Research Article

MACHINE LEARNING METHODS FOR INTRUSION DETECTION IN COMPUTER NETWORKS: A COMPARATIVE ANALYSIS

Authors: Serkan Keskin, Ersan Okatan

To cite to this article: Keskin, S. & Okatan, E. (2023). Machine Learning Methods for Intrusion Detection in Computer Networks: A Comparative Analysis . International Journal of Engineering and Innovative Research ,5(3),268 -279 . DOI: 10.47933/ijeir.1360141

DOI: 10.47933/ijeir.1360141

To link to this artice: https://dergipark.org.tr/tr/pub/ijeir/archive

International Journal of Engineering and Innovative Research

http://dergipark.gov.tr/ijeir

MACHINE LEARNING METHODS FOR INTRUSION DETECTION IN COMPUTER NETWORKS: A COMPARATIVE ANALYSIS

Serkan Keskin[1*], Ersan Okatan[2]

[1] Burdur Mehmet Akif Ersoy University, Institute of Science and Technology, Department of Computer Engineering, Burdur, Turkey.
[2] Burdur Mehmet Akif Ersoy University, Gölhisar School of Applied Sciences, Department of Computer Technologies and Information Systems, Burdur, Turkey

[*]Corresponding Author: serkankeskin@isparta.edu.tr
(**Received:** 14.09.2023; **Accepted:** 12.10.2023)

https://doi.org/10.47933/ijeir.1360141

ABSTRACT: The widespread use of the Internet and the exponential increase in the number of devices connected to it bring along significant challenges as well as numerous benefits. The most important of these challenges, and the one that needs to be addressed as soon as possible, is cyber threats. These attacks against individuals, organisations and even entire nations can lead to financial, reputational and temporal losses. The aim of this research is to compare and analyse machine learning methods to create an anomaly-based intrusion detection system that can detect and identify network attacks with a high degree of accuracy. Examining, tracking and analysing the data patterns and volume in a network will enable the creation of a reliable Intrusion Detection System (IDS) that will maintain the health of the network and ensure that it is a safe place to share information. To have high accuracy in the prediction of the data set by using Decision Trees, Random Forest, Extra Trees and Extreme Gradient Boosting machine learning techniques. CSE-CIC-IDS2018 dataset containing common malicious attacks such as DOS, DDOS, Botnet and BruteForce is used. The result of the experimental study shows that the Extreme Gradient Boosting algorithm has an impressive success rate of 98.18% accuracy in accurately identifying threatening incoming packets.

Keywords: Intrusion detection systems, machine learning, network anomaly, IDS, XGBoost, CSE-CIC-IDS2018

1. INTRODUCTION

In contemporary life, information and communication technologies are integral. Countries rely heavily on their respective infrastructures. Currently, about two billion people use the Internet and Microsoft estimates that this number will exceed four billion by 2025 [1]. Given this massive expansion of Internet usage, it is becoming increasingly important to ensure "Cyber Security", also known as information technology.

As the complexity of network attacks continues to increase, Intrusion Detection and Prevention Systems (IDPS) are becoming indispensable defense tools. An Intrusion Detection System (IDS) serves to monitor the network by quickly identifying potential security threats. Upon detection, the system immediately sends an alarm to the administrator, alerting Intrusion Prevention Systems (IPS) that block traffic from the source address. An IDS is a popular choice for many organizations today. They are often an indispensable system for firewall manufacturers. A number of approaches have been seen in the development of IDS's, ranging

from rule-based systems, statistical methods, thresholding, artificial neural networks, data mining, fuzzy logic and artificial immune systems [1].

IDS's can be divided into two types: signature-based and anomaly-based. Signature-based IDS's store known attacks in a database and match incoming packets against this database, while anomaly-based IDS's do not rely on pre-existing attack information. Therefore, when detecting an attack, if it exists in the database, the attack is blocked. Commercial applications tend to rely heavily on signature-based IDSs. However, Anomaly Based Intrusion Detection Systems (ABIDS) are being developed using artificial intelligence techniques by training the system with both normal and anomalous data [2]. Numerous systems such as ACARM-ng, AIDE, Bro NIDS, Fail2ban, Samhain, Snort, Suricata, etc. utilize these methods.

When faced with unprecedented attacks, ABIDS is a reliable solution. Statistical methods and model classification techniques are used to model ABIDS based on pre-established knowledge. Datasets can contain a large number of features to represent a sample. However, it is not recommended to design a model with a large number of features if optimal performance is desired [3]. This approach can lead to a high computational cost and a higher error rate for the system. Therefore, different features are used in this study to improve the attack model.

The aim of this research is to create an ATSTS with optimum performance. It will be provided through training against common attacks of DOS, DDOS, Bot and BruteForce by using various features that are not available in normal network traffic. It was built using the "Python" programming language, using machine learning techniques in the Google Colab development environment. It was tested on the CSE-CICIDS2018 dataset shared by the Canadian Cyber Security Institute to ensure its effectiveness.

1.1. Literature Review

In the literature study, similar studies on CSE-CIC-IDS-2017, CSE-CIC-IDS-2018, KDD CUP99, UNR-IDD and NSL-KDD data sets were analysed. All of the data sets are on intrusion detection systems, and there are attack types such as Botnet, DOS, DDOS, Web attacks and BruteForce.

In a study on machine learning based network intrusion detection systems, the University of Nevada's Intrusion Detection Dataset (UNR-IDD) was analysed with a 96% success rate [4]. A success rate of 97.22% was achieved in a study examining network attacks, concept drift of data streams and changes in the statistical distribution of data [5]. Random Forest (RF) and Support Vector Machines classification models were used in the study in which feature selection was performed with stacked auto encoder and Select Best method. In the experimental study conducted on the NSL-KDD dataset, an accuracy rate of 99.67% was achieved [6]. In another study based on polynomial interpolation technique and statistical analysis, network anomaly detection was performed on CSE-CIC-IDS-2018 dataset. The success rate was 94.50% [7]. RF algorithm was the best classifier with 94.00% success rate in the study conducted with CSE-CIC-IDS-2017 database using iterative feature elimination and forward selection techniques [8].

In the study conducted after the transformation of one-dimensional network packets of the CSE-CIC-IDS-2018 dataset into vectors with a CNN-based classifier, it was observed that the average success rate was 95% [9]. In the study on the detection of various types of attacks with deep learning method, 96.97% success was achieved [10]. In the study in which CSE-CIC-IDS-

2018 dataset was used and the performance improvement obtained by solving the data imbalance was 92.41% with CatBoost and the RF algorithm remained at 89.88% [11]. In the study conducted on the KDD CUP99 data set in temporal convolutional networks, 97% success was observed. In this study, long short-term memory networks (LSTM) were more successful than other classical machine learning methods [12]. In the study examining the success factor of feature selection on intrusion detection systems, many algorithms were examined. Especially in the study where chi-square test and recursive feature elimination methods were used, the best result was found to be 98.79% with the extra trees model [13].

In an experimental study on the CSE-CIC-IDS-2018 dataset, where the features of the data were made more prominent by using Spearman's rank correlation coefficient, the RF algorithm achieved 98.8% success [3]. In the study presenting a two-level deep learning architecture for multiple attack classes, it is mentioned that it is more successful than a single-level approach. The average success rate of the two-level architecture is 98.25% in all attacks [14]. A comparative performance analysis was carried out in the study in which intrusions into the network on three different data sets were handled by deep learning method. As a result, NSL-KDD dataset achieved 97.89%, UNSW-NB15 dataset 89.99% and CSE-CIC-IDS2018 dataset 76.47% [15]. In the deep learning based study using convolutional neural network and recurrent neural network using KDD CUP99 and CSE-CIC-IDS2018 datasets, they identify intrusions on their own by training the experimental data. For the CSE-CIC-IDS2018 dataset, there is a success rate of 91.5% for the CNN model and 65% for the RNN model. For KDD CUP99, a 99% success rate for CNN and RNN model is mentioned [16].

In the study to analyse seven deep learning models, the performances of binary and multiple classifications were examined. After the experimental study, it is seen that the best performance is at 98% [17]. In the study using the long short-term memory (LSTM) model of the RNN architecture, 99% success was achieved on the CSE-CIC-IDS2018 dataset [18]. In the study where a two-level hybrid method was proposed, synthetic minority oversampling technique was used. CNN+RF algorithm showed the best performance with a success rate of 98% [19]. In the study where KSL-KDD dataset was used, filtering-based and correlation-based feature extraction was used to reduce the data size. Attribute selection was done according to the ranking procedure. In the study where the RF method was the most successful, an accuracy rate of 93.40% was achieved [20]. In the AdaBoost-based intrusion detection system, an experimental study was conducted by improving the imbalance of the training data in the study where synthetic minority oversampling technique and enseble feature selection were used. As a result of the study, an accuracy rate of 81.83 was obtained [21].

2. METHODS

Existing research has shown that the use of machine learning methods can significantly improve the effectiveness of detecting and preventing network attacks. With ever-evolving attack types and a wide variety of machine learning techniques, this field is not saturated for innovation and development. By adopting an anomaly detection approach to build a powerful and efficient intrusion detection system, this research provides insight for future researchers in this field. By utilising recursive feature elimination techniques and machine learning, this work aims to provide valuable guidance for future work in this area.

2.1. Data Set

The dataset used in this study is "CSE-CIC-IDS2018", a joint project between the Communications Security Establishment (CSE) and the Canadian Institute for Cyber Security (CIC) [22]. The CSE-CIC-IDS2018 dataset contains complex intrusion descriptions for low-level entities, applications or protocols. It combines profiles as a framework for building datasets covering generalised deployment models. These profiles can be used by human operators or agents to generate network events. The abstract nature of profiles allows them to be implemented in a range of network protocols with various topologies. By combining multiple profiles, a customised dataset can be created to meet specific requirements. Six different scenarios for attack are considered. These are BruteForce, DOS, DDOS+PortScan, Web, Infiltration and Botnet attacks.

Real-time information about network traffic is provided by the Amazon platform through the acquisition of AWS data [23]. This data is considered one of the most reliable sources for the evaluation of intrusion detection models based on network anomalies [24]. Divided into 10 categories, the data covers FTP-BruteForce, SSH-Bruteforce, Benign, Bot, DDOS attack-LOIC-UDP, DDOS attack-HOIC, DoS attacks-GoldenEye, DoS attacks-Slow HTTP test, SQL Injection and intrusion attacks. Table 1 shows the number of attacks in each category and their percentage of the original data volume. The attack infrastructure shown in Figure 1 is spread over 50 devices, while the victim organisation consists of 30 servers, 420 terminals and 5 partitions. Attacks were carried out periodically for 10 days. The data contains 80 attributes obtained with the CICFlowMeter-V3 tool. Table 2 gives an overview of some of the traffic characteristics [25].

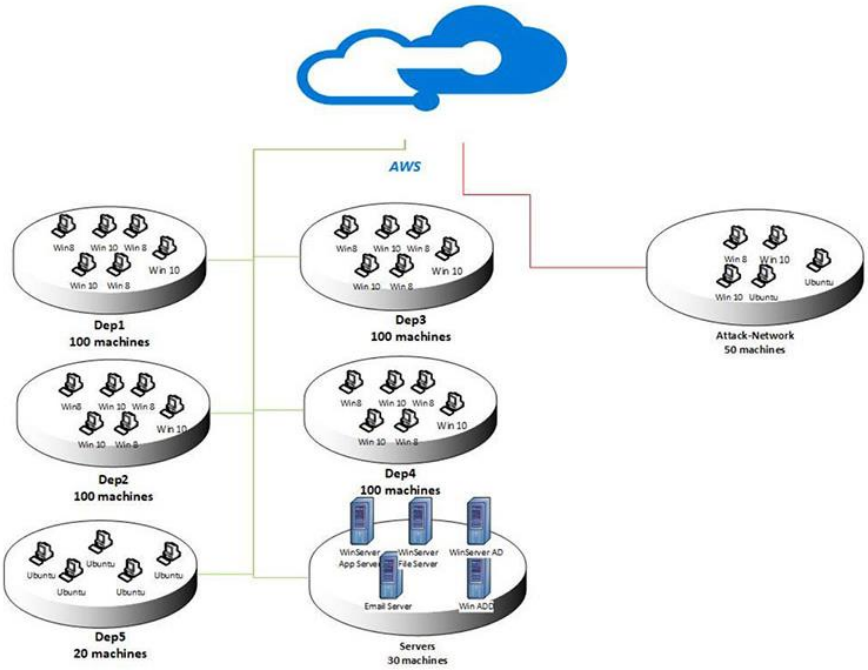

Figure 1. Network Topology [25].

Table 1. Volume of data points in attack class and ratio of it

Attack Class	Volume of data point in class (10574041)	Ratio from the original data (%)
Benign	8699178	80,101217
DDOS (HOIC)	686012	6,3167343
DoS (Hulk)	461912	4,2532425
BOT	286191	2,6352199
BruteForce (FTP)	193360	1,7804408
Bruteforce (SSH)	187589	1,727302
Infilteration	161934	1,4910731
DoS (SlowHTTP)	139890	1,288094
DoS (GoldenEye)	41508	0,3822018
DDOS (LOIC-UDP)	1730	0,0159297
Brute Force (Web)	611	0,005626
Brute Force (XSS)	230	0,0021178
SQL Injection	87	0,0008011

Table 2. Sample from CIC-IDS 2018 dataset features

Feature name	Description of feature
Down-up-ratio	Upload and download rate
Fw-win-byt	Number of bytes sent in forward direction
Fw-pkt-std	Standard deviation size
Fw-act-pkt	Transmission control protocol packet count
Fw-pkt-avg	Average size of packet
atv-max	Maximum time active before idle
Down-up-ratio	Download and upload ratio
Tot-bw-pk	Total number of packages
Tot-fw-pk	Total packages
Pkt-len-var	Package inter-arrival time
Bw-pkt-max	Max size of packet
Bw-pkt-min	Min size of packet
Bw-hdr-len	Total bytes used

2.2.Preprocessing on Dataset

The initial dataset consisted of 80 features and there appeared to be few features with minimal impact. It is very important to interpret the data and traffic behaviour carefully to determine if it is normal. However, some features such as timestamp and IP addresses hinder the model's ability to detect errors and intrusions. As a result, 78 useful features were selected from the original set to train the model. Feature names that were redundant and appeared in more than one row in the datasets were removed. Finally, to ensure the accuracy of the analysis, the dataset is split into two parts, 80% training set and 20% test set.

2.3.Decision Tree

Decision trees (DT) are a particularly effective algorithm for supervised learning tasks such as classification and regression. It is considered non-parametric and has a hierarchical structure

consisting of a root node, internal nodes, branches and leaf nodes [26]. The root node without incoming branches is the first step of this algorithm. The decision nodes or internal nodes receive inputs from the branches originating from the root node. Depending on the available features, these nodes, together with terminal or leaf nodes, form homogeneous subsets. The leaf nodes contain all possible outcomes for a given data set [3]. The structure of the decision tree provides a clear and concise description of the decision-making process. Its simplicity is designed to help various groups within an organisation understand the rationale behind a decision. The divide-and-conquer method used in decision tree learning uses a greedy search to identify the best split points in a tree.

2.4.Random Forest

Random Forest (RF) is an algorithm for ensemble learning based on DT. This type of machine learning algorithm combines multiple DT to improve the accuracy and robustness of the output. Supervised learning is a method that involves building DT and prediction models through the learning process. This technique is used to build multiple models. When creating a decision forest, the decision tree is used as the basis. Each node in a RF contains a random element [27]. The algorithm splits the data using the most superior among a carefully selected subset of predictions. This particular method is used to overcome difficulties in both classification and regression. By dividing the data set into smaller subsets, this process creates various DT with multiple branches to analyse the given task [28]. In the regression domain, the results obtained from an input are compatible with the outputs produced by DT.

2.5.Extra Trees

The procedure of the Extra Trees (ET) algorithm involves the construction of a large number of DT that are not pruned using the training data set. In the case of regression, predictions are obtained by averaging the predictions of the DT, whereas in classification, majority voting is used [29]. RF classifier is also utilised in this process. The model follows a different version, similar to the RF approach, where copies of the dataset are used to train it. However, instead of using specific decision criteria to separate the data in the branching stages, the method randomly selects the criteria to be followed. The technique of using branching paths to solve data analysis problems is known to simplify the process and reduce complexity. However, the efficiency of this method is reduced when it comes to processing larger datasets with a high degree of noise. In such cases, statistical analysis is recommended. This approach usually leads to increased bias and reduced volatility [13].

2.6.Extreme Gradient Boosting

The Extreme Gradient Boosting (XGBoost) algorithm is an optimisation of the Gradient Boosting algorithm, aiming to help prevent overlearning. In addition to its ability to process empty data easily and quickly, the most important aspect of this algorithm is its exceptional predictive power [30]. Unlike other well-known machine learning techniques and algorithms, it is quite different. It can be used for both regression and classification problems. The computation involved in this method is significantly simpler than commonly used machine learning techniques. When building a tree, XGBoost applies the maximum depth value. If the generated tree is moving downwards excessively, it is pruned to prevent overlearning. Thus, over-prevention is prevented [31]. The loss function in the Gradient Boosting algorithm is calculated using a first-order function. XGBoost performs its calculations using second-order functions and parallel processing. This feature provides faster results in a shorter time compared

to other algorithms. Multidimensional data analysis can be easily done using this tool. Using the XGBoost method, many very large and multidimensional data analyses such as click-through rates to advertisements, malware, patient and disease prediction, price analysis, customer satisfaction prediction can be performed.

2.7.Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) is a process used to oversample data by creating synthetic data. This method is widely used in data science projects. The main purpose of SMOTE is to create new instances of the minority class by performing certain operations between instances of the same class [32]. The number of neighbours from k nearest neighbours is randomly selected depending on the amount of oversampling desired. This technique eliminates the overfitting problem and provides good classification performance. Unlike random sampling methods, SMOTE does not simply copy the minority class data, but instead generates artificial samples based on the k nearest neighbours of the analysed samples [33]. Synthetic samples are generated as follows:

- To determine the discrepancy between the analysed feature vector (Si) and its closest counterpart, the two are compared.
- To obtain the final result, the discrepancy is magnified by a chance variable (t) ranging from 0 to 1.
- After analysing the feature vector, the data obtained are included in it and then a new sample is created.

Using formula 1, SMOTE can be calculated as;

$$Snew=Si+(Sx-Si)*t \quad (1)$$

2.8.Hyperparameter Optimisation of Decision Tree

Optimisation of hyperparameters for DT is a procedure that attempts to determine the optimal values that will improve the performance of the decision tree in question [31]. Hyperparameter tuning is also challenging as there is no direct way of how a change in the hyperparameter value can computationally reduce the loss of your model. For this reason, we experiment to find the best result. These experiments start with a set of possible values for all hyperparameters. Let's come to the main question and where many people get stuck. To answer this question, we first need to know what these hyperparameters mean. We need to understand how changing a hyperparameter will affect your model architecture. After defining the range of values, the next step is to use a hyperparameter tuning method. The most common and expensive is Grid Search, while others such as Random Search and Bayesian Optimisation will provide a "smarter", less expensive tuning.

3. EXPERIMENTAL

XGBoost, ET, DT and RF algorithms were used in the experimental study. For data with low data class, the number of data in the class was increased by generating synthetic data with SMOTE technique. Cross validation was performed 5 times in XGBoost, ET, DT and RF algorithms. By using the SMOTE technique, the small number of data was increased and data imbalance was prevented. The data set consists of 10 parts. These parts consist of attack attacks created for each day. In our study, our 10-day data set was combined. Table 3 shows the

accuracy percentages of the algorithms in the experimental study. In the combined data set, the accuracy rates are generally close to each other and the most successful algorithm is XGBoost algorithm.

Table 3. Comparison of the accuracy rates of the experimental study

	XGBoost	ET	DT	RF
Study 1	98.02	97.41	96.71	97.69
Study 2	98.18	98.05	97.03	98.09

Our study consists of 2 stages. Table 3 shows the accuracy rates of the studies. The first phase is named as Study 1 and the second phase is named as Study 2. The 10-day combined data set was subjected to classification with 4 different algorithms. In Study 2, unlike Study 1, extra hyperparameter optimisation was added. The most successful algorithm was XGBoost algorithm with 98.02% accuracy rate in Study 1. In Study 2, the decision tree hyperparameter optimisation on the same data set improved the success rate and the XGBoost algorithm outperformed the other algorithms with an accuracy rate of 98.18%. Table 4 and Table 5 show the performance metrics of the study.

Table 4. Performance metrics for Study 1

	XGBoost	ET	DT	RF
Accuracy	98.02	97.41	96.71	97.69
Precision	97.89	97.58	97.92	98.01
Recall	98.02	97.41	96.71	97.69
F1-score	97.88	97.55	96.88	97.82

Table 5. Performance metrics for Study 2 (Decision tree Hyperparameter Optimization)

	XGBoost	ET	DT	RF
Accuracy	98.18	98.05	97.03	98.09
Precision	98.10	98.06	97.90	98.01
Recall	98.14	98.05	97.03	98.09
F1-score	97.92	97.90	97.86	97.93

The comparison of the literature studies and our study is given in Table 6. The table includes UNR-IDD, NSL-KDD, KDD CUP99, UNSW-NB15, CSE-CIC-IDS-2017 and CSE-CIC-IDS-2018 data sets used in this study.

Table 6. Similar studies in the literature

Works in Progress	Used Data Set	Method Used	Success Rate (%)
T. Das vd., 2023 [4]	UNR-IDD	RF	96.00
M. A. Shyaa vd., 2023[5]	CSE-CIC-IDS-2018	GPC-FOS	97.22
M. Safa Bıçakcı And S. Toklu, 2023 [6]	NSL-KDD	SAE-4-SKB-RF	99.67
P. Dini vd., 2022 [7]	CSE-CIC-IDS-2018	SVM	94.50
B. Ekici And H. Takcı, 2022 [8]	CSE-CIC-IDS-2017	RF	94.00
J. Yoo vd., 2021 [9]	CSE-CIC-IDS-2018	MLP and LSTM	95.00
S. Seth vd., 2021 [10]	CSE-CIC-IDS-2018	LSTM +AM	96,97
A. Jumabek vd., 2021 [11]	CSE-CIC-IDS-2018	CatBoost	92.41
S. Emanet vd., 2021 [12]	CSE-CIC-IDS2018	ET	98.79
B. Çakır And P. Angın vd., 2021 [13]	KDD CUP99	LSTM	97.00
Q. R. S. Fitni and K. Ramli 2020 [3]	CSE-CIC-IDS2018	RF	98.80
M. Catillo vd., 2020 [14]	CSE-CIC-IDS2018	two-level approach	98.25
G. C. Amaizu vd., 2020 [15]	NSL-KDD	DNN	97.89
G. C. Amaizu vd., 2020 [15]	UNSW-NB15	DNN	89.99
G. C. Amaizu vd., 2020 [15]	CSE-CIC-IDS2018	DNN	76.47
J. Kim vd., 2020 [16]	CSE-CIC-IDS2018	CNN	91.50
J. Kim vd., 2020 [16]	KDD CUP99	CNN and RNN	99.00
M. A. Ferrag vd., 2019 [17]	CSE-CIC-IDS2018	DBN (Deep belief networks)	98.00
B. I. Farhan and A. D. Jasim, 2019 [18]	CSE-CIC-IDS2018	LSTM	99.00
İ. Seviyeli d., 2019 [19]	CSE-CIC-IDS2018	CNN+RF	98.00
Ö. Emhan and M. Akın, vd., 2019 [20]	NSL-KDD	RF	95.60
M. Blanchard, vd, 2019 [21]	CSE-CIC-IDS-2017	AdaBoost	81.83
Study conducted 1	**CSE-CIC-IDS2018**	**XGBoost**	**98.02**
Study conducted 2	**CSE-CIC-IDS2018**	**XGBoost**	**98.18**

In the study conducted by S. Emanet et al. with the CSE-CIC-IDS2018 dataset, 40 features were used and a success rate of 98.76% was obtained [13]. Similarly, Q. R. S. Fitni and K. Ramli obtained 98.80% accuracy rate by using 23 features in their study [3]. The fact that these rates

are higher than this study is due to the low number of features. In general, our study on the CSE-CIC-IDS2018 dataset with 78 attributes has a higher success rate than other studies.

4. CONCLUSION

As a result of this research, it has been shown that intrusion detection can be performed efficiently with the help of machine learning based classifiers. In this study, four algorithms, namely XGBoost, ET, DT and RF, were compared. The CSE-CIC-IDS2018 dataset was used for both training and testing. In previous studies with this dataset, attack types were usually evaluated separately. In this study, all attack types and dataset were combined to detect different attack types in the same model. The study has shown that the use of machine learning techniques produces a remarkable level of success in detecting attacks. As a result of the experimental study, the XGBoost algorithm was the most successful method with an accuracy rate of 98.18% in preventing cyber-attacks on the data set used. XGBoost algorithm is a high performance algorithm in certain scenarios. In the study, the unbalanced data set was balanced with the SMOTE technique. In addition, hypermetre optimisation of the decision tree was also used and the success rate was increased with these two methods. Other algorithms were also successful at a close rate. In future studies, it is planned to analyse artificial neural networks and deep learning algorithms with this combined dataset.

REFERENCES

[1] M. Salih Karaman, M. Turan, and M. Ali Aydın, (2021), 'Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması', Avrupa Bilim ve Teknol. Derg., no. Ejosat Ek Özel Sayı (HORA), pp. 10–17 doi: 10.31590/EJOSAT.1115825.

[2] M. Baykara and R. Daş, (2019), 'Saldırı tespit ve engelleme araçlarının incelenmesi', Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Derg., vol. 10, no. 1, pp. 57–75 doi: 10.24012/DUMF.449059.

[3] Q. R. S. Fitni and K. Ramli, (2020), 'Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems', Proc. - 2020 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol. IAICT 2020, pp. 118–124 doi: 10.1109/IAICT50021.2020.9172014.

[4] T. Das, O. A. Hamdan, R. M. Shukla, S. Sengupta, and E. Arslan, (2023), 'UNR-IDD: Intrusion Detection Dataset using Network Port Statistics', pp. 497–500 doi: 10.1109/CCNC51644.2023.10059640.

[5] M. A. Shyaa, Z. Zainol, R. Abdullah, M. Anbar, L. Alzubaidi, and J. Santamaría, (2023), 'Enhanced Intrusion Detection with Data Stream Classification and Concept Drift Guided by the Incremental Learning Genetic Programming Combiner', Sensors (Basel)., vol. 23, no. 7, p. 3736 doi: 10.3390/s23073736.

[6] M. S. Bıçakcı and S. Toklu, (2022) , ' Bilgisayar Ağı Güvenliği için Hibrit Öznitelik Azaltma ile Makine Öğrenmesine Dayalı Bir Saldırı Tespit Sistemi Tasarımı' Accessed: Apr. 26, 2023. [Online]. Available: http://dergipark.gov.tr/gbad

[7] P. Dini et al., (2022), 'Design and Testing Novel One-Class Classifier Based on Polynomial Interpolation with Application to Networking Security', IEEE Access, vol. 10, pp. 67910–67924 doi: 10.1109/ACCESS.2022.3186026.

[8] B. Ekici and H. Takcı, (2022), 'Bilgisayar Ağlarında Anomali Tespiti Yaklaşımı ile Saldırı Tespiti', Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilim. Derg., vol. 22, no. 5, pp. 1016–1027 doi: 10.35414/AKUFEMUBID.1114906.

[9] J. Yoo, B. Min, S. Kim, D. Shin, and D. Shin, (2021), 'Study on Network Intrusion Detection Method Using Discrete Pre-Processing Method and Convolution Neural Network', IEEE Access, vol. 9, pp. 142348–142361 doi: 10.1109/ACCESS.2021.3120839.

[10] S. Seth, K. K. Chahal, and G. Singh, (2021), 'A Novel Ensemble Framework for an Intelligent Intrusion Detection System', IEEE Access, vol. 9, pp. 138451–138467 doi: 10.1109/ACCESS.2021.3116219.

[11] A. Jumabek, S. Yang, and Y. Noh, (2021), 'CatBoost-Based Network Intrusion Detection on Imbalanced CIC-IDS-2018 Dataset', vol. 46, no. 12, pp. 2191–2197 doi: 10.7840/KICS.2021.46.12.2191.

[12] B. Çakır and P. Angın, (2021), 'Zamansal Evrişimli Ağlarla Saldırı Tespiti: Karşılaştırmalı Bir Analiz', Eur. J. Sci. Technol., vol. 22, no. 22, pp. 204–211 doi: 10.31590/ejosat.848784.

[13] S. Emanet, G. Karatas Baydogmus, O. Demir, (2021), 'Effects of Feature Selection Methods on Machine Learning Based Intrusion Detection System Performance', DUJE (Dicle Univ. J. Eng., vol. 12, pp. 743–755 doi: 10.24012/dumf.1051340.

[14] M. Catillo, M. Rak, and U. Villano, (2020), '2L-ZED-IDS: A Two-Level Anomaly Detector for Multiple Attack Classes', Adv. Intell. Syst. Comput., vol. 1150 AISC, pp. 687–696 doi: 10.1007/978-3-030-44038-1_63/TABLES/3.

[15] G. C. Amaizu, C. I. Nwakanma, J. M. Lee, and D. S. Kim, (2020), 'Investigating Network Intrusion Detection Datasets Using Machine Learning', Int. Conf. ICT Converg., vol. 2020-October, pp. 1325–1328 doi: 10.1109/ICTC49870.2020.9289329.

[16] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, (2020), 'CNN-Based Network Intrusion Detection against Denial-of-Service Attacks', Electron. 2020, Vol. 9, Page 916, vol. 9, no. 6, p. 916 doi: 10.3390/ELECTRONICS9060916.

[17] M. A. Ferrag, L. A. Maglaras, H. Janicke, and R. Smith, (2019), 'Deep Learning Techniques for Cyber Security Intrusion Detection : A Detailed Analysis' doi: 10.14236/EWIC/ICSCSR19.16.

[18] B. I. Farhan and A. D. Jasim, (2022), 'Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset', Indones. J. Electr. Eng. Comput. Sci., vol. 26, no. 2, pp. 1165–1172 doi: 10.11591/ijeecs.v26.i2.pp1165-1172.

[19] İ. Seviyeli et al., (2019), 'İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti', Gazi Mühendislik Bilim. Derg., vol. 5, no. 3, pp. 258–272 doi: 10.30855/GMBD.2019.03.07.

[20] Ö. Emhan and M. Akın, (2019), 'Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi', DÜMF Mühendislik Derg., vol. 10, no. 2, pp. 549–559 doi: 10.24012/dumf.565842.

[21] M. Blanchard et al., (2019), 'Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset', J. Phys. Conf. Ser., vol. 1192, no. 1, p. 012018 doi: 10.1088/1742-6596/1192/1/012018.

[22] E. Kharısmadhany, (2022), 'IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018)', Kaggle. Accessed: May 10, 2023. [Online]. Available: https://www.kaggle.com/code/ekkykharismadhany/dataset-checking/data%0Akaggle kernels output ekkykharismadhany/dataset-checking -p /path/to/dest

[23] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, (2020), 'Building an efficient intrusion detection system based on feature selection and ensemble classifier', Comput. Networks, vol. 174, p. 107247 doi: 10.1016/j.comnet.2020.107247.

[24] R. I. Farhan, A. T. Maolood, and N. F. Hassan, (2020) , 'Optimized Deep Learning with Binary PSO for Intrusion Detection on CSE-CIC-IDS2018 Dataset', J. Al-Qadisiyah Comput. Sci. Math., vol. 12, no. 3, p. 16 doi: 10.29304/jqcm.2020.12.3.706.

[25] 'IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB'. Accessed: Apr. 28, 2023. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[26] I. F. Kilincer, F. Ertam, and A. Sengur, (2021), 'Machine learning methods for cyber security intrusion detection: Datasets and comparative study', Comput. Networks, vol. 188, p. 107840 doi: 10.1016/j.comnet.2021.107840.

[27] O. Sevli, (2019), 'Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması', Eur. J. Sci. Technol., no. 16, pp. 176–185 doi: 10.31590/ejosat.553549.

[28] M. B. Keles, A. Keles, A. Keles, (2020) , 'Yapay Zekâ Teknolojisi ile Uçuş Fiyatı Tahmin Modeli Geliştirme' doi: 10.29228/TurkishStudies.45993.

[29] E. Efeoğlu, (2022), 'Kablosuz Sinyal Gücünü Kullanarak İç Mekan Kullanıcı Lokalizasyonu için Karar Ağacı Algoritmalarının Karşılaştırılması', Acta Infologica, vol. 6, no. 2, pp. 163–173 doi: 10.26650/ACIN.1076352.

[30] T. Oluwatosin Omotehinwa and D. Opeoluwa Oyewola, (2023), 'Hyperparameter Optimization of Ensemble Models for Spam Email Detection', Appl. Sci. 2023, Vol. 13, Page 1971, vol. 13, no. 3, p. 1971 doi: 10.3390/APP13031971.

[31] M. A. Çakıroğlu, G. İnce, H. T. Kabas, and A. A. Süzen, (2021), 'Experimental Examination of the Behavior of Shotcrete-Reinforced Masonry Walls and Xgboost Neural Network Prediction Model', Arab. J. Sci. Eng., vol. 46, no. 11, pp. 10613–10630 doi: 10.1007/S13369-021-05466-1/TABLES/6.

[32] O. Sevli, (2022), 'Farklı Sınıflandırıcılar ve Yeniden Örnekleme Teknikleri Kullanılarak Kalp Hastalığı Teşhisine Yönelik Karşılaştırmalı Bir Çalışma', J. Intell. Syst. Theory Appl., vol. 5, no. 2, pp. 92–105 doi: 10.38016/JISTA.1069541.

[33] M. Yavaş, A. Güran, and M. Uysal, (2020), 'Covid-19 Veri Kümesinin SMOTE Tabanlı Örnekleme Yöntemi Uygulanarak Sınıflandırılması', Avrupa Bilim ve Teknol. Derg., pp. 258–264 doi: 10.31590/EJOSAT.779952.