Acta Scientific COMPUTER SCIENCES

Volume 5 Issue 11 November 2023

Research Article

Cybersecurity in Machine Learning Techniques: Detecting Network Attacks

Saif Rawashdeh*

Department of Computer Science, Jordan University of Science and Technology, Jordan

***Corresponding Author:** Saif Rawashdeh, Department of Computer Science, Jordan University of Science and Technology, Jordan.

Received: October 13, 2023

Published: October 21, 2023

Abstract

Using the well-known dataset HTTP DATASET CSIC 2010, this work intends to build seven machine learning methods (Decision Tree, Random Forest, Gradient Boosting, XGBoost, AdaBoost, Multilayer Perceptron, and Voting) to identify anomaly assaults. Accuracy, precision, recall, and f1-score are four common evaluation metrics used to rate the effectiveness of these models. In order to identify several attack methods on this dataset, we conducted one experiment: Binary Classification into two categories (normal and malicious attacks). The findings demonstrated that in this experiment, the voting classifier and decision tree provided the greatest performance outcomes.

Keywords: HTTP DATASET CSIC 2010; Machine Learning; Cybersecurity Attacks; Detection Attacks

Introduction

Cybersecurity is the safeguarding of computer systems and networks against information leakage, data theft, damage to their electronic data, software, hardware, or other components, as well as disruption or misdirection of the services they offer [1]. The Internet of Things (IoT) has become more significant as a result of growing reliance on computer systems, the Internet, wireless network standards like Bluetooth and Wi-Fi, as well as the spread of “smart” devices like smartphones, televisions, and the myriad devices that make up the Internet of things [1]. Because to its complexity in terms of both political usage and technology, cybersecurity is one of the most urgent problems in the modern world. The system’s main objectives are dependability, integrity, and data security [2].

Any harmful activity that targets IT systems or the users of those systems in an effort to gain unauthorized access to the systems and the data or information they contain is referred to as a cybersecurity attack [3]. Cyberattackers are typically crooks looking to profit financially from the attack. In other instances, the goal is to disrupt operations by restricting access to IT systems or obliterating actual physical equipment [3]. State actors or cybercriminals working for them are frequently involved in state-sponsored cybercrime. Attacks on cybersecurity can be narrowly focused, affecting a number of businesses spread over numerous regions and nations, or they can be broad in scope and target specific companies or persons [4]. Targeted attacks frequently spread beyond their original targets, endangering all businesses. The NotPetya infection that swept the world in June 2017 was most likely brought on by a state-sponsored strike against Ukrainian banks and utilities. The clean-up had the desired effect on Ukraine, but it also had a global impact, costing almost $10 billion in IT system recovery and lost productivity, according to publications documenting the clean-up [2-4].

The study of machine learning (ML) focuses on comprehending and developing “learning” techniques, or techniques that employ data to enhance performance on a variety of tasks. Artificial intelligence is related to it [5]. Without the need for explicit programming, machine learning algorithms create a model from training data to produce predictions or judgments. When it is difficult or impossible to create traditional algorithms, machine learning algorithms are employed in a range of fields, including medicine, email filtering, speech recognition, and computer vision [5].

Cybersecurity systems can employ machine learning to analyze patterns and learn from them to help prevent repeat assaults and adapt to changing behavior. It can aid cybersecurity teams in risk reduction and quicker attack response [5]. By cutting down on time spent on regular tasks, it can help firms utilize their resources more strategically. In summary, machine learning has the ability to simplify, be proactive, be more cost-effective, and be more successful in cybersecurity [4,5]. It can only accomplish this, though, if the data that powers machine learning gives a complete picture of the environment. Garbage in, garbage out, as the saying goes [5].

In order to determine whether each sample is normal or anomalous and then to differentiate between the anomalous attacks in this dataset, many machine learning methods are applied to a well-known network intrusion dataset in this research. The rest of this essay is structured as follows: Some earlier works that are relevant to this study are presented in Section 2. The research approach is described in Section 3 of this publication. The findings are illustrated in Section 4. The paper's conclusion is provided in Section 5, which also makes some recommendations for additional research.

Related Work

In the subject of cybersecurity, a number of writers used machine learning algorithms to identify various sorts of assaults using real-time information or pre-existing datasets from a number of sources as shown in Table 1.

Kim., *et al*. [6] used two deep learning algorithms to classify the sample as normal or anomalous: Convolutional Neural Network (CNN) with long short-term memory (LSTM) and Deep Neural Network (DNN). They used the HTTP DATASET CSIC 2010 that contains 61,065 instances and 16 features divided into two categories: normal (36,000 samples) and anomalous (25,065 samples). The results showed that the CNN with LSTM achieved an accuracy of 91.54%. Vartouni., *et al*. [7] used a new algorithm to classify the sample if it was normal or anomalous, called a Stacked Auto-Encoder. They used the HTTP DATASET CSIC 2010 that contains 61,065 instances and 16 features divided into two categories: normal (36,000 samples) and anomalous (25,065 samples). The results showed that the Stacked Auto-Encoder achieved an accuracy of 88.32%.

Betarte., *et al*. [8] classified the sample as normal or abnormal using three machine learning algorithms: Random Forest, K-Nearest Neighbors (K-NN) with a k value of = 3, and Support Vector Machine (SVM). They used the HTTP DATASET CSIC 2010 dataset that contains 61,065 instances and 16 features divided into two categories: normal (36,000 samples) and anomalous (25,065 samples). The results showed that the Random Forest achieved a higher accuracy of 91.54% when compared with the others. Tuan., *et al*. [9] used five machine learning algorithms to detect different types in well-known cybersecurity dataset. They used a popular dataset called UNSW-NB15 that contains several kinds of network attacks. This dataset contains nine attack samples (DoS, Reconnaissance, Backdoor, Fuzzers, Analysis, Exploits, Worms, Shellcode and Generic) and normal attacks with 44 features. The machine learning algorithms are SVM, ANN, Naïve Bayes (NB), DT, and Unsupervised Learning (USML). They showed that the USML obtained good performance in this dataset an accuracy of 94.78%.

Anwer., *et al*. [10] used four machine learning techniques to detect malicious network traffic based on a well-known dataset. They used a popular cybersecurity dataset named NSL-KDD that has 148,517 samples divided into training (125,973) and testing (22,544) datasets. This dataset contains five classes divided into two categories (normal and non-normal attacks) and each of non-normal attacks has several types. R2L (Xsnoop, Guess_Password, Named, Ftp_write, Phf, Multihop, Imap, Warezmaster, Warezclient, Snmpgetattack, Spy, Xlock, Snmpguess, Httptunnel, Sendmail), DoS (Smurf, Back, Land, Processtable, Neptune, Pod, Apache2, Udpstorm, Worm, Teardrop), U2R (Xterm, Buffer_overflow, Sqlattack, Rootkit, Perl, Loadmodule, Ps), and Probe (Portsweep, Satan, Nmap, Ipsweep, Mscan, Saint) are non-normal attacks. Support Vector Machine (SVM), Gradient Boosted Decision Trees (GBDT), and Random Forest are the machine learning algorithms (RF). They evaluated these algorithms using four metrics: accuracy, specificity, training time, and prediction time. In comparison to the other algorithms, the accuracy of RF equals to 85.34% provided the best performance.

Su., *et al*. [11] showed the BAT model, which is a deep learning method for finding network threats that are trying to get in. They used the NSL-KDD dataset, which is well-known in the field of network attack. This dataset contains 148,517 samples divided into training (125,973) and testing (22,544) datasets. This dataset contains five classes divided into two categories (normal and non-normal attacks), and each of the non-normal attacks has several types. R2L (Xsnoop, Guess_Password, Named, Ftp_write, Phf, Multihop, Imap, Warezmaster, Warezclient, Snmpgetattack, Spy, Xlock, Snmpguess, Httptunnel, Sendmail), DoS (Smurf, Back, Land, Processtable, Neptune, Pod, Apache2, Udpstorm, Worm, Teardrop), U2R (Xterm, Buffer overflow, Sqlattack, Rootkit, Perl, Loadmodule, Ps), and Probe (Portsweep, Satan, Nmap, and MSnamp are non-normal attacks. In the intrusion detection procedure, this model gave an accuracy value of 84.25%.

Xu., *et al*. [12] came up with a new 5-layer autoencoder (AE)-based model that is better at finding unusual things in a network. They used the NSL KDD dataset, which is well-known in the field of network attacks. This dataset contains 148,517 samples divided into training (125,973) and testing (22,544) datasets. This dataset contains five classes divided into two categories (normal and non-normal attacks), and each of the non-normal attacks has several types. R2L (Xsnoop, Guess_Password, Named, Ftp_write, Phf, Multihop, Imap, Warezmaster, Warezclient, Snmpgetattack, Spy, Xlock, Snmpguess, Httptunnel, Sendmail), DoS (Smurf, Back, Land, Processtable, Neptune, Pod, Apache2, Udpstorm, Worm, Teardrop), U2R (Xterm, Buffer overflow, Sqlattack, Rootkit, Perl, Loadmodule, Ps), and Probe (Portsweep, Satan, Nmap, and Mscan, Saint) are non-normal attacks. They have shown that this model gave 90.61% as an accuracy value in the intrusion detection process.

Kavitha., *et al*. [13] came up with a new algorithm called One Class Support Vector Machine (OCSVM) to find network attacks that break in.They used the NSL-KDD dataset, which is well-known in the field of network attacks and contains 148,517 samples divided into training (125,973) and testing (22,544) datasets. This dataset contains five classes divided into two categories (normal and non-normal attacks), and each of the non-normal attacks has several types. R2L (Xsnoop, Guess_Password, Named, Ftp_write, Phf, Multihop, Imap, Warezmaster, Warezclient, Snmpgetattack, Spy, Xlock, Snmpguess, Httptunnel, Sendmail), DoS (Smurf, Back, Land, Processtable, Neptune, Pod, Apache2, Udpstorm, Worm, Teardrop), U2R (Xterm, Buffer overflow, Sqlattack, Rootkit, Perl, Loadmodule, Ps), and Probe (Portsweep, Satan, Nmap, and Mscan, Saint) are non-normal attacks. They have shown that this model gave an accuracy value of 81.29% in the intrusion detection procedure.

Ferriyan., *et al*. [14] built a new cybersecurity dataset named ALLFLOWMETER_HIKARI2021 and used several machine learning models to detect the different types of attacks. This dataset contains 86 features extracted by Zeek [https://zeek.org/] and 555,278 instances divided into six categories of attacks: background, benign, bruteforce, bruteforce-XML, probing, and XMRIGCC CryptoMiner. Background and Benign belong to normal attacks, while Bruteforce, Bruteforce-XML, probing, and XMRIGCC CryptoMiner belong to malicious attacks. The machine learning models are KNN, SVM, RF, and MultiLayer Perceptron (MLP). They have shown that these models achieve the same accuracy in the detection process as 0.99%.

Ref	Year	Attack	Cybersecurity Dataset	Algorithm	Result
[6]	2020	Normal Anomalous	CSIC 2010 dataset	CNN and LSTM DNN	CNN and LSTM accuracy = 91.54%
[7]	2018	Normal Anomalous	CSIC 2010 dataset	Stacked Auto-Encoder	Accuracy = 88.32
[8]	2018	Normal Anomalous	CSIC 2010 dataset	Random Forest KNN-3 SVM	Random Forest accuracy = 72%
[9]	2019	• Analysis Reconnaissance • DoS • Exploits • Fuzzers • Generic • Normal • Worms • Backdoor • Shellcode	UNSW-NB15	SVM, ANN, NB, DT, and USML	USML accuracy = 94.78%.
[10]	2021	• R2L • DoS • U2R • Probe • Normal	NSL-KDD	SVM, GBDT, and RF	RF Accuracy = 85.34%
[11]	2020	• R2L • DoS • U2R • Probe • Normal	NSL-KDD	BAT model	Accuracy = 84.25%
[12]	2021	• R2L • DoS • U2R • Probe • Normal	NSL-KDD	5-layer autoencoder (AE)-based model	Accuracy = 90.61%
[13]	2021	• R2L • DoS • U2R • Probe • Normal	NSL-KDD	One Class SVM	Accuracy = 81.29%
[14]	2021	• Background • Benign • Bruteforce • Bruteforce-XML • Probing • XMRIGCC CryptoMiner	ALLFLOWMETER HIKARI2021	KNN, SVM, RF, and MLP	Accuracy = 0.99

Table 1: Previous Related Work for Cybersecurity Attacks Detection.

Methodology

Figure 1 shows how the proposed method for this study was used to find different kinds of attacks in different cybersecurity datasets. The following sections illustrate the proposed methodology.

Dataset description

We used in our experiment a well-known dataset and contains many cybersecurity attacks: HTTP DATASET CSIC 2010[1] dataset. Table 2 shows the name of each dataset, number of Instances, number of features, and what are the cybersecurity attacks.

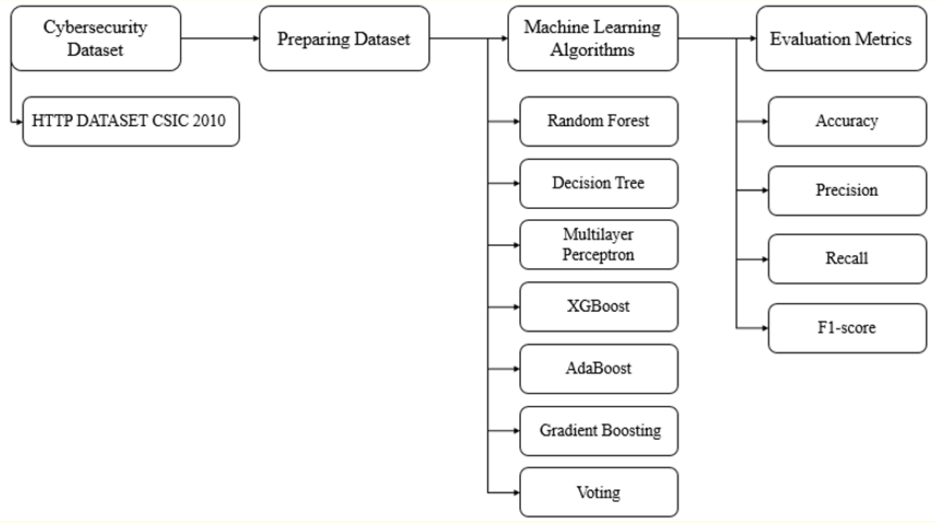

Figure 1: Flow-chart of Proposed Methodology.

Table 2: Information of Cybersecurity Datasets.

Dataset Name	No. of Instances	No. of Features	Cybersecurity Attacks	
HTTP DATASET CSIC 2010	61,065	16	Normal	36,000
			Malicious	25,065

The HTTP dataset CSIC 2010 is generated traffic to an e-commerce web application built at our department. Users can purchase things using a shopping cart and register by giving personal information in this web application. The data collection contains some Latin characters because it is a web application in Spanish [15]. The dataset was generated automatically and contains over 25,000 malicious queries in addition to 36,000 normal requests. SQL injection, buffer overflow, information gathering, file disclosure, CRLF injection, XSS, server-side include, parameter tampering, and more threats are included in the dataset. In prior studies, this dataset was effectively employed for web detection [15]. Table 3 shows the features in this dataset, and Figure 2 presents the attack frequency in the HTTP dataset CSIC 2010.

Preparing dataset

To apply different types of machine learning algorithms to these datasets, we convert the non-numerical features in each dataset to numerical features using a popular encoding technique [16]. This method is known as "Label Encoder," and it turns non-numerical data into machine-readable forms by replacing each value with a unique number starting at 0 [16]. In this dataset, all the features are categorical features and we must convert it to numerical features.

Machine learning algorithms

After the dataset is prepared, it is fitted to seven machine learning algorithms to detect the aforementioned cybersecurity attacks in each dataset. We used a hold-out technique to split the dataset into training and testing datasets with a test size of 0.1, which means that 0.9 of the whole datasets is a training dataset and 0.1 of the whole datasets is a testing dataset. The training dataset is used to build several machine learning models, while the testing dataset is used to assess the performance of these models.

Random forest algorithm (RF)

Random forest is an ensemble supervised learning method for regression and classification tasks in which a large number of decision trees are trained. For classification tasks, the prediction result is the class picked by the most trees. For regression tasks, the mean or average forecast of each tree is returned [17]. Decision trees have a tendency to overfit their training set, which is corrected by random decision forests. As a result, the Random Forest classifier

[1] ALLFLOWMETER_HIKARI: https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks

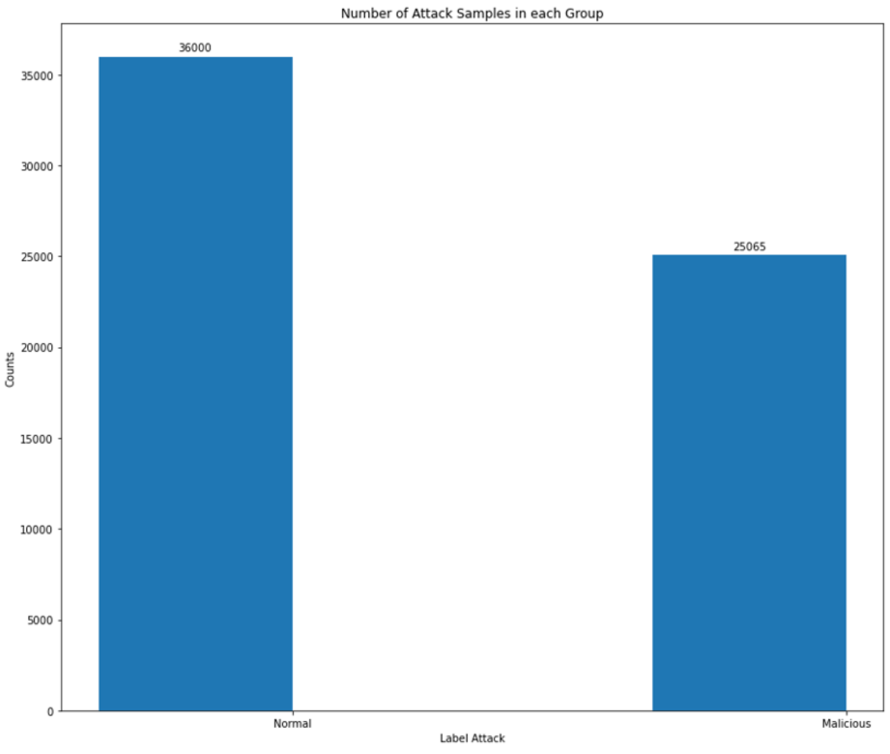

Figure 2: Attacks Frequency in HTTP DATASET CSIC 2010 Dataset.

No.	Feature
1	Method
2	User-Agent
3	Pragma
4	Cache-Control
5	Accept
6	Accept-encoding
7	Accept-charset
8	language
9	host
10	cookie
11	content-type
12	connection
13	length
14	content
15	URL
16	label

Table 3: HTTP DATASET CSIC 2010 Features.

is used to classify each cybersecurity dataset, which incorporates several attacks [18,19]. In our experiment, we used the random forest with classification type because the label is discrete and the parameters of RF that used are as the following: n_estimators (number of decision trees) = 100, max_features = sqrt, max_depth = None, random_state = 42.

Decision tree algorithm (DT)

Decision Tree is a supervised machine learning algorithm that makes judgments based on a set of rules, similar to how people do [20]. Decision tree learning, also known as induction of decision trees, is one of the predictive modeling approaches used in three fields: data mining, statistics, and machine learning [21]. It goes from observations about a sample (represented in the branches) to inferences about the sample's target value using a decision tree (represented in the leaves that are attack type) [22].

Classification trees are tree models with a discrete target variable; in these tree structures, leaves represent class labels (types of attacks) and branches represent features in dataset that lead to predict the class labels [23,24]. Regression trees are decision trees with a continuous target variable (typically real numbers) [23]. Decision trees are one of the well-known machine learning algorithms due to their comprehensibility and simplicity. In our experiment, we used the Decision Tree with classification type because the label is discrete and the parameters of DT that used are as the following: criterion = gini and random_state = 42.

Multilayer perceptron algorithm (MLP)

A multilayer perceptron (MLP) is a type of feedforward artificial neural network. In some contexts, the term MLP refers to networks made up of multiple layers of perceptrons (with threshold activation), whereas in others, it refers to any feedforward ANN. The term "vanilla" neural networks refer to multilayer perceptrons, particularly those having a single hidden layer [25].

An MLP has at least three node layers: an input layer, a hidden layer, and an output layer. Each node is a neuron with a nonlinear activation function, with the exception of the input nodes [25,26]. MLP employs backpropagation as a supervised learning technique during training. The multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can tell the difference between data that isn't linearly separable [27].

In our experiment, we used the MLP with classification type because the label is discrete and the parameters of MLP that used are as the following: activation = relu and random_state = 42.

eXtreme gradient boosting algorithm (XGBoost)

Extreme Gradient Boosting is abbreviated as XGBoost that used for classification and regression tasks. The gradient boosting algorithm has been parallelized and carefully optimized [28]. The training time is greatly reduced by parallelizing the entire boosting procedure. We train hundreds of models on different subsets of the training dataset and then vote on the best-performing model [28], rather than creating the best model possible on the data (as in traditional approaches). In many cases, XGBoost outperforms classic gradient boosting approaches [29]. The Python implementation gives you access to a slew of inner parameters that you may modify for better precision and accuracy [30].

The general work of this algorithm is to convert weak learners (decision trees) into strong learners, which means that the strong learner produces the final prediction label (average of each prediction by week classifier) [30]. The XGBoost [28-30] has a number of significant features: 1) Parallelization: The model is designed to run simultaneously on several CPU cores. 2) Regularization: XGBoost offers a variety of regularization penalties to prevent overfitting. Regularizations with penalties result in successful training, allowing the model to generalize successfully. 3) Non-linearity: XGBoost can recognize and learn from non-linear data patterns. 4) Cross-validation is built-in and immediately available. 5) Scalability: XGBoost can run distributed, allowing you to manage huge volumes of data, thanks to distributed servers and clusters like Hadoop and Spark. Many programming languages are supported, including C++, JAVA, Python, and Julia.

In our experiment, we used the this algorithm with classification type because the label is discrete and the parameters of XGBoost that used are as the following: colsample_bylevel = 1, learning_rate = 0.1, gamma = 0, n_estimators = 100, and random_state = 42.

AdaBoost algorithm

The statistical classification meta-algorithm AdaBoost (short for Adaptive Boosting) is a statistical classification meta-algorithm. It can be combined with a variety of other learning algorithms to boost performance. Other learning algorithms' output ('weak learners') are blended into a weighted total that represents the boosted classifier's final output [31]. AdaBoost is adaptive in that it tweaks succeeding weak learners in favor of instances misclassified by earlier classifiers. It may be less prone to the overfitting problem than other learning algorithms in some situations. Individual learners may be poor, but as long as their performance is marginally better than random guessing, the final model will converge to a powerful learner [32]. Although AdaBoost is most commonly used to combine weak base learners (such as decision stumps), it has been demonstrated that it can also be used to combine strong base learners (such as deep decision trees), resulting in a more accurate model [33].

Every learning algorithm has many different parameters and configurations to tweak before it reaches optimal performance on a dataset, and most of them fit some problem types better than others. AdaBoost is frequently referred to as the best out-of-the-box classifier (with decision trees as weak learners) [31-33]. When combined with decision tree learning, data obtained at each stage of the AdaBoost algorithm on the relative 'hardness' of each training sample is fed into the tree-growing process, causing later trees to focus on more difficult-to-classify samples [33].

In our experiment, we used the AdaBoost with classification type because the label is discrete and the parameters of GB that used are as the following: algorithm = SAMME.R, learning_rate = 1.0, n_estimators = 50 and random_state = 42.

Gradient boosting algorithm (GB)

Gradient boosting is a machine learning approach that can be used for regression and classification, among other things. It returns a prediction model in the form of a group of weak prediction models, which are decision trees [34]. Gradient boosting is a technique that combines multiple weak learners (decision trees) into a single strong learner. In this instance, individual decision trees are poor learners [35]. Each tree in the sequence is related to the one before it, with each tree striving to correct the error of the one before it. Due to this sequential relationship, boosting algorithms are often slow to train yet incredibly exact. In statistical learning, models that learn slowly perform better [34,35]. The weak learners are fitted in such a way that each new learner fits into the residuals of the previous stage as the model improves. The final model brings together the outcomes of each phase to produce a strong learner. A loss function is used to detect residuals. For example, in a regression work, mean squared error (MSE) can be utilized, and in a classification task, logarithmic loss (log loss) can be employed. It's worth noticing that when a new tree is added to the model, nothing changes. The added decision tree fits the current model's residuals [34-36].

In our experiment, we used the GB with classification type because the label is discrete and the parameters of GB that used are as the following: subsample= 1.0, learning_rate = 0.1, criterion= friedman_mse, n_estimators = 100, and random_state = 42.

Voting algorithm

A Voting Classifier is a ensemble machine learning model that learns from a group of models and predicts an output (class) based on the output's highest chance of being the desired class [37]. It simply adds up the results of each classifier fed into the Voting Classifier and predicts the output class with the highest votes [38]. We propose a single model that trains on numerous models and predicts output based on the cumulative majority of votes for each output class, rather than building separate specialized models and determining their performance. Two forms of voting are supported by Voting Classifier [39,40]. 1) Hard voting: the projected output class with the most votes, i.e. the one having the highest likelihood of being predicted by each of the classifiers, is the one with the best chance of being predicted by each of the classifiers. 2) Soft voting: the output class is a forecast based on an average of the likelihood provided to that class.

In our experiment, we used the Voting with classification type because the label is discrete and the parameters of Voting that used are as the following: estimators = DT, RF, and XGB, and voting type = hard.

Results and Discussion

This section presents the experimental results for each cybersecurity dataset in each machine learning algorithm based on four evaluation metrics.

Evaluation metrics

There are several assessment measures to examine the machine learning algorithms that were utilized, including accuracy, precision, recall, and f1-score [41]. These metrics' formulas are as follows: TP = True Positives, TN = True Negative, FP = False Positives, and FN = False Negative.

Accuracy: is the most intuitive performance metric is the ratio of properly predicted samples to total samples, which is simply a ratio of correctly predicted samples to total samples.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP + TN + FP + FN}} \qquad \text{--------------} \ (1)$$

Precision: is the ratio of correctly predicted positive tweets to the total predicted positive samples.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad \text{-----------} \ (2)$$

Recall: is the proportion of accurately anticipated positive samples to the total number of positive samples predicted.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad \text{-----------} \ (3)$$

F1-score: is the weighted average of Precision and Recall.

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision+ Recall}} \qquad \text{-------------} \ (4)$$

In this dataset, we applied one experiments to detect several types of attacks: Binary classification (normal and malicious attacks). This experiment is done to know if each algorithm is able to distinguish between the malicious attacks types in HTTP DATASET CSIC 2010 dataset. In this experiment, machine learning algorithms are applied to detect if the sample in this dataset is normal or a malicious attack. As shown in the Table 4 and Figure 3, the performance results for machine learning algorithms used are based on four metrics: precision, accuracy, f1-score, and recall. The Voting, and DT classifiers outperforms the higher performance results compared with the others in detection attack process.

Table 4: Performance Results of Binary Classification.

Models	Accuracy	Precision	Recall	F1-score
DT	0.894056	0.878958	0.864066	0.871448
RF	0.879974	0.857993	0.852246	0.85511
GB	0.829376	0.733313	0.92632	0.818593
XGB	0.888489	0.852908	0.884161	0.868253
AdaBoost	0.774685	0.731106	0.724192	0.727633
MLP	0.735713	0.747059	0.550433	0.633848
Voting	0.893074	0.869463	0.873916	0.871684

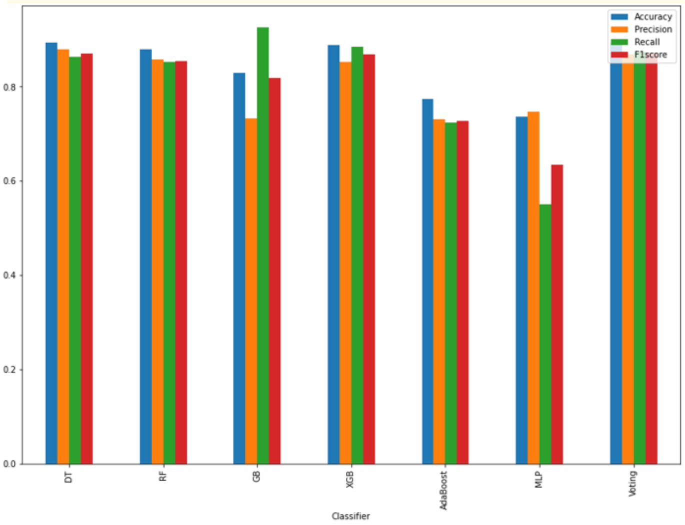

Figure 3: Performance Results of Binary Classification.

Bibliography

1. Seemma PS., *et al*. “Overview of cyber security”. *International Journal of Advanced Research in Computer and Communication Engineering* 7.11 (2018): 125-128.

2. Ervural B C and Ervural B. “Overview of cyber security in the industry 4.0 era”. In Industry 4.0: managing the digital transformation (2018): 267-284.

3. Chowdhury A. “Recent cyber security attacks and their mitigation approaches–an overview”. In International conference on applications and techniques in information security (2016): 54-65.

4. El-Rewini Z., *et al*. “Cybersecurity challenges in vehicular communications”. *Vehicular Communications 23* (2020): 100214.

5. Handa A., *et al*. “Machine learning in cybersecurity: A review”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9*.4 (2019): e1306.

6. Kim A., *et al*. “AI-IDS: Application of deep learning to real-time Web intrusion detection”. *IEEE Access* 8 (2020): 70245-70261.

7. Vartouni A M., *et al*. “An anomaly detection method to detect web attacks using stacked auto-encoder”. In 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS) (2018): 131-134.

8. Betarte G., *et al*. “Web application attacks detection using machine learning techniques”. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (2018): 1065-1072.

9. Tuan T A., *et al*. “Performance evaluation of Botnet DDoS attack detection using machine learning”. *Evolutionary Intelligence* (2019): 1-12.

10. Anwer M., *et al*. “Attack Detection in IoT using Machine Learning”. *Engineering, Technology and Applied Science Research 11*.3 (2021): 7273-7278.

11. Su T., *et al*. “BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset”. *IEEE Access* 8 (2020): 29575-29585.

12. Xu W., *et al*. “Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset”. *IEEE Access* 9 (2021): 140136-140146.

13. Kavitha S and Uma Maheswari N. “Network Anomaly Detection for NSL-KDD Dataset Using Deep Learning”. *Information Technology in Industry 9*.2 (2021): 821-827.

14. Ferriyan A., *et al*. “Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic”. *Applied Sciences 11*.17 (2021): 7868.

15. Giménez C T., *et al*. “HTTP data set CSIC 2010”. *Information Security Institute of CSIC (Spanish Research National Council)* (2010).

16. Hancock J T and Khoshgoftaar T M. “Survey on categorical data for neural networks”. *Journal of Big Data 7*.1 (2020): 1-41.

17. Pal M. “Random forest classifier for remote sensing classification”. *International Journal of Remote Sensing* 26.1 (2005): 217-222.

18. Farnaaz N and Jabbar M A. “Random forest modeling for network intrusion detection system”. *Procedia Computer Science* 89 (2016): 213-217.

19. Idhammad M., *et al*. “Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest”. *Security and Communication Networks* (2018).

20. Kingsford C and Salzberg SL. “What are decision trees?”. *Nature Biotechnology* 26.9 (2008): 1011-1013.

21. Quinlan J R. “Induction of decision trees”. *Machine Learning* 1.1 (1986) 81-106.

22. De Ville B. “Decision trees”. *Wiley Interdisciplinary Reviews: Computational Statistics* 5.6 (2013): 448-455.

23. Kotsiantis SB. “Decision trees: a recent overview”. *Artificial Intelligence Review 39*.4 (2013): 261-283.

24. Amor N B., *et al*. “Naive bayes vs decision trees in intrusion detection systems”. In Proceedings of the 2004 ACM symposium on Applied computing (2004): 420-424.

25. Noriega L. “Multilayer perceptron tutorial”. *School of Computing. Staffordshire University* (2005).

26. Tang J., *et al*. “Extreme learning machine for multilayer perceptron”. IEEE transactions on neural networks and learning systems 27.4 (2015): 809-821.

27. Ramchoun H., *et al*. “Multilayer perceptron: Architecture optimization and training” (2016).

28. Mitchell R and Frank E. “Accelerating the XGBoost algorithm using GPU computing”. *Peer Journal of Computer Science 3* (2017): e127.

29. Pan B. “Application of XGBoost algorithm in hourly PM2. 5 concentration prediction”. In IOP conference series: earth and environmental science 113.1 (2018): 012127.

30. Dong W., *et al*. “XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring”. *Automation in Construction* 114 (2020): 103155.

31. Hu W and Hu W. "Network-based intrusion detection using Adaboost algorithm". In The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05) (2005): 712-717.

32. Jabri S., *et al*. "Moving vehicle detection using Haar-like, LBP and a machine learning Adaboost algorithm". In 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS) (2018): 121-124.

33. Yuan L and Zhang F. "Ear detection based on improved adaboost algorithm". In 2009 International Conference on Machine Learning and Cybernetics (2009): 2414-2417.

34. Son J., *et al*. "Tracking-by-segmentation with online gradient boosting decision tree". In Proceedings of the IEEE international conference on computer vision (2015): 3056-3064.

35. Peter S., *et al*. "Cost efficient gradient boosting". *Advances in Neural Information Processing Systems* 30 (2017).

36. Lusa L. "Gradient boosting for high-dimensional prediction of rare events". *Computational Statistics and Data Analysis* 113 (2017): 19-37.

37. Kumar U K., *et al*. "Prediction of breast cancer using voting classifier technique". In 2017 IEEE international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM) (2017): 108-114.

38. El-Kenawy E S M., *et al*. "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images." *IEEE Access 8* (2020): 179317-179335.

39. Khan M A., *et al*. "Voting classifier-based intrusion detection for iot networks". In Advances on Smart and Soft Computing (2022): 313-328.

40. Mahabub A. "A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers". *SN Applied Sciences 2*.4 (2020): 1-9.

41. Dalianis H. "Evaluation metrics and evaluation". In Clinical text mining (2018): 45-53.