Method of detecting cyberattacks on communication channels based on spectral clustering and machine learning methods

Serhii Danchuk[1,*,†], Olena Geidarova[1,†], Andrii Nicheporuk[1,†], Andrzej Kwiecien[2,†]

[1] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*
[2] *Silesian University of Technology, Akademicka str., 2A, Gliwice, Poland*

Abstract

This article proposes a method for detecting cyberattacks on communication channels in TCP/IP networks based on spectral clustering and machine learning methods. The proposed method includes the following steps: data collection and normalization, application of spectral clustering to obtain clusters, training a machine learning classifier, and testing the classifier. Spectral clustering is used to detect DDoS attacks and requires diverse network traffic data to build a similarity matrix for clustering. Feature sets such as the number of server requests over time, total volume of transmitted data, unique IP addresses, average server processing time for requests, and failed connection/authentication attempts are used to construct the similarity matrix. The proposed detection model combines spectral clustering with machine learning classifiers such as Random Forest, J48, and Naive Bayes. During training, the data set is divided into clusters using spectral clustering, and a Random Forest classifier is trained for each cluster. During detection, spectral clustering determines the membership of test data to clusters, and the corresponding Random Forest classifier determines whether the sample is normal or anomalous. The performance of the model is evaluated using previously unused data to assess its effectiveness. Overall, the proposed approach demonstrates promising results in detecting DDoS attacks on communication channels.

Keywords

Cyberattacks, spectral clustering, random forest

1. Introduction

In today's digital world, where communication networks are an essential component of virtually all aspects of life and activities, the importance of detecting cyberattacks on communication channels becomes critical. The increasing reliance on technology and the widespread use of the Internet lead to an increase in digital information exchange, but also

IntelITSIS'2024: 5th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 28, 2024, Khmelnytskyi, Ukraine
* Corresponding author.
† These authors contributed equally.
segey.danchuk.p@gmail.com (S.Danchuk); geydarova@ukr.net (O. Geidarova);
andrey.nicheporuk@gmail.com (A. Nicheporuk); andrzej.kwiecien@polsl.pl (A. Kwiecien);
0009-0003-4510-0363 (S.Danchuk); 0000-0002-7253-893X (O. Geidarova);
0000-0002-7230-9475 (A. Nicheporuk); 0000-0003-1447-3303 (A. Kwiecien);

to a rise in the number of threats and attacks aimed at disrupting the integrity, confidentiality, and availability of these channels.

Cyberattacks on communication channels can lead to serious consequences, including loss of confidential information, privacy breaches, financial losses, and even threats to national security. Attackers, using various methods and techniques, attempt to exploit vulnerabilities in communication infrastructure to gain unauthorized access to systems and data.

Therefore, detecting cyberattacks on communication channels is a necessary element of digital security. The use of advanced technologies and network activity analysis methods allows for timely recognition and mitigation of potential threats to ensure the reliability and security of communication infrastructure. The importance of this process becomes particularly significant in the context of constantly evolving cyber threats and the emergence of new attack vectors, which require continuous improvement of security measures and effective detection methods.

Moreover, the problem is complicated by the use of code obfuscation techniques, including obfuscation and metamorphism. The technique of detecting metamorphic viruses is an important tool in cybersecurity. It analyzes substitution and obfuscation functions to detect changes in virus code. Additionally, the method of modified emulators is a promising direction in this field. It allows analyzing virus behavior in a virtual environment, facilitating their detection. Furthermore, the technique of detecting bots with polymorphic code is an important aspect of cybersecurity. It analyzes the polymorphic code of botnets, making their detection more complex. Finally, the method of searching for equivalent functional blocks helps detect viruses by analyzing their functional structure and relationships between code parts.

The use of clustering methods and random forest is a relevant strategy for detecting cyberattacks on communication channels, as they allow adaptation to the complex conditions of cyberspace and ensure the effectiveness of threat detection and response.

2. Related works

The problem of cyberattacks on communication channels of information systems poses a serious threat to data security and the functioning of any information system. Cyberattacks can affect various types of communication channels, including wired, wireless, satellite, and others. Therefore, modern methods for detecting cyberattacks on communication channels of information systems include various approaches to identifying unusual or malicious activities in the network. They are oriented towards detecting anomalies, attacks, intrusions, or vulnerabilities. Some of the approaches include intrusion detection systems (IDS), which monitor traffic for unusual patterns or anomalies, signature-based methods for detecting known attacks, machine learning algorithms for analyzing and detecting anomalies, event log analysis for detecting unexpected changes or unauthorized access, as well as the use of intelligent systems to recognize unusual or malicious behavior in the network. Let's take a closer look at some methods of detecting cyberattacks on communication channels.

The author of the study [1] proposes ForkDec, a system for detecting mining attacks based on a fully connected neural network aimed at effectively deterring attackers. The neural network consists of a total of 100 neurons (10 hidden layers and 10 neurons per layer), trained on a training set containing approximately 200,000 fork samples. The dataset used to train the model is generated by a Bitcoin mining simulator previously created. Evaluation experiments show that ForkDec has practical value and research prospects. Thus, the advantages of the presented solution include high accuracy due to the use of artificial neural networks and a large dataset for training. However, the possibility of system overload when working with a large number of neurons and voluminous data can be a negative aspect.

In [2], a VGA detection system based on enhanced machine learning is proposed. Specifically, a semi-supervised learning methodology is utilized, employing a hybrid combination of algorithms. This includes a heuristic clustering method based on linear fragmentation of group classes. The ELM methodology is employed as an algorithm for obtaining hidden variables using convex optimization. However, it should be noted that the use of such complex methods may lead to a high level of computational complexity and system resource requirements.

Another approach proposed by the authors [3] fully leverages the benefits of deep reinforcement learning in decision-making and develops a continuously learning training system. Particularly, an industrial control network and deep reinforcement learning training characteristics were applied to develop a unique reward mechanism. Additionally, an industrial anomaly detection system based on deep reinforcement learning was constructed. The algorithm was tested on a dataset of industrial control of a gas pipeline at the University of Mississippi. Experimental results showed that the convergence speed of this model is significantly higher than that of traditional deep learning methods. However, the high complexity of implementing such systems and their limited applicability may be a negative factor.

In [4], authors proposed CyDDoS, an integrated Intrusion Detection System (IDS) for combating DDoS attacks on communication channels, which combines a set of feature engineering algorithms with a deep neural network. Feature selection for the ensemble is based on five machine learning classifiers used to identify and extract the most relevant features utilized in the predictive model. This approach enhances model performance by processing only a subset of relevant features, thereby reducing computational demands. The model's performance is evaluated on the CICDDoS2019 dataset, consisting of regular traffic and DDoS attack traffic. Various evaluation metrics such as accuracy, F1-Score, and preservation are considered to justify the effectiveness of the proposed structure against state-of-the-art IDS. The advantage of the solution proposed by the authors is its high efficiency in processing a limited set of features. However, limited flexibility in feature selection and dependence on the quality of the dataset may limit real-world applicability.

Research authors [5] introduced XNBAD, a novel unsupervised network behavior anomaly detection system. XNBAD integrates higher-order host states in the context of dynamic host interactions with conversation models to represent behavior. Higher-order states can better generalize latent interaction patterns but are difficult to obtain directly. Thus, XNBAD employs a Graph Neural Network (GNN) for automatic feature generation of

higher-order features from extracted base series. We evaluated the detection effectiveness of XNBAD on the publicly available ISCX-2012 dataset. To report detailed and accurate experimental results, we carefully curated the dataset before evaluation. The results show that XNBAD effectively detected various attack behaviors and significantly outperformed existing representative methods, at least in terms of overall weighted AUC improvement. However, its dependency on the accuracy of graph model construction and large computational resources may be negative factors.

In [6], a Markov chain model is utilized for detecting anomalous intrusion in wireless networks. Through parameter analysis and selection, the experimental results are ideal, and various evaluation methods are compared and analyzed. Firstly, this method can easily distinguish normal data from anomalies, reducing the processing time by approximately 50% compared to the previous method. The new method proposed in this article has characteristics of simple computation, low algorithm complexity, and easy online detection. This method addresses the drawback that single-stage Markov chain analysis and detection methods cannot strictly establish the nature of the Markov chain, has lower algorithm complexity than multi-step Markov chain analysis and detection methods, and is simpler than computing parameters of hidden Markov chain models. Thus, the presented work based on the Markov chain model for detecting anomalous intrusion in wireless networks is characterized by simplicity of computation and the ability for easy online detection. However, limitations in defining the nature of the Markov chain and the difficulty in constructing accurate models may affect its accuracy.

Authors [7] propose a network security defense mechanism to address the network collapse problem that can be caused by DDoS attacks. Specifically, based on formulating stochastic queue dynamics with jump noise, a mechanism characterizing queue behavior on routers is presented to stabilize queue length during DDoS attacks with constant speed. Applying stochastic control theory to analyze queue dynamics performance during DDoS attacks with constant speed, certain clear conditions are established under which the instantaneous queue length converges to any given target on the route. Simulation results demonstrate the satisfaction of the proposed defense mechanism with a sharp contrast to contemporary Active Queue Management (AQM) schemes. The network security mechanism for stabilizing queue length during DDoS attacks is based on stochastic queue dynamics. It distinguishes itself by its ability to stabilize queues during DDoS attacks, but its effectiveness may be limited depending on network conditions and other parameters.

In [9], authors propose streamlining flows associated with IoT for efficient Intrusion Detection Systems (IDS). As a result, machine learning algorithms generate accurate results from large and complex datasets. The machine learning results can be utilized for anomaly detection in IoT network systems. Several machine learning classifiers and a deep learning model are employed in this document for intrusion detection using seven datasets from the TON_IoT telemetry dataset. The proposed IDS achieved an accuracy of 99.7% using datasets from Thermostat, GPS Tracker, Garage Door, and Modbus through a voting classifier. However, implementation constraints and dataset requirements may complicate its application in various scenarios.

Attacks on hardware, such as side-channel analysis attacks or fault injection attacks, can significantly compromise or even eliminate the desired level of security of the

corresponding infrastructure. One of the most dangerous types of attacks of this kind is Voltage Glitch Attacks (VGA), which can alter the planned behavior of the system. By effectively manipulating voltage at a certain time, an error can be introduced that alters the intended behavior and bypasses system security features or even obtain confidential information such as encryption keys by analyzing incorrect microcode outputs. This study proposes a VGA detection system based on enhanced machine learning. Specifically, a semi-supervised learning methodology is used, employing a hybrid combination of algorithms. This includes a heuristic clustering method based on linear fragmentation of group classes. Conversely, the ELM methodology is used as an algorithm for obtaining hidden variables using convex optimization.

Thus, the review of known methods for detecting cyberattacks on communication channels confirms that modern methods for detecting cyberattacks on communication channels possess significant effectiveness. Specifically, they are capable of detecting potentially dangerous anomalies and attacks with a relatively high level of efficiency. However, the presence of certain drawbacks, such as limited flexibility in dealing with diverse scenarios and high computational costs, raises doubts about the universality of the application of these methods. Therefore, the development of new methods and approaches for detecting cyberattacks on communication channels remains a highly relevant task.

3. Detection method of cyberattacks on communication channels based on a combination of spectral clustering and random forest

The proposed method for detecting cyberattacks on communication channels in TCP/IP networks based on spectral clustering and machine learning methods includes the following steps:

The method for detecting cyberattacks on communication channels involves analyzing the communication channels, specifically TCP/IP connections. Clustering method is utilized for analysis. For detecting DDoS attacks using spectral clustering, it is crucial to have diverse network traffic data, which is used to construct a matrix suitable for further processing using clustering algorithms. The following features were used to create the similarity matrix: the number of requests to the server within a certain time frame, the total data volume transmitted by all network packets, the number of unique IP addresses interacting with the server, the average time the server spends processing requests, the number of failed connection or authentication attempts. NSL-KDD dataset [14] is used. In practice, intelligent data collection [26] can also be applied. Since some data is discrete, min-max normalization is applied.

The detection model proposed in this article is based on the semi-supervised learning approach, specifically spectral clustering which receives normalized data [15]. Spectral clustering utilizes the properties of graph spectral theory to group data points that interact more with each other than with other points. The collected data is clustered for further analysis to determine which group they belong to. A classifier is applied to the obtained set of clusters to determine if the data is anomalous. The model also includes the use of the

random forest algorithm, J48, Naive Bayes. These are classifiers that determine if the data in the cluster is anomalous.

During the training phase, the defined dataset S consists of (X_i, Y_i), $i = 1, 2, \ldots, N$, where X_i represents an N-dimensional matrix, and $Y_i = \{0,1\}$, where 0 indicates a normal flow, and 1 indicates an anomalous flow. During training, the dataset is initially divided into k non-overlapping clusters using spectral clustering. Then, for each cluster separately, a random forest is trained on the corresponding data. During the detection phase, spectral clustering is used to determine which of the k clusters the test data sample belongs to. Then, using the corresponding random forest corresponding to the selected cluster, it is determined whether the test data sample is normal or anomalous.

Afterward, testing is conducted on a portion of the data that was not used in training to observe the performance results. The overall algorithm is presented in Figure 1.

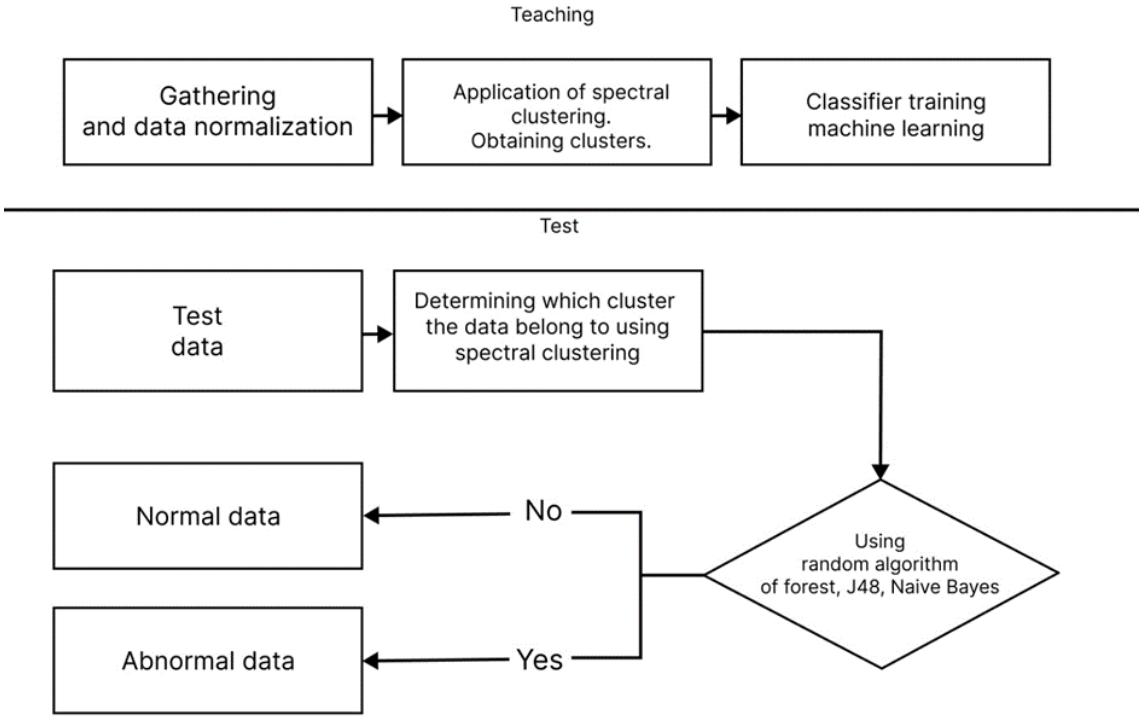

Figure 1: Classifier Training and Testing Algorithm.

3.1 Data Normalization

Before constructing the similarity matrix to perform spectral clustering on the data, it is important to normalize the data to eliminate the potential influence of differences in feature scales. We will use Min-Max normalization [16]. This method transforms values to fit within a range between 0 and 1. It can be useful when preserving relative distances between values is important. However, this method may be sensitive to outliers.

Min-Max normalization is a method of scaling data to a specific range, typically from 0 to 1. The normalization process involves the following steps:

- Determine the minimum (min) and maximum (max) values for each column of data.
- Subtract the minimum value from each value in the column (this centers the data around 0).
- Divide the obtained value by the difference between the maximum and minimum values in the column (this scales the data to fit within the range from 0 to 1).

Mathematically, the process of min-max normalization can be expressed as follows:

$$X_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where: X_{normal} - normalized value, X - original value, X_{min} - minimum value in the column, X_{max} - maximum value in the column.

This process ensures scaling the data so that their distribution falls within the range of 0 to 1, while preserving the relative distances between values. This allows the model to better capture the data features and improve their convergence during training.

3.2 The model of spectral clustering algorithm

This article utilizes a clustering algorithm based on spectral analysis, which theoretically is used to establish spectra. Compared to traditional clustering algorithms, spectral clustering can better partition the data samples into clusters with high similarity regardless of the sample space. The working principle of the spectral clustering algorithm is as follows. Firstly, the data of the sample set are transformed into a similarity matrix, which reflects the similarity between the sample data. Next, the eigenvalues and eigenvectors of the matrix are computed. Finally, a feature vector is selected, which can effectively cluster the data. This algorithm can converge to the globally optimal solution [17].

Given x data points z1, . . . , zn and a similarity function $f(||zi - zj||, \sigma)$, the weight matrix H is defined:

$$Hij = f(||zi - zj||, \sigma) \quad (2)$$

The similarity between two data points depends on both the distance between them and the scaling parameter σ; for example, the Gaussian similarity function:

$$f(||zi - zj||, \sigma) = e^{-||zi-zj||^2/2a^2} \quad (3)$$

This scaling parameter is commonly used. It determines the local structure of connections between data points. The degree matrix D is defined as follows:

$$\text{Hii} = \sum_{j=1}^{n} \text{Lij} \quad (4)$$

and the Laplacian matrix K is defined as follows:

$$K = H - L \quad (5)$$

The eigenvalues and eigenvectors of the matrix K are then used for data clustering; Laplacian normalization before computing the spectral decomposition leads to more balanced clusters. The number of clusters k is a mandatory input parameter. We apply the spectral clustering algorithm proposed. Initially, the Laplacian is normalized as follows:

$$K_{SYM} = C^{-1/2} K C^{1/2} \tag{6}$$

Let W be an n by k matrix, whose columns are eigenvectors corresponding to the k smallest eigenvalues of Ksym. Then the rows of W are normalized to obtain a new matrix J:

$$J_{IJ} = \frac{W_{IJ}}{\sum_{J=1}^{N} W_{IJ}} \tag{7}$$

Now, considering the rows of J as a collection of n data points in R^k the k-means algorithm is applied to cluster the data.

3.3 Model of Random Forest Algorithm, J48, Naive Bayes

Random Forest [18] is based on the fundamental concept of ensemble learning for training a series of decision trees and refining them according to the characteristics of each tree. During the training of a random forest, attributes are randomly selected to enhance the relative independence of the formed decision trees, leading to improved performance. In traditional decision trees, when the number of nodes equals n, the choice of the best attribute is based on all n attributes of the nodes. In a random forest, each decision tree node is based on k randomly chosen attributes, where k is a crucial parameter for the degree of randomness. Additionally, the value of k can be 1 or d, corresponding to randomly selecting an attribute or using a method of selection via a standard decision tree.

From the training process of random forests, it can be seen that it makes only minor modifications to the ensemble process by adding an element of randomness to the selection of feature attributes based on random sampling and generalizes the final integration of random forests. This increase in randomness contributes to improved results. Therefore, due to its high performance and relatively low computational complexity, the random forest algorithm is utilized as the classifier algorithm in this work.

The C4.5 algorithm [19] is a classification algorithm that constructs decision trees based on information theory. It is an extension of the previous ID3 algorithm by Ross Quinlan, also known as J48 in Weka, where J stands for Java. Decision trees created by C4.5 are used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. This algorithm builds decision trees based on a set of training data, similar to how the ID3 algorithm does, using the concept of information entropy. The training data is a set S={s1, s2, …} of already classified samples. Each sample si consists of a p-dimensional vector (x1, i, x2, i, …, xp, i), where xj represents the values of attributes or features of the corresponding sample, as well as the class to which the sample belongs. To achieve the highest classification accuracy, the best attribute for splitting is the one with the most information.

Naive Bayes [20] is a set of supervised learning algorithms based on the application of Bayes' theorem with the "naive" assumption of conditional independence between each pair of features given the class variable. Classifiers can be extremely fast compared to more complex methods. Separating the distribution of conditional features of the class means that

each distribution can be independently estimated as a univariate distribution. This, in turn, helps alleviate problems arising from the curse of dimensionality.

The use of these classifiers determines whether the data in the cluster is anomalous.

4. Experiments

The experimental part included investigating the effectiveness of the proposed method. Let's take a closer look at the dataset used in the study, the analyzed attacks, and the metrics used to evaluate the experimental results.

4.1 Data Collection for Training

In this experiment, the NSL-KDD dataset is utilized, serving as an efficient benchmark for intrusion detection methods. With a substantial number of records in both the training and testing sets, NSL-KDD is used to conduct experiments on the full dataset without the need for randomly selecting a small sample. This ensures consistent and comparable results in evaluating different research works. The experiment utilized the following data from the dataset: the number of requests to the server within a certain time frame, the total volume of data transmitted by all network packets, the number of unique IP addresses interacting with the server, the average time the server spends processing requests, and the number of failed connection or authentication attempts.

4.2 Attacks on communication channels

In the experiment, three types of attacks were considered: DDoS attack, Brute Force attack on authentication, Slowloris attack.

DDoS attack is characterized by a high number of requests to the server within a certain period of time, leading to server overload and decreased availability. The total volume of data transmitted over the network also increases as attackers attempt to flood the network with malicious traffic.

Brute Force attack increases the number of failed connection or authentication attempts as attackers try an excessively large number of combinations to gain access to the system. The number of unique IP addresses interacting with the server may increase as attackers may use botnets or different proxy servers to conceal their identity.

Slowloris attack. During this attack, the average time the server spends processing requests increases as attackers keep connections open, delaying their termination and overloading server resources. The number of failed connection attempts may also increase as Slowloris tries to utilize all available connections to the server.

4.3 Training classifiers of Random Forest, J48, Naive Bayes

Training was conducted based on the k-cross validation principle [22], where each experiment utilized a subset of data not used in previous experiments. This model is used to test the trained model, where other datasets become available. Training of the model is done on the training set, and then the model is tested on k different subsets of this set. The

results of each experiment, i.e., the performance of the model, are averaged over all k experiments. Figure 2 illustrates the general principle of training using k-cross validation.

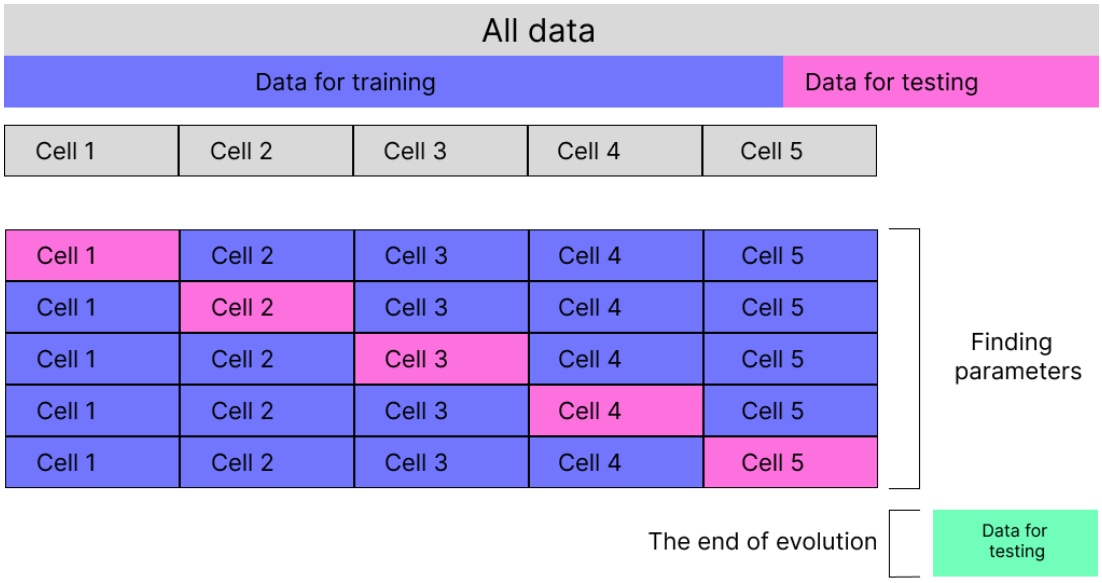

Figure 2: General principle of training using k-cross validation

4.4 Performance Metrics

The performance metrics used to evaluate the experiment results are calculated based on the standard confusion matrix.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{8}$$

$$TPR = \frac{TP}{TP + FN}, \tag{9}$$

$$FPR = \frac{FP}{FP + TN}, \tag{10}$$

In the formula, N represents the total number of data samples. Equation (8) represents the detection rate, which reflects the ratio of correctly classified anomalous data to the total amount of data. Equation (9) defines the true positive rate, indicating the proportion of correctly identified attack instances in all attack data. Equation (10) calculates the false positive rate, which reflects the ratio of incorrectly classified normal data to all anomalous data. The lower these metrics, the better the model's performance.

4.5 Results of the experiment

Table 1
Performance index for a random forest.

№	Attack type	Accuracy	TPR	FPR
1	DDoS	95	92	2.3
2	Brute Force	94	91	2
3	Slowloris	95	93	2.5

Table 2
Performance indicator for the J48 tree.

№	Attack type	Accuracy	TPR	FPR
1	DDoS	89	90	2.8
2	Brute Force	90	88	2.3
3	Slowloris	90	88	2.4

Table 3
After applying the Naive Bayes

№	Attack type	Accuracy	TPR	FPR
1	DDoS	92	88	2.2
2	Brute Force	91	89	2.4
3	Slowloris	94	90	2.5

In the provided table (Table 1,2,3), the performance of the spectral clustering method with the random forest algorithm is compared with spectral clustering methods using J48 and Naive Bayes algorithms. As shown in the table, the spectral clustering algorithm based on random forest proves to be more effective compared to J48 and Naive Bayes.

Experimental results indicate that the semi-supervised learning model proposed in this article achieves high accuracy, has a low false positive rate, and demonstrates good efficiency. This method is better suited for detecting channel attacks compared to other detection models.

According to the experimental results, the proposed method shows high effectiveness in detecting new types of network traffic data attacks and maintains a relatively low false positive rate. It outperforms J48 and Naive Bayes in all aspects such as sensitivity (TPR), specificity (FPR), and accuracy.

5. Conclusions

During the conducted research, a learning system based on spectral clustering methods and random forest algorithms was developed to enhance the efficiency of detecting DDoS attacks on communication channels. The principles of operation of the spectral clustering algorithm and the random forest algorithm were thoroughly analyzed in the work. Based on their advantages and principles of operation, they were combined with J48 and Naive Bayes algorithms to create a semi-supervised model for detecting DDoS attacks.

Additionally, a comparative analysis of the proposed semi-supervised model with other existing detection methods was conducted to verify its effectiveness. It was found that the proposed model demonstrates improvements in the detection of DDoS attacks while reducing the level of false positives. Therefore, it proves to be more useful for detecting DDoS attacks on communication channels.

References

[1] Z. Wang, Q. Lv, Z. Lu, Y. Wang, S. Yue, ForkDec: Accurate Detection for Selfish Mining Attacks. Security and Communication Networks 5959698 (2021) 8 , doi:10.1155/2021/5959698

[2] W. Jiang, Machine Learning Methods to Detect Voltage Glitch Attacks on IoT/IIoT Infrastructures. Computational Intelligence and Neuroscience 6044071 (2022) 7, doi:10.1155/2022/6044071

[3] Z. Liu, C. Wang, W. Wang, Online Cyber-Attack Detection in the Industrial Control System: A Deep Reinforcement Learning Approach. Mathematical Problems in Engineering 2280871 (2022) 9, doi:10.1155/2022/2280871

[4] I. O. Lopes, D. Zou, F. A Ruambo, S. Akbar, B. Yuan Towards Effective Detection of Recent DDoS Attacks: A Deep Learning Approach. Security and Communication Networks 5710028 (2021) 14, doi:10.1155/2021/5710028

[5] Z.-Quan Qin, H.-Z. Xu, X.-K. Ma, Y.-J. Wang, Interaction Context-Aware Network Behavior Anomaly Detection for Discovering Unknown Attacks. Security and Communication Networks 3595304 (2022) 24, doi:10.1155/2022/3595304

[6] H. Zhang, W. Lan, and D. Zhang, Anomaly Intrusion Detection of Wireless Communication Network-Based on Markov Chain Model. Security and Communication Networks 3255006 (2022) 1, doi:10.1155/2022/3255006

[7] K. Huang, L. Tan, G. Peng, Stability of SDE-LJN System in the Internet to Mitigate Constant-Rate DDoS Attacks. Security and Communication Networks 4733190 (2021) 17, doi:10.1155/2021/4733190

[8] A. Al Abdulwahid, Detection of Middlebox-Based Attacks in Healthcare Internet of Things Using Multiple Machine Learning Models. Computational Intelligence and Neuroscience 2037954 (2022) 15, doi: 10.1155/2022/2037954

[9] T. Saba, A. R. Khan, Tariq Sadad, Seng-phil Hong, Securing the IoT System of Smart City against Cyber Threats Using Deep Learning. Discrete Dynamics in Nature and Society 1241122 (2022) 9, doi:10.1155/2022/1241122

[10] W. Jiang, Machine Learning Methods to Detect Voltage Glitch Attacks on IoT/IIoT Infrastructures. Computational Intelligence and Neuroscience 6044071 (2022) 7, doi: 10.1155/2022/6044071

[11] M. Ghiasi, T. Niknam, Z. Wang, M. Mehrandezh, M. Dehghani, N. Ghadimi, A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future. Electric Power Systems 108975 (2023) 7, doi:10.1016/2022.108975

[12] G.J. Oyewole, G.A. Thopil Data clustering: application and trends. Artif Intell Rev 56 (2023) 6439–6475, doi:10.1007/s10462-022-10325-ytrends

[13] N. Boyko, R. Kovalchuk Data Update Algorithms in the machine learning system, Computer systems and information technologies, 1 (2023) 6-13

[14] M. Mittal,. K. Kumar, and S. Behal Deep learning approaches for detecting DDoS attacks: a systematic review. Soft Comput 27 (2023) 13039–13075 doi:10.1007/s00500-021-06608-1

[15] F. TÜRK, Analysis of Intrusion Detection Systems in UNSW-NB15 and NSL-KDD Datasets with Machine Learning Algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi 12 (2023) 465–477 doi: 10.17798/bitlisfen.1240469
[16] T. Mizutani Improved analysis of spectral algorithm for clustering. Optim Lett 15 (2021) 1303–1325 doi:10.1007/s11590-020-01639-3
[17] S. Gopal Krishna Patro, Kishore Kumar Sahu, Normalization: A Preprocessing Stage, 4, 2015, doi:10.48550/arXiv.1503.06462
[18] L. Fu, P. Lin, A. V. Vasilakos, S. Wang, An overview of recent multi-view clustering. Neurocomputing 402 (2020) 148-161, doi:10.1016/j.neucom.2020.02.104
[19] A .Sekulić, et al. Random Forest Spatial Interpolation. Remote Sensing, 12 (2020) 1687-1697, doi:10.3390/rs12101687
[20] N. Tambake, B. Deshmukh, A. Patange, Development of a low cost data acquisition system and training of J48 algorithm for classifying faults in cutting tool. Materials Today: Proceedings 72 Part 3 (2023) 1061-1067, doi:10.1016/j.matpr.2022.09.163tool
[21] A. J Wilson, B.S. Lakeland, T. J Wilson, T. Naylor, A naive Bayes classifier for identifying Class II. YSO 10 (2023) 23-46, doi:10.1093/mnras/stad301
[22] A.F. Otoom, W. Eleisah, E. E. Abdallah,Deep Learning for Accurate Detection of Brute Force attacks on IoT Networks. Procedia Computer Science 220 (2023) 291-298, doi:10.1016/j.procs.2023.03.038
[23] A. María P. Chacón, Isaac S.Ramírez, F. P. García Márquez, K-nearest neighbour and K-fold cross-validation used in wind turbines for false alarm detection. Sustainable Futures 6 (2023) 100132, doi:10.1016/j.sftr.2023.100132
[24] G. Markowsky, O. Savenko, S. Lysenko, A. Nicheporuk, The technique for metamorphic viruses' detection based on its obfuscation features analysis, CEUR Workshop Proceedings, 2104 (2018) 680–687.
[25] O. Pomorova, O. Savenko, S. Lysenko, A. Nicheporuk, Metamorphic Viruses Detection Technique based on the the Modified Emulators, CEUR Workshop Proceedings, 1614 (2016) 375-383.
[26] 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
[27] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko Approach for the Unknown Metamorphic Virus Detection, Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest Romania, September 21–23, 2017, pp. 71–76.