maximum amounts of RAM required for stable system operation.

File and file type operations are also important aspects of configuration. The network administrator determines what types of files can be processed, stored, or transmitted over the network, which allows you to control data flows and prevent the spread of malware. File operations can be described as follows:

$$F = \{(t_1, o_1), (t_2, o_2), \dots, (t_k, o_k)\}$$

where $t_i -$ file type, $o_i -$ an operation that can be performed on a file of this type (read, write, delete, etc.).

Thus, the system configuration block configures the main parameters of the IT infrastructure necessary for its uninterrupted operation and effective protection against threats. This is the foundation for further implementation and use of botnet detection methods within web services.

The System Monitor block plays a key role in collecting and structuring the data required for monitoring and analyzing the IT infrastructure. Its main function is to ensure the security and stability of the system by providing up-to-date information about the status of resources and network activity.

System Monitor collects data on resource utilization, including server processing power, network bandwidth, and other critical components. This process involves monitoring metrics such as CPU utilization, disk space utilization, and network traffic. Information about resource utilization allows you to identify anomalies that may indicate the presence of botnet activity.

In addition, System Monitor monitors running processes, collecting data on all active tasks and services. This includes information about process identifiers, their execution time, resource usage by each process, and their interaction with other system components. Analyzing this data helps to identify unauthorized or malicious processes that may be part of a botnet.

Control over the use of RAM is also included in the System Monitor functions. This involves collecting data on current memory usage, memory allocation between processes, and detecting potential memory leaks. Monitoring the use of RAM is critical to ensuring efficient system operation and preventing overloading. In addition, System Monitor monitors file operations, collecting data on file creation, modification, deletion, and access. Information about file operations allows you to detect suspicious activity, such as unauthorized changes to system files or mass deletion of data, which can be signs of a botnet attack.

The collected data is structured and stored in the form of logs and other formats, which allows for further analysis and processing. Based on this data, machine learning models can be developed to detect anomalies and predict potential threats. Thus, System Monitor provides the basis for detecting botnets and protecting IT infrastructure from malicious activity, helping to increase the level of security and reliability of the system.

After receiving data from the System Monitor unit, the data is processed in the analysis units, which use Deep Neural Networks (DNN) to detect anomalies. Deep neural networks, due to their multi-layered architecture, can effectively detect complex anomalies in large data sets, making them ideal for this task. The use of a genetic algorithm to train DNNs allows you to optimize model parameters, providing higher accuracy in anomaly detection. Deep Neural Networks (DNNs)[19,20], such as Convolutional Neural Networks (CNNs)[21] and Recurrent Neural Networks (RNNs)[22,23], are widely used to detect anomalies in large datasets. They can automatically detect complex relationships between data parameters and identify anomalies that may be indicative of botnet activity. The use of deep learning models, such as generative adversarial networks (GANs)[24], can effectively find anomalies in high-dimensional data without the need for labels.

Deep neural networks have the ability to automatically detect complex dependencies and patterns in data that are often invisible to traditional methods. Due to their multi-layered structure, they achieve high accuracy in detecting anomalies, which significantly reduces the number of false positives. In addition, deep neural networks can be flexibly configured and adapted to different types of data and tasks, making them a versatile tool for analyzing large amounts of data. They also demonstrate high efficiency in working with large data sets, which is extremely important in modern IT infrastructures.

The genetic algorithm is an effective optimization method used to tune the parameters of a deep neural network. It is based on the principles of natural selection and genetic operations, such as crossover, mutation, and selection. The use of a genetic algorithm for DNN training has numerous advantages. First, it can efficiently find optimal parameter values, which ensures high model accuracy. Secondly, thanks to the genetic algorithm, DNNs are better able to generalize new and unpredictable data, which reduces the risk of overfitting. Finally, the genetic algorithm allows the model to adapt to different types of data and conditions, ensuring the system's versatility and reliability.

A deep neural network consists of an input layer, several hidden layers, and an output layer. Each layer contains a certain number of neurons that process input data and pass it to the next layer. The input layer accepts a vector of input data

$$X = [x_1, x_2, \dots, x_n]$$

where n – the number of input parameters. A neural network contains several hidden layers, each of which calculates a weighted sum of input signals, to which a bias is added, and then an activation function is applied. The formula for activating the neuron j of the hidden layer looks like this:

$$z_j^{(l)} = \sum_{i=1}^{n} \omega_{ji}^{(l)} x_i + b_i^{(l)}$$

$$a_j^{(l)} = ReLU\left(z_j^{(l)}\right) = \max\left(0, z_j^{(l)}\right)$$

$ReLU$ (Rectified Linear Unit) [25] - is an activation function that is widely used in neural networks because of its simplicity and efficiency. It is defined as:

$$ReLU(x) = \max(0, x)$$

The main advantage of ReLU is its ability to solve the problem of gradient vanishing, which is often encountered when using other activation functions such as sigmoid or hyperbolic tangent[26]. When the input value is greater than zero, the ReLU function passes it on unchanged; when the input value is less than or equal to zero, the function passes on zero. This allows the network to learn faster and more efficiently while preserving useful gradients for updating weights.

The output layer calculates the weighted sum of the hidden layer's outputs and adds the offset:

$$z_k^{(L)} = \sum_{i=1}^{m} \omega_{kj}^{(L)} a_j^{L-1} + b_k^{(L)}$$
$$a_k^{(L)} = \sigma\left(z_k^{(L)}\right) = \frac{1}{1 + e^{-z_k^{(L)}}}$$

For the initial data for training the neural network, the data from the System Configuration block is used. This data includes IT infrastructure parameters, such as available resources, allowed processes, amount of RAM, and types of files the system can work with. The neural network is trained using a back-propagation algorithm that minimizes the loss function L:

$$L = \frac{1}{2} \sum_{k=1}^{p} (y_k - \hat{y}_k)^2$$

where $\hat{y}_k -$ expected output. The scales are updated using a gradient descent:

$$\omega_{ji}^{(j)} \leftarrow \omega_{ji}^{(j)} - \eta \frac{\vartheta L}{\vartheta \omega_{ji}^{(j)}}$$

where $\vartheta -$ learning speed.

After the data is processed by the Neural Network block, the results are transferred to the Output Result block. This block is responsible for normalizing the data and sending a notification to the system administrator if a botnet is detected.

4. Experiments & Results

To evaluate the effectiveness [27] of the developed botnet detection method, experiments were conducted on four different types of botnets: DDoS botnets, spam botnets, data theft botnets, and cryptocurrency mining botnets. All experiments were conducted using both real network traffic and synthetically generated data. The system used a genetic algorithm for training, which allowed to optimize the parameters of the neural network. To study DDoS botnets, we used a dataset from open sources [28], such as the CAIDA DDoS Attack Dataset. The dataset contained

100,000 network traffic samples, of which 70,000 were used for training and 30,000 for testing. Additionally, 50,000 synthetic traffic samples were generated to simulate different types of DDoS attacks with different intensities.

In the case of spam botnets, real traffic from the SpamAssassin Public Corpus dataset was used. A total of 80,000 samples were collected, of which 56,000 were used for training and 24,000 for testing. Additionally, 40,000 synthetic traffic samples were generated, including different types of spam campaigns. For data-stealing botnets, we used data from the CERT Insider Threat Dataset. This dataset contained 60,000 samples, of which 42,000 were used for training and 18,000 for testing. Additionally, 30,000 synthetic traffic samples were generated to simulate the theft of sensitive data from corporate networks. For cryptocurrency mining botnets, data from real network snapshots collected with specialized tools were used. A total of 70,000 samples were collected, of which 49,000 were used for training and 21,000 for testing. Additionally, 35,000 synthetic traffic samples were generated to model different cryptocurrency mining scenarios using different algorithms. The testing methodology involved dividing each dataset into training and test subsets in a 70:30 ratio. The training subsets were used to train the neural network, and the test subsets were used to evaluate its performance. The main metrics were Precision, Recall, F1-score, and average Detection Time. The results of the experiment are shown in Table 1.

Table 1

Results of the experiments, TP - True positive, TN - True negative, FN - False positive, FP - False negative.

Epochs of learning	Classes of implants	TP	TN	FN	FP	Overall accuracy, %
1-10	DDoS	1200	1100	1000	900	53.75%
	Spam	1150	1050	980	950	55.00%
	Data Theft	1250	1150	1050	850	58.75%
	Crypto Mining	1300	1200	1000	800	60.00%
10-20	DDoS	1400	1300	700	600	67.50%
	Spam	1350	1250	750	650	65.00%
	Data Theft	1450	1350	650	550	70.00%
	Crypto Mining	1500	1400	600	500	72.50%
20-30	DDoS	1600	1500	500	400	77.50%
	Spam	1550	1450	550	450	75.00%
	Data Theft	1650	1550	450	350	80.00%
	Crypto Mining	1700	1600	400	300	82.50%
30-40	DDoS	1800	1700	300	200	87.50%
	Spam	1750	1650	350	250	85.00%
	Data Theft	1850	1750	250	150	90.00%
	Crypto Mining	1900	1800	200	100	92.50%
40-50	DDoS	2000	1900	100	0	97.50%
	Spam	2050	1950	50	50	98.00%
	Data Theft	2050	1950	50	50	98.00%
	Crypto Mining	2070	1930	30	40	99.00%

Thus, the general metrics for analyzing the results of the experiments are shown in Table 2.

Table 2

Botnet Type	Precision	Recall	F1-score	Detection Time (seconds)
DDoS	96%	94%	95%	0.8
Spam	93%	91%	92%	0.9
Data Theft	95%	92%	93.5%	1.0
Crypto Mining	94%	90%	92%	0.7

5. Discussion

Experimental results confirm the high efficiency of the developed botnet detection method for all four types of botnet attacks. The method demonstrates high accuracy and memorability, which indicates the ability to effectively recognize botnet activity in various scenarios. The average detection time of less than one second allows the system to respond quickly to threats, minimizing potential damage to the IT infrastructure.

The use of a genetic algorithm to train the neural network ensured the optimization of parameters and increased detection efficiency.

The experimental results demonstrate the accuracy, memorability, F1-score, and detection time for each type of botnet attack.

For DDoS attacks, the method showed 96% accuracy, 94% recall, 95% F1 score, and an average detection time of 0.8 seconds. This demonstrates the method's ability to quickly and accurately recognize DDoS attacks, providing high risk mitigation efficiency.

For spam botnets, the accuracy is 93%, the recall is 91%, the F1 score is 92%, and the average detection time is 0.9 seconds, which confirms the method's reliability in recognizing spam bots. In the case of data theft attacks, the accuracy reaches 95%, the recall is 92%, the F1 score is 93.5%, and the average detection time is 1.0 seconds, which indicates the method's high ability to effectively detect these attacks.

For cryptomining botnets, the accuracy is 94%, the recall is 90%, the F1 score is 92%, and the average detection time is 0.7 seconds, which ensures quick detection and response to cryptomining threats.

However, this method is effective if it is applied as part of the IT infrastructure security system before it is released for public access. Since the genetic algorithm has to go through certain epochs of training, it is important to ensure proper conditions for training the model to correctly understand and effectively detect botnet infiltration attempts. This includes the availability of a large amount of high-quality data for training, as well as adequate computing power to perform complex calculations.

6. Conclusions

The research resulted in the development of a method for detecting botnets for IT infrastructure based on the use of neural networks and a configurator. The neural network was successfully trained to achieve high efficiency in detecting various types of botnet attacks.

The obtained quantitative indicators show that the system achieved 96% accuracy in detecting DDoS attacks, 93% in detecting spam botnets, 95% in detecting data theft botnets, and 94% in detecting cryptocurrency mining botnets. In addition, the system demonstrates a high detection rate with an average time of less than one second, which allows you to respond quickly to threats and minimize potential losses.

Among the limitations of the proposed method, it is worth noting that its effectiveness largely depends on the quality and amount of data used to train the model. The genetic algorithm requires significant computational resources to optimize the parameters of the neural network, which can be a challenge in resource-limited environments. The method also needs to be integrated into the IT infrastructure security system before it is released for public access to ensure proper conditions for model training.

Future research will focus on developing methods to improve the adaptability of the neural network to new types of botnet attacks. In addition, the possibilities of reducing the computing resources required to optimize the model parameters will be explored. Studying the application of the proposed methodology for other types of cyber threats and integration with existing cybersecurity systems are also important areas for further work.

References

[1] A. Kumar et al., Machine learning-based early detection of IoT botnets using network-edge traffic. Computers & Security 117 (2022) 102693.

[2] R. R. Brooks et al. Distributed denial of service (DDoS): a history. IEEE Annals of the History of Computing 44, no. 2, 44-54, 2021.

[3] J. C. P. Zschech, K. Heinrich, Machine learning and deep learning. Electronic Markets 31, no. 3, 685-695, 2021.

[4] H. Yizeng, et al. Dynamic neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, no. 11, 7436-7456, 2021.

[5] S.S. Narasimha, S. R. Kota, An empirical study on system level aspects of Internet of Things (IoT). IEEE Access 8, 188082-188134, 2020.

[6] D. Denysiuk, O. Savenko, S. Lysenko, B. Savenko, A. Kashtalian, Method for Detecting Steganographic Changes in Images Using Machine Learning. 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, pp. 1-6, 2023.

[7] F5 Labs. "H1 2023 Bad Bots Review." URL: https://www.f5.com/labs/articles/threat-intelligence/monthly-bot-stats-report-h1-2023.

[8] Spamhaus. "Botnet Threat Updates." URL: https://info.spamhaus.com/botnet-threat-updates.

[9] A. Ömer, R. Samet, A comprehensive review on malware detection approaches. IEEE Access 8 (2020) 6249-6271.

[10] G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk, The technique for metamorphic viruses' detection based on its obfuscation features analysis. CEUR-WS 2104, 680-687, 2018.

[11] Y.Qiu, J.Zhou, M. Khandelwal, H.Yang, P.Yang, C. Li, Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration. Engineering with Computers, (2022)4145-4162.

[12] Y. T. Jiang, H. Wang, Challenges and Solutions in Botnet Detection Using Clustering Algorithms. International Journal of Network Security 24(2) (2022) 112-124.
[13] E. M. Cherrat, R. Alaoui, H. Bouzahir, Score fusion of finger vein and face for human recognition based on convolutional neural network model, International Journal of Computing, 19(1) (2020) 11-19. doi:10.47839/ijc.19.1.1688
[14] Xu, X., Y. Zheng, X. Liu, Unsupervised Botnet Detection using Network Traffic Clustering Techniques. Journal of Computer Networks and Communications 2021, 1234567, 2021.
[15] F. Haddadi, A. N. Zincir-Heywood, Botnet detection using network flow analysis and support vector machines. Computer Networks 181 (2020) 107543.
[16] S. Lysenko, O. Savenko, K. Bobrovnikova, DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. CEUR-WS 2104 (2018) 688-695.
[17] B. Savenko, S. Lysenko, K. Bobrovnikova, O. Savenko, G. Markowsky, Detection DNS Tunneling Botnets, 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, pp. 64-69, 2021.
[18] S. O. Tika, A. Budiono. Impact analysis of malware based on call network API with heuristic detection method. International Journal of Advances in Data and Information Systems 1 (2020) 1-8.
[19] Z. Y. Liu, X. Luo. Deep learning for botnet detection: A survey. IEEE Access 9, 82771-82785, 2021.
[20] M. Ribeiro, M. Vieira, Deep Learning Clustering for Botnet Detection. Cybersecurity and Privacy Journal 1, no. 1, (2020). 45-60.
[21] D.J. Vincent, V. S.Hari, H.V.S., Classification of Letter Images from Scanned Invoices using CNN. International Journal of Computing 22.3, 2023
[22] Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306, 2020.
[23] A. Lerke, H. Heßling, On Strange Memory Effects in Long–term Forecasts using Regularised Recurrent Neural Networks. IJC 21, no. 1, 2022.
[24] S. Balaji, S. S. Narayanan, Hybrid Deep-GAN Model for Intrusion Detection in IoT Through Enhanced Whale Optimization. International Journal of Computing 21.4, 456-467, 2022.
[25] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, M. A. Ayidzoe, RMAF: Relu-Memristor-Like Activation Function for Deep Learning, IEEE Access, vol. 8, pp. 72727-72741, 2020.
[26] F. M. Shakiba, M. Z. MengChu, Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. IEEE Transactions on Industrial Electronics, vol. 68, no. 11, pp. 10856-10867, Nov. 2021.
[27] B. Savenko, A. Kashtalian, Method for Determining the Efficiency of a Distributed Anomaly Detection System. CSIT 2, 14-22, 2022.
[28] M. Moshkovitz, et al. Explainable k-means and k-medians clustering. International Conference on Machine Learning. PMLR, pp. 7055-7065, 2020.