Performance of Machine Learning Techniques in Anomaly Detection with Basic Feature Selection Strategy - A Network Intrusion Detection System

Md. Badiuzzaman Pranto, Md. Hasibul Alam Ratul, Md. Mahidur Rahman, Ishrat Jahan Diya, and Zunayeed-Bin Zahir
Department of ECE, North South University, Dhaka, Bangladesh
Email: {badiuzzaman.pranto, hasibul.ratul, mahidur.rahman, ishrat.jahandiya, zunayeed.zahir01}@northsouth.edu

***Abstract*—With the proliferation of internet users around the world, it is becoming imperative to make communications safer than before. A network intrusion detection system is pivotal for network security because it enables us to detect and respond to malicious traffics. There are several ways and available tools to detect attacks in a computer network but machine learning techniques are one of the most efficient methods to detect abnormal traffics precisely and accurately. In this work, a method has been demonstrated to classify if incoming network traffic is normal or anomalous using machine learning techniques. Several classifiers have been evaluated based on the NSL-KDD dataset. Experiments were conducted with k-nearest neighbor, decision tree, naïve Bayes, logistic regression, random forest, and their ensemble approach. A basic feature selection strategy has been applied to reduce the calculation time complexity and dataset's dimension. The highest accuracy obtained 99.5% with a 0.6% false alarm rate.**

***Index Terms*—intrusion detection system, machine learning, cyber security, inductive learning**

I. Introduction

Nowadays, the internet is probably the most available thing around the world. According to internet world stats, there are around 4.57 billion internet users among 7.79 billion people till 31 Dec, 2019 [1]. Online servers apparently are the store-house of data. Nearly every important information: personal, private, or confidential data are being stored on servers. However, there appears a malicious attack every 39 seconds, which is why about half a billion personal records were stolen by hackers in 2018 [2]. With the vigorous growth of data available on online servers, the significance of accurate intrusion detection systems is becoming more essential.

An intrusion refers to the unauthorized activities on a computer network [3], while an Intrusion Detection System (IDS) monitors the events occurring in a computer system or network and analyze them for the sign of intrusions [4]. There are various methods for performing network attacks. Some intrusions are meant to force network shutdown, some are used to interrupt services and some are for stealing data. A few popular attacks are DOS, DDOS, Brute-force, Man-in-the-Middle (MitM), Phishing and spear phishing, Drive-by attack, Password attack, SQL injection, Cross-site scripting (XSS) attack, etc. Several prestigious organizations have experienced numerous severe network attacks in the past decades. For instance, Amazon Web Services (AWS) encountered a DDoS attack which lasted for around eight hours in 2019 [5]. During the same time Google Cloud Platform (GCP) also dealt with a range of issues [5]. Around 75% of the entire healthcare, industries have been infected with malware in 2015 [6]. In 2018, the Federal Bureau of Investigation (FBI) reported: due to Business Email Compromise (BEC) and Email Account Compromise (EAC) scams have reached around $12.5 billion worldwide [7]. The Corero network security performed a survey among 327 security professionals in 2018 and reported [8]:

- Every month, more than two-thirds of organizations experience around 20 to 50 DDoS attack attempts.
- 91% of security professionals believe that an individual DDoS attack can cost their organizations up to $50,000.
- 78% of security professionals stated the most damaging effect of DDoS attacks on businesses is the loss of customer trust and confidence.

Therefore, the statistics suggest that the online servers of information storage are most vulnerable to face a considerable amount of threats in the future if the intruders are not inhibited from such activities. This compels to design an efficient intrusion detection system. Apparently, this research work has substantial importance in order to secure cyberspace.

A. Why Machine Learning for Intrusion Detection?

Machine learning is one of the world's most noteworthy techniques to learn from data without explicitly programmed [9], [10]. It's also called *the inductive learning method* where learner discovers rules by observing examples. Machine learning focuses to provide algorithms that can be trained to perform a task. It involves various methods for analyzing as well as

Manuscript received July 13, 2021; revised December 3, 2021.

© 2022 J. Adv. Inf. Technol.
doi: 10.12720/jait.13.1.36-44

solving classification and regression problems in an effective way. This particular approach is so powerful that it can deal with both labeled (supervised) and unlabeled (unsupervised) data. The reasons to choose machine learning for the detection of intrusion in the first place are pervasive. A machine learning IDS is capable to change its execution strategy as it is acquainted with new information [11]. It also follows an anomaly-based detection method rather than a signature-based method. We will learn more about the detection methods of IDS in Section II.

The rest of the paper is organized as:- a short description about the types of IDSes are provided in Section II, Section III makes appearance of remarkable works on related fields, a summary of the dataset is described in Section IV followed by the research methodology in Section V and results in Section VI.

II. Types of IDS

Any Intrusion Detection System is either host-based IDS (HIDS) or Network-based IDS (NIDS) by architecture [11]. Traditionally, they can further be divided into two categories for their detection strategies: signature-based detection and anomaly-based detection [11], [12]. Let's understand the explanations of these four types of IDSes.

A. HIDS

HIDS analyses a host/computer events such as system calls and process identifiers, usually OS information. They work on a specific host and secure the host from all kinds of skeptical activities [13]-[15].

B. NIDS

A NIDS traces network-related data such as IP addresses, traffic volume, protocol usage, service ports, flags, etc. NIDS scans every incoming packet in an active network and looks for skeptical activities [11].

C. Signature-Based Detection

Signature-based detection methods work similarly with to-day's antivirus system. They store the known signatures of attacks in their database and compare the incoming traffics with stored ones. This method works well until a new attack pattern encounters. The signature-based detection system is unable to identify unknown intrusion patterns [12], [16].

D. Anomaly-Based Detection

Unlike signature-based detection methods, anomaly-based IDSes create a baseline of normal traffics and activity taking place on the network rather than memorizing the known patterns. They can measure the present state of traffic on the network against this baseline to detect intrusions that are not present in the traffic [11], [12]. Such methods are useful to detect intrusions that are designed to skip IDSes.

This research work focuses on NIDS with anomaly-based detection strategies since these types of IDS are more promising than others.

III. Related Work

Intrusion detection systems are one of the vital concerns of security professionals and researchers for ages. Vigorous researches have been conducted in the past two decades, and more are still going on. This section will go through some notable works done by researchers on the KDD'99 and NSL-KDD dataset. The core differences between this two dataset are described in Section IV.

In one work [17], C. Yin *et al.* evaluated deep learning performance in intrusion detection systems using Recurrent Neural Networks (RNN). They have tested through binary classification as well as multi-class classification and compared the results with each classifier. They obtained the highest accuracy of 83.28% in binary classification and 81.29% in multi-class classification. For both cases, RNN appeared to be the best performer compared to J48, naïve bayes, random tree, random forest, NB tree, multilayer perceptron, and SVM.

In another work [4], S. Peddabachigari *et al.* experimented with a combination of decision tree and support vector machine as a hierarchical hybrid intelligent system model (DT-SVM) and an ensemble approach combining the base classifiers. They considered the problem as a multi-class (five class) classification problem. Their experimental results illustrate that the ensemble approach provided the best average accuracy of around 92.95% as a five class classifier.

L.-H Zhang *et al.* conducted one more mentionable work [18]. They focused on Rough Set Classification (RSC) for feature extraction and generation of intrusion detection models. They have obtained around 91% average detection rate on three-class classification problem while evaluating with RSC algorithm. However, SVM performed better than RSC with an average detection rate of 99.45% in their experiment.

In another paper [19], P. Sangkatsanee *et al.* developed a real-time intrusion detection system using the decision tree. They used only 12 important features to detect anomaly activities over the internet. The highest detection rate they obtained was around 98% within 2 sec while detecting DOS and Probe.

Several feature selection techniques have also been used in researches. In one paper [20], Y. Bouzida *et al.* used Principal Component Analysis (PCA) [21] techniques for reducing the dimensionality of the dataset (KDD'99) without losing any information. They have experimented with the k-nearest neighbor and decision tree on their research and demonstrated that decision tree with a combination of PCA techniques performs better than others and gives an average accuracy of 92.63%.

Unsupervised techniques have also been experimented with in some researches. In one notable work [22], K. Leung *et al.* used clustering algorithms on the KDD'99 dataset and found that the performance was remarkable compared to the existing results. The modified clustering-tv algorithm performed best. It obtained 97.3% area under the ROC curve, and they have inferred the

© 2022 J. Adv. Inf. Technol.

clustering techniques provide advantages in calculation complexity.

Even host-based IDSes have experimented with machine learning techniques. For instance, in this work [23] R. Moskovitch *et al.* have collected 323 features to measure the host's behavior. They have evaluated four classification algorithms on several feature subsets. Their deduction implies, they achieved above 90% average accuracy on their experiment.

In few pieces of research, authors tried to bring up the adversity of some particular algorithms. For example, in one research [24], Z. Wang *et al.* have investigated the state-of-the-art attack algorithms against deep learning-based intrusion detection system. The author believed Deep Neural Networks (DNN)s leave opportunities for attackers to deceive the DNN into misclassification. By his work, he has acquired the validation of such vulnerabilities of DNN in intrusion detection.

Research Goal:

So, there exist abundant researches on intrusion detection combined with machine learning. This paper tends to get the following answer by several experiments:

- Can we acquire a competitive performance compared to the existing ones from this particular dataset by pursuing a very basic feature selection strategy. (Algorithm 1)?
- Can a competitive performance be acquired by traditional classifiers? Or by ensemble learning?

IV. Dataset

For this experiment, we have used the NSL-KDD dataset. The origin of this dataset is KDD Cup 1999 (KDD'99). The dataset was generated for *3rd International Knowledge Discovery and Data Mining Tools Competition* in conjunction with *KDD-99 The 5th International Conference on Knowledge Discovery and Data Mining* [25]. KDD'99 contains around 78% redundant and 75% duplicate records [26]. NSL-KDD is cleaner and more balanced than KDD'99 containing a total of 25192 instances and 41 features with no duplicates and redundant instances [27]. The Table I shortly summarizes NSL-KDD dataset.

V. Methodology

The methodology outlined in Fig. 1 were pursued throughout this experiment. First of all, the dataset was pre-processed before applying any set of rules, and the techniques of data pre-processing are explained in Section V-A. Then, feature selection strategies are explained in Section V-B. After that, Section V-C describes the classifiers and their hyperparameters. Finally, the evaluation results are discussed in Section VI.

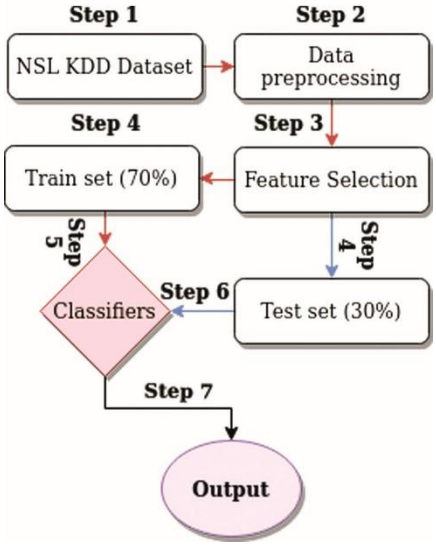

Figure 1. Methodology followed by the experiment.

A. Data Pre-processing

Though the dataset is already cleaned up by removing the duplicates, it requires more pre-processing before applying machine learning techniques. Data can not be fed to most machine learning classifiers if any of the features contain non- numeric values [9]. The *protocol type*, *service* and *flag* of NSL-KDD appeared to be non-numeric but categorical. So these non-numeric data were encoded into numeric numbers. For example, protocol contained the values *ICMP*, *TCP*, *and UDP*. After encoding, the values replaced with *0*, *1*, *2* where 0 indicates ICMP, 1 indicates TCP, and 2 represents UDP.

After that, normalization was performed using the equation (1) on all features to scale values between 0 and 1.

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

The reason behind normalization is, it inhibits the higher valued feature to dominate over the lower valued feature during the classification. If we ignore normalization, there is a high chance of misclassification by some classifiers like k-nn [9], [10].

TABLE I. Summary of the NSL-KDD Dataset Used in This Research Work

Dataset	Total instances	Anomaly	Normal	No. of features before feature selection	No. of features after feature selection
NSL KDD	25192	11743	13449	41	11
NSL KDD train set	17634	8238	9396	N/A	11
NSL KDD test set	7558	3505	4053	N/A	11

© 2022 J. Adv. Inf. Technol.

B. Feature Selection

A machine learning model requires unique and new information to perform best [9], [10]. The dataset we are working on has 41 features, but all of them might not be important or provide new information. Reducing the dimension by removing unimportant features also improves the calculation's time complexity. So the dataset was examined, applied the rules described in algorithm 1 and selected essential features for the classifiers.

Algorithm 1: The rules of selecting important features for this experiment. All the steps must be applied in serial as it is written.

1 Remove the features whose correlation with any other feature is more than 0.5
/* Correlating more than 0.5 with another feature means both of them are giving comparatively the same information, and therefore anyone can be dropped.
2 Remove features whose variance is lower than 0.5
/* variance lower than 0.5 indicates most of the instances are nearly constant and not providing much new information.

The name of selected features and their correlations are visualized in Fig. 2. It is evident from Fig. 2 that selected features contain the lowest correlation between themselves.

C. Classifiers and Their Hyperparameters

Once the most suitable features are selected for our experiment, we are ready to implement the machine learning models by training the classifiers. The dataset was split into a 70:30 ratio where 70% was used for training purposes and 30% for evaluating the classifiers. Training set and test set summaries are given in Table I.

This section, the penultimate of this research methodology, is for explaining and feeding data to the classifiers. Some classifiers require hyperparameter tuning before proceeding to the training phase. For example, a value of *k* for k-nn, *maximum depth* of tree for decision tree and *individual numbers of trees to grow* for random forest need to be chosen. The algorithm 2 illustrates how the best hyperparameter of a classifier was obtained.

In the below sections, each classifier has been explained to make us understood how they work. All the calculations and simulations were performed with the help of python's Scikit-learn [28] library.

1) K-Nearest Neighbor: K-Nearest Neighbor (k-nn) is a member of the family of instance-based learning and is also known as a lazy classifier. It is called lazy because it does all the calculations during the testing phase and remains lazy during training time [9]. In knn, the classifier calculates the distance from a test data point (also known as the query data point) to all the training set data points. Following this, it finds the *N* nearest neighbors to this test data point. Simple voting is then conducted between the *N* nearest data points to decide on the class that the classifier predicts. We need to assign the value for *N* explicitly. *Here, the number of neighbors N is considered as hyper-parameter*. Algorithm 2 was executed and found that $N = 1$ gives the best performance. The distances between neighbors were measured by using *Euclidean distance* written in equation (2).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)} \quad (2)$$

Algorithm 2: Hyperparameter tuning algorithm used in this research for k-nn, decision tree and random forest.

1 $n \leftarrow 0$
2 *hyperparameter* 1
3 **while** $n \leq 17500$ **do**
4 *Perform a 10 fold cross-validation on NSL-KDD train set*
5 *Record the average 10 fold cross-validation accuracy*
6 *Increment hyperparameter by 1*
7 *Increment n by 1*
8 *Choose the final hyperparameter which gives the highest average cross-validation accuracy (in step 5)*

2) Decision Tree (CART): Decision tree [9], [29] is a learning approach where discrete-valued target functions are approximated. Here the learned function is represented as a set of if-else/then rules to improve human readability. Instances in decision trees are classified by sorting them down the tree, starting from the root node and ending to some leaf node. Each node specifies a test of some feature, and each branch descending from that node corresponds to one of the possible values for this feature. The trees select the root node based on a statistical calculation called *information gain*.

The information gain of a node can be measured through gini index or entropy [10]. This research work used *entropy* to calculate the *information gain* using equation (3) and (4). Entropy tells us how impure a collection of data is. In other words we can say that *entropy* is the measurement of homogeneity. It returns us the information about an arbitrary dataset that how impure/non-homogeneous the dataset is [9]. The decision tree implemented in this research work is based on Classification and Regression Trees (CART) algorithm [30].

$$Entropy(S) = -\left(P_{(+)} log_2 P_{(+)} + P_{(-)} log_2 P_{(-)}\right) \quad (3)$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (4)$$

Any decision tree's characteristic is it over-fits on the training set when the depth of the tree increases. So, again using the ten-fold cross-validation and algorithm 2, it was found that, for our dataset, the decision tree performs best at *maximum depth = 1155*.

© 2022 J. Adv. Inf. Technol.

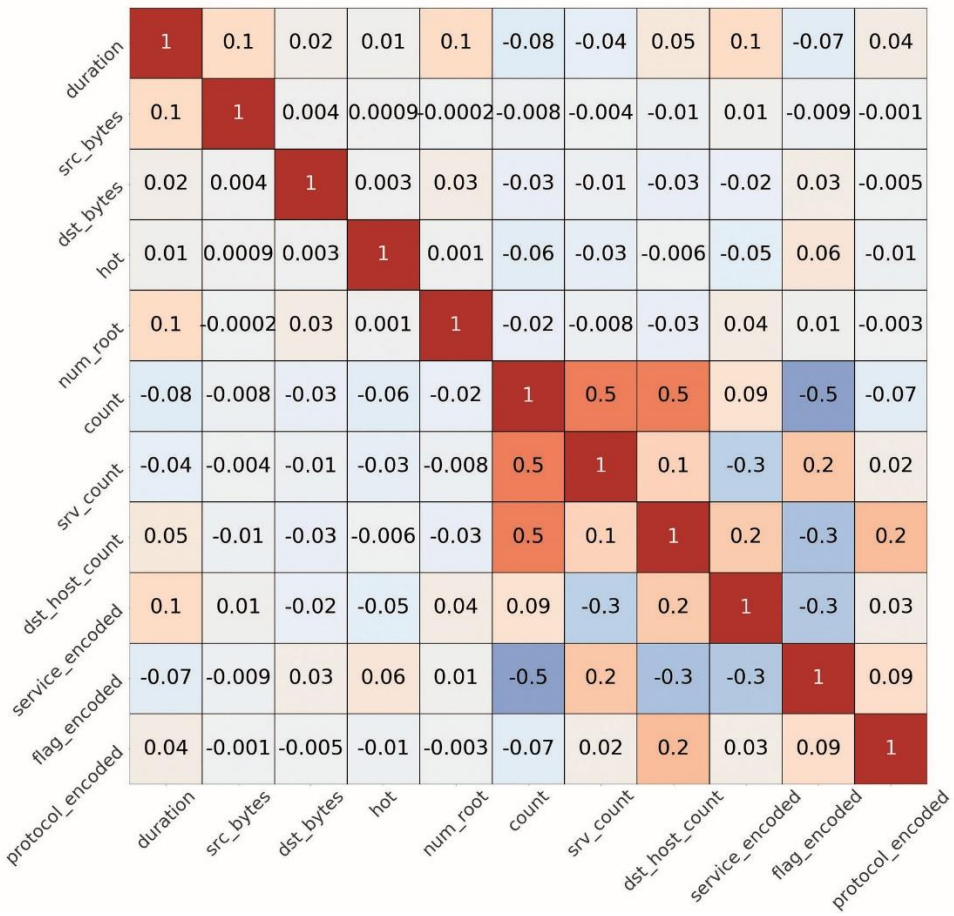

Figure 2. Correlation between selected features.

3) Random Forest (ensemble of decision trees): The random forest itself is an ensemble of decision trees, usually trained via the bagging (or sometimes pasting) method [10]. It is one of the most efficient ensemble learning methods to get more accurate and reliable predictions. The algorithm 3 describes the random forest creation process used in this experiment.

Algorithm 3: Algorithm used in this research for creating a random forest.

1 choose *T*: number of trees to grow.
/* The value for *T* or how many trees to grow is determined by algorithm 2
2 choose *m:* number of features to be used to learn each tree.
//m = √total features
3 **for** *each tree* **do**
4 prepare a training subset from the given training set via random sampling with replacement.
5 prepare a feature subset by randomly choosing *m* features
6 learn a decision tree using those selected features and training subset.
7 Use majority voting (soft voting [31]) among all the trees to predict the outcome.

By applying algorithm 2, it was determined that *1400 individual decision trees* should be grown for the random forest in order to get the best performance. All 1400 decision trees were based on CART.

4) Naïve Bayes: Naïve Bayes classifier is an efficient Bayesian learning method. In most cases, its performance has been compared with decision trees and neural networks [9]. The equation (5) illustrates naïve bayes theorem.

$$P(Y|X) = \frac{P(X_1, X_2, X_3 \dots X_n|Y)P(Y)}{P(X_1, X_2, X_3 \dots X_n)}$$
$$or, P(Y|X) = P(X_1, X_2, X_3 \dots X_n|Y)P(Y) \quad (5)$$

where, $(X_1, X_2, X_3 \dots X_n)$ are the feature values.

Naïve bayes assumes all the features are independent. This classifier does not depend on any hyperparameter. *So we did not have to worry about that.*

5) Logistic Regression: Logistic Regression is a statistical method that uses a logistic function, also known as the sigmoid function (equation (6)) to predict the probability of a certain class or event existing. It generates an S-shaped curve, displayed at Fig. 3 which can take any real-valued number and maps it into a value between 0 and 1 [32].

$$g(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

© 2022 J. Adv. Inf. Technol.

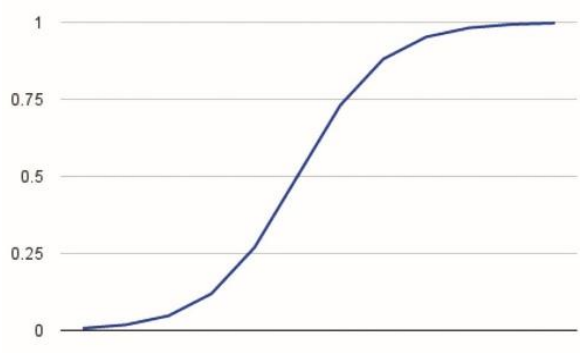

Figure 3. Visualization of logistic function which maps probabilities between 0 and 1.

Like naïve bayes, *logistic regression also doesn't have to deal with hyperparameter*.

6) Ensemble Learning or Voting Classifier: Ensemble learning is a process of combining multiple models/classifiers to solve a particular problem. Ensemble learning of this experiment takes all classifiers' prediction and makes a final prediction by considering majority voting. We combined k-nn, decision tree, naïve bayes, and logistic regression to build an ensemble model or a voting classifier in our work. The Fig. 4 visualized the ensemble learning approach. This voting classifier is also called "hard voting," where the ensemble approach uses predicted class labels for majority rule voting [31].

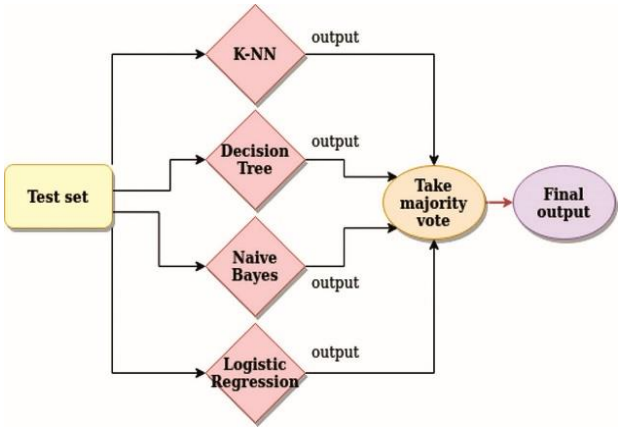

Figure 4. Ensemble learning or Voting classifier by combining the base classifiers used in this experiment.

VI. Results

In this section, the evaluation results have been exhibited to demonstrate how the classifiers have performed. Before proceeding to the results, let us understand the evaluation criterion. Section VI-A describes the equations, which were used to calculate accuracy, precision, recall, f1-Score and false alarm rate from Fig. 5. Section VI-B explains: how we should evaluate classifiers in terms of area under the ROC curve graphs. Fig. 6 represents the ROC curves obtained from this experiment. Evaluation results of each classifier are exhibited in Table II.

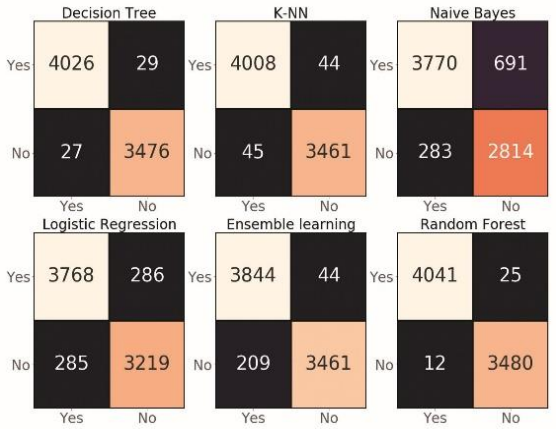

Figure 5. Comparison of confusion matrices.

A. Equations for Measuring Performance

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$F1\ Score = \frac{2 * recall * precision}{recall + precision} \quad (10)$$

$$False\ alarm\ rate = \frac{FP}{FP + TN} \quad (11)$$

where,

TP = Numbers of correctly predicted anomaly class
TN = Numbers of correctly predicted normal class
FP = Numbers of normal class predicted as anomaly
FN = Numbers of anomaly class predicted as normal

TABLE II. Comparison of Classification Reports

Classifier Name	Accuracy	Precision	Recall	F1-score	AUC	False alarm rate
K-nn	0.988	0.987	0.987	0.987	0.988	0.011
Decision Tree	0.993	0.992	0.992	0.992	0.994	0.007
Naïve Bayes	0.871	0.803	0.909	0.853	0.860	0.155
Logistic Regression	0.924	0.918	0.919	0.918	0.942	0.071
Ensemble of 4 base classifiers	0.967	0.987	0.943	0.964	N/A	0.011
Random Forest	**0.995**	**0.993**	**0.997**	**0.995**	**0.999**	**0.006**

© 2022 J. Adv. Inf. Technol.

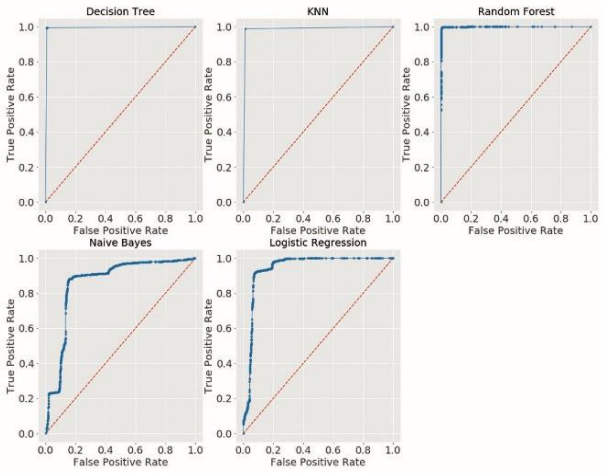

Figure 6. Comparison of ROC curves.

B. Area under the ROC Curve Explanations

ROC curve is a performance measurement for classification problems. ROC is a probability curve, and AUC represents the degree or measure of reparability. It tells how much a model is capable of distinguishing between classes. Higher the AUC defines a better model at predicting anomaly as an anomaly and normal as normal. The ROC curve is plotted with TPR (True Positive Rate) against the FPR (False Positive Rate), where TPR is on the y-axis and FPR is on the x-axis.[1]

TABLE III. COMPARISON OF PROPOSED WORK WITH EXISTING ONES

SL	Authors	Classification type	Classification accuracy %
1	Chuanlong Yin *et al.* [17]	Binary	83.28
2	S. Peddabachigar *et al.* [4]	Multi-class	92.95(Avg)
3	L.-h. Zhang *et al.* [18]	Multi-class	99.45
4	Y. Bouzida *et al.* [20]	Multi-class	92.63
5	**Proposed model**	**Binary**	**99.5**

VII. DISCUSSION

From Table II, it is visible that each classifier and even their ensemble approach performed very well. However, the random forest seems to prevail over others in terms of accuracy, precision, recall, f1-score, auc, and false alarm rate. The reason is relatively straightforward and understandable. One thousand four hundred individual decision trees constructed the random forest, and each tree contains arbitrary features and instances. So, one thousand four hundred individual and independent predictions and their majority voting made the detection more accurate as well as reliable than other ones.

The dataset is quite balanced and contains abundant instances. Besides that, each researcher worked a lot by following variant approaches on this dataset, and nearly all of them have achieved remarkable performances. As of the Table III, this particular work has provided a competitive performance compared to the existing works. The feature selection techniques and the research methodology performed very well to acquire better performance. As per the performance summary of Table II and Table III, the research question can be answered as follows:

- A competitive performance can be acquired by following a simple dimensionality reduction technique (Algorithm 1).
- Competitive performance, in this case, is acquired by an ensemble of decision trees or random forest. Not by any regular/base classifier.

VIII. CONCLUSION

An intrusion detection system is a software application or device that scans a network for abnormal or malicious activities. Attackers can apply several different approaches when attempting to infiltrate into a system. A sound intrusion detection system must understand the patterns of different types of updated intrusions. While using machine learning techniques to detect such skeptical approaches, the classifiers should be updated and trained with newly available samples. The characteristic of an efficient IDS is to detect attacks accurately with a minimum misclassification rate.

This research work showed the experiments and results of different approaches to use intrusion data methodically for detecting intrusions using machine learning techniques. The research goals mentioned in Section III-A have been achieved and answered with the support of experimental results and discussions. However, we can deduce that this research's methodology and analysis can be deployed in a network intrusion detection system to get better detection performance. Cost-sensitive learning is a type of learning that considers misclassification costs. The goal of this kind of learning is to minimize the total cost. In the future, research can be conducted to reduce the misclassification rate and increase the performance more by applying cost-sensitive learning on this methodology.

CONFLICT OF INTEREST

The authors declare no conflict of interest in the research.

AUTHOR CONTRIBUTIONS

Conceptualization, B. Pranto, H. A. Ratul and M. Rahman; data analysis I. J. Diya and M. Rahman; data analysis and investigation B. Pranto and H. A. Ratul; methodology I. J. Diya and M. Rahman; writing-draft I. J. Diya, M. Rahman and B. Pranto; writing - review and editing I. J. Diya, M. Rahman, B. Pranto and H. A. Ratul; validation, Zunayeed-Bin Zahir, B. Pranto; supervision, Zunayeed-Bin Zahir. All authors read and agreed to the published version of the manuscript.

[1] ROC curve for ensemble approach of Section V-C6 can not be generated because the voting system is 'hard voting'. Hard voting does not provide any probabilities, and that is why we will not get TPR against FPR [31]. This is why Fig. 6 demonstrates ROC curve for all classifiers excluding *Ensemble learning*. The same reason is applicable for Table II to not have AUC of *ensemble of 4 base classifiers*.

© 2022 J. Adv. Inf. Technol.

REFERENCES

[1] Internet world stats. (March 3, 2020). [Online]. Available: https://www.internetworldstats.com/stats.htm

[2] D. Milkovich. (September 23, 2019). 15 alarming cyber security facts and stats. [Online]. Available: https://www.cybint-solutions.com/cyber-security-facts-stats/

[3] R. Moskowitz. (Dec. 25, 2014). Network intrusion: Methods of attack. [Online]. Available: https://www.rsaconference.com/industry-topics/blog/network-intrusion-methods-of-attack

[4] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132, 2007.

[5] A. Spadafora. (October 23, 2019). Aws hit by major DDoS attack. [Online]. Available: https://www.techradar.com/news/aws-hit-by-major-ddos-attack

[6] Security Scorecard, "Report reveals healthcare industry lacking in basic security awareness among staff, putting entire medical infrastructure at risk," Oct. 27, 2016.

[7] I. C. C. Center. (Jul. 12, 2018). Business e-mail compromise the 12 billion dollar scam. [Online]. Available: https://www.ic3.gov/media/2018/180712.aspx

[8] A. D. Rayome, "Here's how much money a business should expect to lose if they're hit with a DDoS attack," April 17, 2018.

[9] T. M. Mitchell, *Machine Learning*, 1st ed., USA: McGraw-Hill, Inc., 1997.

[10] A. Gron, *Hands-on Machine Learning with Scikit-Learn and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed., O'Reilly Media, Inc., 2017.

[11] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-Based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18-28, 2009.

[12] J. Andress, "Chapter 10 - Network security," in *The Basics of Information Security*, second ed., J. Andress, Ed., Boston: Syngress, 2014, pp. 151-169.

[13] A. J. Hoglund, K. Hatonen, and A. S. Sorvari, "A computer host-based user anomaly detection system using the self-organizing map," in *Proc. the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 2000, pp. 411-416.

[14] P. Deshpande, S. C. Sharma, S. K. Peddoju, and S. Junaid, "Hids: A host based intrusion detection system for cloud computing environment," *International Journal of System Assurance Engineering and Management*, vol. 9, no. 3, pp. 567-576, 2018.

[15] L. Vokorokos and A. Balaz, "Host-based intrusion detection system," in *Proc. IEEE 14th International Conference on Intelligent Engineering Systems*, 2010, pp. 43-47.

[16] A. S. Abed, T. C. Clancy, and D. S. Levy, "Applying bag of system calls for anomalous behavior detection of applications in linux containers," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1-5.

[17] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017.

[18] L. H. Zhang, G. H. Zhang, L. Yu, J. Zhang, and Y. C. Bai, "Intrusion detection using rough set classification," *Journal of Zhejiang University - Science A*, vol. 5, no. 9, pp. 1076-1086, 2004.

[19] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Computer Communications*, vol. 34, no. 18, pp. 2227-2235, 2011.

[20] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, and S. Gombault, "Efficient intrusion detection using principal component analysis," in *Proc. 3rd Conference on Security and Network Architectures*, La Londe, France, 2004, pp. 381-395.

[21] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37-52, 1987.

[22] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Proc. the Twenty-Eighth Australasian Conference on Computer Science*, 2005, pp. 333-342.

[23] R. Moskovitch, S. Pluderman, I. Gus, D. Stopel, C. Feher, Y. Parmet, Y. Shahar, and Y. Elovici, "Host based intrusion detection using machine learning," in *Proc. IEEE Intelligence and Security Informatics*, 2007, pp. 107-114.

[24] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367-38384, 2018.

[25] U. K. D. in Databases Archive. (October 28, 1999). Kdd cup 1999 data. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[26] D. D. Protic, "Review of KDD cup'99, NSL-KDD and Kyoto 2006+ datasets," *Military Technical Bulletin*, vol. 66, no. 3, pp. 580-596, 2018.

[27] S. Sapre, P. Ahmadi, and K. Islam, "A robust comparison of the kddcup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," arXiv preprint arXiv:1912.13204, 2019.

[28] F. Pedregosa, *et al.*, "Scikit-Learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[29] B. Pranto. (2019). Entropy calculation, information gain decision tree learning. [Online]. Available: https://medium.com/ analytics-vidhya/entropy-calculation-information-gain-decision-tree-learning-771325d16f

[30] Scikit-learn 0.22.2. Tree algorithms. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

[31] Sklearn. Sklearn voting classifier documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

[32] A. Pant. (Jan 22, 2019). Introduction to logistic regression. [Online]. Available: https://towardsdatascience.com/introduc-tion-to-logistic-regression-66248243c148

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Md Badiuzzaman Pranto was born on 31st December 1996 in Amtali, Bangladesh. He is currently pursuing Master of Science (MSc) in Artificial Intelligence (AI) at the Friedrich-Alexander-Universität (FAU) Erlangen-Nürnberg, Germany. He received his Bachelor of Science (BSc) degree in Computer Science and Engineering (CSE) from North South University (NSU), Dhaka, Bangladesh in the year of 2019. He had been a software engineer at Software Global Consultancy, Dhaka, Bangladesh, and Strativ AV, Dhaka, Bangladesh respectively. His current research interests are in the area of Artificial Intelligence, Machine Learning, Image Processing, Natural Language Processing, and IoT.

Md Hasibul Alam Ratul has graduated from North South University, Dhaka, Bangladesh with a degree of Bachelor of Science (BSc) in Computer Science and Engineering (CSE) in the year of 2019. He has been working as a Software Engineer at Go Zayaan LTD, Dhaka, Bangladesh. His current research interests are in the area of Cloud Computing, Cyber Security, Network Security, and Cryptography.

Md Mahidur Rahman was born on 1st December 1995 in Brahmanbaria, Bangladesh. He has been graduated with Bachelor of Science (BSc) degree in Computer Science and Engineering (CSE) from North South University (NSU), Dhaka, Bangladesh in the year of 2020. He has been working as a Data Analyst in BZM Graphics Limited, Bangladesh. Which is a partner company of Pixelz Inc. His current research interests are in the area of Data Science, Data Analytics, Data Mining and IoT.

Ishrat Jahan Diya has graduated from North South University, Dhaka, Bangladesh with a degree of Bachelor of Science (BSc) in Computer Science and Engineering (CSE) in the year of 2019. She has been working as an Analyst at Quantanite, Dhaka, Bangladesh. Her current research interests are in the area of Cloud Computing, Data Analytics, and IoT.

Zunayeed-Bin Zahir is currently pursuing Doctor of Philosophy (Ph.D.) in Electrical and Computer Engineering (ECE) at the University at Albany, SUNY, New York, USA. He received his Master of Science (MSc) in Electrical Engineering from The State University of New York at Buffalo, New York, USA, and Bachelor of Science (BSc) in Electrical and Electronics Engineering (EEE) from North South University (NSU), Dhaka, Bangladesh. He is a former lecturer at NSU in the ECE department. His current research interests are in the area of Communication, Networking, and Deep Learning Optimization.

© 2022 J. Adv. Inf. Technol.