Effectiveness of Machine Learning based Intrusion Detection Systems

Mohammed Alrowaily[1], Freeh Alenezi[2], and Zhuo Lu[1]

[1] Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA
malrowaily@mail.usf.edu, zhuolu@usf.edu
[2] Department of Mathematics and Statistics, University of South Florida, Tampa, FL 33620, USA
fnalenezi@usf.edu

Abstract. Security is the most significant issue in concerns of protecting information or data breaches. Furthermore, attackers present a new variety of cyber-attacks in the market, which prevent users from managing their network or computer system. For that reason, the growth of cybersecurity research studies, such as intrusion detection and prevention systems have great significance. The intrusion detection system (IDS) is an effective approach against malicious attacks. In this work, a range of experiments has been carried out on seven machine learning algorithms by using the CICIDS2017 intrusion detection dataset. It ensued to compute several performance metrics to examine the selected algorithms. The experimental results demonstrated that the K-Nearest Neighbors (KNN) classifier outperformed in terms of precision, recall, accuracy, and F1-score as compared to other machine learning classifiers. Nevertheless, All of the used machine learning classifiers except KNN trained their models in a reasonable time.

Keywords: Intrusion Detection System · Machine learning · IDS dataset.

1 Introduction

Intrusion is an intense problem in security and a prime complication of data breaches, given that a single circumstance of intrusion may steal or even delete information coming from computer as well as network units in a few seconds. Intrusion can easily also destroy system equipment. Additionally, the intrusion may trigger significant reductions economically as well as weaken the IT crucial facilities, thereby causing info inferiority in cyber war. For that reason, intrusion detection is necessary, and also its prevention is required [1]. The appearance of cutting-edge attacks drives the commercial enterprise and academic community to look into for unique approaches, which manage to tightly keep track of this competition and fine-tune rapidly to the transformations in the field [6].

Network security can be attained by employing a software application called an Intrusion Detection Systems (IDS) that helps to withstand network breaches.

The objective of these systems is to have shield wall which prevents such types of attacks. It identifies the illegal activities of a network or a computer system. Generally, there are two major categories of IDS, namely Anomaly detection and Misuse detection. The former learns from recorded normal behavior to identify new intrusion attacks. Any variance from existing baseline patterns is determined as attacks and alarms are triggered. Nevertheless, misuse detection detects the intrusion based on the repository of attacks signatures but has no false alarm.

Machine learning approaches has been extensively utilized in determining different sorts of attacks, which is a powerful tool to enhance network security. In addition, it can assist the network's monitoring team in taking the necessary countermeasures for protecting against intrusions.

In this paper, we utilize the public real-world intrusion dataset CICIDS2017 [7], which includes benign and the most sophisticated attacks and presenting results of seven machine learning classifiers, such as AdaBoost, Naive-Bayes (NB), Random Forest(RF), Decision Tree, Multi-layer perceptron (MLP), K-Nearest Neighbors (KNN), and Quadratic Discriminant Analysis (QDA).

The main contributions of this paper at hand are as follows:

- First, the discussion of various existing literature studies for building an IDS using different machine learning classifiers is presented, emphasizing on the detection mechanism, applied feature selection, attacks detection efficiency.
- Second, we examine the CICIDS2017 dataset that includes benign and the most cutting-edge common attacks. Likewise, we carried out various machine learning algorithms to analyze the detection performance of IDS.
- Finally, we extensively evaluate our system over different performance metrics such as accuracy, precision, recall, and F1-score, training and prediction time.

The remaining parts of this paper are organized as follows. Section 2, presents a literature review of the related work that only uses the same CICIDS2017 dataset for intrusion detection. Section 3, introduces the implemented dataset in details with explanation of the attack scenarios. Section 4, gives a brief overview of machine learning classifiers. Section 5, discusses the performance results of the classifiers over different evaluation metrics. Finally, the conclusion to our work is given in Section 6.

2 Related Work to the CICIDS2017 Dataset

Over the last few years, attempts to attacks on determining sizable data have revved up. In this part, different research studies employing machine learning for intrusions detection have been analyzed. In each research study, the applied machine learning algorithms and system performance are provided. When selecting these research studies, the focus was on the ones that used different machine learning algorithms on the CICIDS2017 dataset.

Sharafaldin et al. [7] have proposed a new dataset named as the CICIDS2017. Their IDS experiments were performed over seven well-known machine learning classifiers, namely AdaBoost, Random Forest, Naive Bayes, ID3, MLP, KNN, and QDA. They claim that the highest accuracy was achieved by KNN, RF and ID3 algorithms, but this paper is lack of accuracy rate results.

Ustebay et al. [8] propose a hybrid IDS using the CICIDS2017 dataset, which combining classifier model based on tree-based algorithms namely REP Tree, JRip algorithm, and Random Forest. They claim that their proposed system experimental results prove superiority supremacy in terms of false alarm rate, detection rate, accuracy and time overhead as compared to state of the art existing schemes. Attacks are detected with 96.665% accuracy rate.

Boukhamla et al. [4] describe and optimize the CICIDS2017 dataset using Principal Component Analysis (PCA), which results in dimensionality reduction without losing specificity and sensitivity. Hence, decreasing the overall size and bring on faster IDS. This work has been employed on the recorded data of Friday and Thursday, which targeted various attacks (DDoS, Botnet, Port-scan, Web attacks and Infiltration). The dataset is examined employing three classifiers including KNN, C4.5 and Naive Bayes. The highest detection rate for DDoS was achieved by Naive Bayes, and KNN classifiers are 90.6% and 99% respectively. As a result, Naive Bayes has an elevated false alarm rate (59%) which in turn classify KNN (with 1.9% of false alarm rate) as a sufficient classifier for a DDoS attack. The number of attributes had notably been lowered, roughly by 75%, of the total attributes number.

Zegeye et al. [9] proposed a machine learning Multi-Layer Hidden Markov (HMM) model based intrusion detection. This multi-layer approach factors a substantial issue of large dimensionality to a discrete set of reliable and controllable elements. Moreover, it can be broadened further than two layers to capture multi-phase attacks over long periods of time. The portion of Thursday morning records in the CICIDS2017 dataset was used which comprises of Web Attack-Brute Force, SSH Patator, and Benign traffic. The proposed system reveals a good performance among all evaluation metrics as 98.98% accuracy, 97.93% precision, 100% recall, and 98.95% F_measure.

Aksu et al. [2] propose an IDS using supervised learning techniques and Fisher Score feature selection algorithm, on the CICIDS2017 dataset for benign and DDoS attacks. Their work was performed on Support Vector Machine, Decision Tree and K-Nearest Neighbours machine learning algorithms. The performance measurements show that the KNN performed much better outcomes with 30 features; the examination scores did not change for Decision Tree algorithm. Alternatively, SVM's outcomes did not fulfill with both 80 and 30 features. After using Fisher Score feature selection, the dataset was reduced by 60%. As an accuracy outcome of this study, 0.9997% KNN, 0.5776% SVM, 0.99% DT accomplished when selecting 30 features.

Hou et al. [5] presented a machine learning approach based DDoS attack detection via NetFlow analysis. Different machine learning classification algorithms were primarily evaluated namely C4.5 Decision Tree, Random Forest,

AdaBoost, and Support Vector Machines against their NetFlow collected data. This DDoS detection approach was secondarily evaluated by using public dataset CICIDS2017 to prove its validity. The experiment consequences indicate that this approach obtains an average accuracy of 97.4% and a false positive 1.7%.

Bansal and Kaur [3] proposed an intrusion detection approach, named XGBoost. In the study, the relevant system created by employing the Wednesday recorded dataset that consists of various sort of DoS attacks from the CICIDS2017. The accuracy of 99.54% was obtained in the case of multi-classification of DoS attacks.

In the relevant works, it is witnessed that research studies employing the same dataset are presenting excellent results. However, when the research studies examined; it is observed that most of the authors partially used the CICIDS2017 dataset in their IDS implementation, which therefore indicates that their IDS are only exposed to some of the attacks in the subject dataset.

3 Data Pre-processing and Analysis

The process of analyzing any given dataset to develop an IDS should certainly involve understanding the dataset in hand, cleaning, then carrying out some powerful statistical methods, that assure achieving the study's goals, along with their predetermined performance metrics. This section shows the process of analyzing CICIDS2017 dataset.

3.1 Benchmark Dataset

CICIDS2017 Dataset [7] generated by the Canadian Institute for Cybersecurity at the University of New Brunswick. Benign and the most sophisticated widespread attacks, for instance, real-world data (PCAPs), are featured in CICIDS2017 dataset. This dataset includes five days records stream on a network generated by computer systems using updated operating systems (OS) which provides for Windows Vista/ 7/ 8.1/ 10, Mac, Ubuntu 12/16 and Kali. Monday records consist of benign traffic. The employed attacks are Brute Force SSH, Brute Force FTP, Infiltration, Heartbleed, Web Attack, DoS, Botnet, and DDoS. All attacks had been applied between Tuesday and Friday.

The formerly available network traffic datasets suffer from the absence of traffic diversity, volumes, anonymized packet information payload, constraints on the attacks range, the lack of the feature set and metadata. Therefore, the CICIDS2017 came to conquer these concerns like different protocols including HTTP, HTTPS, FTP, SSH and also e-mail protocols, which in turn were not offered in the dataset previously. The first two columns of Table 1 present the attack label and their corresponding counts. This number of attack labels is moderately large, where some labels are sufficiently smaller than others, this in fact what makes analyzing the CICIDS2017 dataset still an open issue and there is always a space for improvements in the existing or new machine learning algorithms.

Table 1. Attack Distribution in CICIDS2017 Dataset

Attack label	**Flow Count**	**Flow Count (w/cleansing)**	**Difference**	**Proportion (%)**
Benign	2273097	1893223	379874	0.167
DoS Hulk	231073	173791	57282	0.247
Port Scan	158930	1956	156974	0.012
DDoS	128027	128020	7	0.000
DoS GoldenEye	10293	10286	7	0.000
FTP-Patator	7938	6093	1845	0.232
SSH-Patator	5897	3360	2537	0.430
DoS Slowloris	5796	5385	411	0.070
DoS Slowhttptest	5499	5242	257	0.046
Botnet	1966	1437	10	0.269
Web: Brute Force	1507	37	0	0.024
Web: XSS	652	652	0	0.000
Infiltration	36	36	0	0.000
Web: SQL Injection	21	21	0	0.000
Heartbleed	11	11	0	0.000
Total	2830743	2230983	599760	2.477%

3.2 Description of Attack Scenarios

Here in this dataset, six attack profiles are covered based upon the most updated list of commonly used attack families, which can be explained as follows:

Web Attack: Three web attacks have been implemented in their dataset. First, SQL Injection is an application security vulnerability in which an attacker interferes with the queries that an application makes to its database, to let the unauthorized users view the data. Second, Cross-Site Scripting (XSS) which is happening when the attacker injects malicious code into the victims web application. Last, Brute Force which tries a probabilistic entire possible passwords to decode the administrators password.

Botnet Attack: A collection of internet-connected devices such as a home, office or public systems, contaminated by harmful malicious code called malware. It can enable the attacker access to the device and its connection for stealing, taking down a network and IT environment. Botnets attack are remotely controlled by cybercriminals and have turned into one of the most significant threats to security systems today.

Heartbleed Attack: is a severe bug in the implementation of OpenSSL, an open-source implementation of the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. This vulnerability allows malicious hackers to read portions and steal data from the memory of the victim server.

Brute Force Attack: is a dictionary attack method that generates many successive estimates as to access encrypted data. This attack is commonly used for

cracking passwords, locating the hidden web page or content, and decoding Data Encryption Standard (DES) keys.

DDoS Attack: is one of the most popular cyber weapons, in which attempt to exhaust the resources available to an online service and network by flooding it with traffic from several compromised systems, deny legitimate users access to the service.

DoS Attack: is a type of cyber attack on a network that is designed to prevent legitimate users temporarily from accessing computer systems, devices, or other network resources due to malicious cyber activities.

Infiltration Attack: is a piece of malicious that attempts to enter or damage the inside of the network which is generally manipulating a susceptible software like Adobe Acrobat/Reader.

3.3 Evaluation Metrics

Our work subject to different evaluation metrics, which are accuracy, precision, recall, F1-score, training time and prediction time. Since achieving the supreme accuracy does not essentially signify that the classifier properly predicts with high reliability. As a result, we utilize other strategies to examine the reliability of the proposed system results. Table 2 shows the description of confusion matrix.

Table 2. Confusion Matrix

		Predicted Class	
		Classified as Normal	**Classified as Attack**
Actual Class	**Normal**	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

The evaluation metrics are specified based on the following explanations:

- True Positive (TP): describes the number of attacks correctly detected.
- True Negative (TN): describes the number of normal correctly detected.
- False Positive (FP): describes the number of normal wrongly detected.
- False Negative (FN): describes the number of attacks wrongly detected.

Afterward, we calculate the evaluation metrics from the following formulas as shown in Table 3.

- Precision: the proportion of correctly predicted attack relative to all data classified as the attack
- Accuracy: the proportion of records are correctly determined as attack and normal
- F1-Score: a combination that measures the harmonic average of precision and recall.
- Recall: indicating the proportion of correctly predicted attack to all attack data.

– Training time: represents the time consumed for a particular algorithm to train the model for the entire dataset.
– Prediction time: represents the time consumed for a particular algorithm to predict the entire dataset as benign or attack.

Table 3. Evaluation Metrics

Metric	Definition
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	$Pr = \frac{TP}{TP+FP}$
Recall	$Rc = \frac{TP}{TP+FN}$
F1-Score	$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$

3.4 Data Cleansing

We observed that CICIDS2017 dataset include some significant pitfalls which cause the classifier to be biased, and the goal of this paper is to address those imperfections and apply machine learning classification properly to make more accurate results. It might be an essential step to make some modifications to the dataset employing it in practice, rendering it more reliable. For this purpose, in this part, some pitfalls of the CICIDS2017 dataset are remedied, and some data are modified. The dataset contains 2830743 records and 86 features. The updated distribution of this dataset can be shown in Table 1. When we examine these records, it can be noticed that 599760 are faulty records. The first step in the data pre-processing will be to remove these undesirable records.

An additional change that requires to be made in the dataset is that we remove all rows with features "Flow Bytes/s" and "Flow Packets/s" that have either "Infinity" or "NaN" values. Furthermore, we remarked that some features have zero values for all rows, namely Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, and Bwd Avg Bulk Rate , hence, they are also excluded.

We noticed that the attack label with small counts still maintains that count before and after cleaning the data. By looking at the proportion column, a tiny proportion of each attack type was deleted during the data cleaning process. Lastly, the first column "Destination Port" is also excluded, even though when it was included, we noticed an improvement in the performance of the classifiers. Therefore, the data size used for the analysis is 2230983 records by 69 features.

After the removal of these features, the dataset is randomly split into two parts, 70% was used for training, and 30% was used for testing the model, in order to evaluate their performance in the intrusion detection system.

3.5 Feature Selection

The Random Forest classifier was used to calculate the importance score for each feature. Then, along with the original dataset 69 features, we selected 10, 30 as the most efficient features that can distinguish the information in the most significant way.

4 Overview of Machine Learning Classifiers

This part presents a concise overview of the various machine learning supervised algorithms and demonstrates the needs to carry out machine learning algorithms in numerous areas just like IDS. The implication of ongoing development of modern technologies creates the demand for machine learning algorithms to emerge as more necessary for extracting and analyzing knowledge from a substantial number of created datasets. In this paper, our interest is employing the following machine learning algorithms; due to the fact that the intended CICIDS2017 dataset consists of the pre-defined classes.

Adaptive Boosting (AdaBoost): a boosting approach, is a machine learning algorithm designed to enhance classification efficiency. The fundamental working concept of boosting algorithms can be described as follows; the data are initially sorted into groups with rough draft rules. On any occasion the algorithm is run, new rules are contributed to this rough draft rules. In this manner, several feeble and low-performance rules called "basic rules" are acquired.

Multi-layer perceptron (MLP): is a category of artificial neural networks (ANN). ANN is a machine learning technique that **takes motivation from the method the human brain works.** The objective of this approach is to mimic the human brain properties, for instance, making decisions and obtaining new information. While the human brain is comprised of interconnected nerve cells, ANN is comprised of interconnected artificial cells.

Decision Tree (DT): is the most potent tool in classification and prediction. A Decision Tree is flow diagram such as tree structure, where each tree includes leaves, branches, and nodes. It divides a dataset into scaled-down subsets while simultaneously an associated decision tree is incrementally formed. The final outcome is a tree with leaf nodes and decision nodes.

Naive Bayes (NB): is a family of probabilistic classification technique that benefits from probability theory and the Bayes Theorem for predictive modeling, which presumes that all attributes are statistically independent. It computes the probabilities for each factor in order to single out the result that has the highest probability.

K-Nearest Neighbors (KNN): is a versatile and sample-based method. It depends on in which the data points are separated into multiple classes, in other words, similar things are near to each other, in order to determine the K-nearest neighbors.

Quadratic Discriminant Analysis (QDA): is a discriminant analysis method that is utilized to identify which variables differentiate between two or

more naturally taking place groups; it may have a predictive or a descriptive goal.

Random Forest (RF): is a machine learning approach that utilizes decision trees. Herein method, a "forest" is produced by putting together a substantial number of various decision tree structures which are created in various ways.

5 Test Results and Discussion

The results of using the aforementioned machine learning classifiers are given in Table 4. Based on the values of precision, recall, and F1-Score, the KNN has the best performance among other classifiers, followed by the MLP and Random Forest classifiers. Then, the performance of the Decision Tree, AdaBoost, and Naive Bayes are ranked as fourth, fifth and sixth, respectively. The QDA algorithm has the lowest performance results.

Table 4. Classifier Performance Results for all 15 Attacks

Algorithm	Precision	Recall	F1-Score	Accuracy
Random Forest	0.9469	0.9571	0.9483	0.9571
KNN	0.9953	0.9955	0.9950	0.9955
Naive Bayes	0.7958	0.8487	0.7794	0.8488
Decision Tree	0.8821	0.9040	0.8920	0.9041
MLP	0.9641	0.9705	0.9662	0.9705
AdaBoost	0.8578	0.9173	0.8854	0.9173
QDA	0.7204	0.8488	0.7794	0.8488

The training and predicting times were also computed during the process, and given by Table 5. It can be noted that the KNN requires significantly more time during the training and testing process. This in fact could be a drawback of the classifier, as it memorizes all the training flows. Naive Bayes has the lowest training and predicting times among other classifiers, but, as mentioned earlier, it performed as a second worst classifier on the CICIDS2017 dataset. Thus, it is a trade-off between the performance and prediction time. On the other hand, the MLP classifier has a good balance between its performance and the prediction time.

Table 5. Classifier Training and Prediction Time

Classifier	Training (Sec.)	Prediction (Sec.)
Random Forest	348.6	5.8
KNN	2590.6	1358.1
Naive Bayes	4.6	7.7
Decision Tree	19.9	0.2
MLP	103.7	1.1
AdaBoost	607.6	15.5
QDA	15.2	10

Since the total number of features after the data cleaning process is 68, the feature importance based on the Random Forest classifier was computed, which helped to rank the 10 and 30 most important features, respectively. The subject machine learning classifiers were carried out on the reduced CICIDS2017 dataset, and the results are given by Table 6. The results indicate similar performance consistency of the classifiers when using only 10 and 30 most important features, respectively. Nevertheless, the performance of the classifiers was higher when considering all the 68 features.

Table 6. Performance Results of 10 and 30 Features

Algorithm	**No. of features**	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest	30	0.9395	0.9484	0.9382	0.9485
	10	0.9287	0.9401	0.9283	0.9401
KNN	30	0.9944	0.9945	0.9941	0.9946
	10	0.9690	0.9675	0.9675	0.9676
Naive Bayes	30	0.7958	0.8487	0.7794	0.8488
	10	0.7204	0.8488	0.7794	0.8488
Decision Tree	30	0.8816	0.9025	0.8907	0.9026
	10	0.9282	0.9417	0.9305	0.9418
MLP	30	0.9536	0.9625	0.9557	0.9626
	10	0.9347	0.9460	0.9356	0.9460
AdaBoost	30	0.8578	0.9173	0.8854	0.9173
	10	0.8692	0.8901	0.8576	0.8901
QDA	30	0.7204	0.8488	0.7794	0.8488
	10	0.7204	0.8488	0.7794	0.8488

The results can be wrapped up in the following points:

- Despite the training and predicting times, the best performer was found to be the K-Nearest Neighbors (KNN) classifier based on all four evaluation metrics.
- The MLP achieved the second highest performance, and it maintained reasonably small training and prediction times.
- The chosen machine learning classifiers excluding KNN trained their models in a reasonable time period.
- The feature selection based on the Random Forest classifier did not support the classifiers to perform better compared to the usage of all features after the data cleansing process.
- There is no significant difference in the performance of the Naive Bayes and QDA classifiers based on the evaluation metrics, where both have the worst overall performance, regardless of their small training and predicting times.

6 Conclusion

In this paper, several IDS experiments were carried out to examine the efficiency of seven machine learning classifiers, namely AdaBoost, Random Forest, Naive Bayes, Decision Tree, MLP, KNN, and finally QDA. We make use of public intrusion detection dataset (CICIDS2017), which includes benign and most sophisticated popular attacks. The experimental results attest the superiority of the K-Nearest Neighbors (KNN) classifier in terms of various performance metrics such as precision, recall, accuracy and F1-score among other machine learning algorithms. However, all of the selected machine learning classifiers excluding KNN trained their models in an acceptable time period.

Acknowledgment

Mohammed and Freeh would thank Jouf and Majmaah Universities, respectively, for the scholarship funds.

References

1. Ahmad, I., Basheri, M., Iqbal, M.J., Rahim, A.: Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access **6**, 33789–33795 (2018)
2. Aksu, D., Üstebay, S., Aydin, M.A., Atmaca, T.: Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. In: International Symposium on Computer and Information Sciences. pp. 141–149. Springer (2018)
3. Bansal, A., Kaur, S.: Extreme gradient boosting based tuning for classification in intrusion detection systems. In: International Conference on Advances in Computing and Data Sciences. pp. 372–380. Springer (2018)
4. Boukhamla, A., Gaviro, J.C.: Cicids2017 dataset: performance improvements and validation as a robust intrusion detection system testbed
5. Hou, J., Fu, P., Cao, Z., Xu, A.: Machine learning based ddos detection through netflow analysis. In: MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM). pp. 1–6. IEEE (2018)
6. Papamartzivanos, D., Mármol, F.G., Kambourakis, G.: Introducing deep learning self-adaptive misuse network intrusion detection systems. IEEE Access (2019)
7. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP. pp. 108–116 (2018)
8. Ustebay, S., Turgut, Z., Aydin, M.A.: Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In: 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT). pp. 71–76. IEEE (2018)
9. Zegeye, W., Dean, R., Moazzami, F.: Multi-layer hidden markov model based intrusion detection system. Machine Learning and Knowledge Extraction **1**(1), 265–286 (2019)