International Journal of Computer Applications Technology and Research
Volume 8–Issue 04, 107-115, 2019, ISSN:-2319–8656

A Comparative Analysis of Standard and Ensemble Classifiers on Intrusion Detection System

Joseph Mbugua
Kabarak University
Kenya

Moses Thiga
Kabarak University
Kenya

Joseph Siror
Kabarak University
Kenya

Abstract
With the increased dependence on the Internet, Network Intrusion Detection system (NIDs) becomes an indispensable part of information security system. NIDs aims at distinguishing the network traffic as either normal or abmormal. Due to the variety of network behaviors and the rapid development of attack strategies, it is necessary to build an intelligent and effective intrusion detection system with high detection rates and low false-alarm rates. One of the major developments in machine learning in the past decade is the ensemble method that generates a set of accurate and diverse classifiers that combine their outputs such that the resultant classifier outperforms all the single classifiers. In this work a comparative analysis on performance of three different ensemble methods, bagging, boosting and stacking is performed in order to determine the algorithm with high detection accuracy and low false positive rate. Three different experiments on NSL KDD data set are conducted and their performance evaluated based on accuracy, false alarms and computation time. The overall performance of the different types of classifiers used proved that ensemble machine learning classifiers outperformed the single classifiers with high detection accuracy and low false rates.

Keywords Ensemble classifiers, intrusion detection, standard classifiers, false alarms

i. INTRODUCTION

Classification [1] algorithms takes instances from a dataset and assigns a class or category to each of them based on supervised learning techniques. The technique can be applied in intrusion detection system to classify the network data as normal or attack. Several researchers have developed models to evaluated machine classifiers to categorise intrusion data set such as Support Vector Machines (SVM), Bayesian belief networking, Artificial Neural Network (ANN). Despite the variety and number of proposed models the construction of a perfect classifier for any given task remains challenging. The main challenge related to machine learning techniques is that no single-classification technique is capable of detecting all classes of attacks within acceptable false alarm rates and detection accuracies [2]–[4].

The ensembles learning models [5]combines multiple and homogeneous, weak classifiers to solve advanced and complex problems and improve the classification accuracy of the final results. These models apply the same algorithm repeatedly through partitioning and weighting of a training data set and improves classification performance by the combined use of two effects i.e. reduction of errors due to bias and variance [6]. Adaptive hybrid systems have become essential in computational intelligence and soft computing, the main reason being the high complementary of its components. The integration of the basic technologies into hybrid machine learning solutions facilitates more intelligent search and reasoning methods that match various domain knowledge with empirical data to solve advanced and complex problems. Implementation of ensemble and combination of multiple predictions are mainly motivated by three aspects which characterize the intrusion detection domain [7]–[9]: (i) relevant information may be present at multiple abstraction levels, (ii) the information

may be collected from multiple sources, and (iii) this information needs to be represented at the human level of understanding.

The remainder of this paper is organized as follows. First, some background on ensemble classifiers is given. Then, we present the proposed methodology, experiments and results Finally, we draw conclusions and future work.

ii. ENSEMBLE CLASSIFICATION SYSTEM

The strategy in ensemble classification systems is to create a set of accurate and diverse classifiers in order to combine their outputs such that the combination outperforms all the single classifiers. A classifier is accurate when its classification error is lower than that obtained when the classes are randomly assigned. Two classifiers are diverse if they make errors at different instances. classifier ensembles are built in two phases: generation and combination.

Generation phase: In the generation phase, the individual components of the ensemble, known as base classifiers, are generated. The techniques used to generate diverse classifiers are based on the idea that the hypothesis of a classifier depends on both the learning algorithm and the subset used to generate these classifiers. Three different approaches can be used to generate an ensemble of classifiers by varying the training set. (i) Resampling the training examples by bagging and boosting to construct the classifier ensemble, (ii) Manipulating the input features achieves diversity between classifiers by modifying the set of attributes used to describe the instances, and (iii) Manipulating the output target to generate a pool of diverse classifiers with each classifier solving a different classification problem. The category that solves multiclass problems by converting them into several binary subproblems falls in class no iii. Methods that vary the learning algorithm can be subdivided in two groups i.e. (i) approaches that use different versions of the same learning algorithm (homogeneous ensembles) and (ii) approaches where diversity is obtained using different learning algorithms (heterogeneous classifiers).

Combination Phase: In the combination phase, the decisions made by the members of the ensemble are combined to obtain one decision. There are two main strategies for combining classifiers i.e. selection and fusion.

(i) Classifier selection presupposes that each classifier is an expert in some local region of the space. Therefore, when an instance is submitted for classification, the ensemble decision coincides with the decision given by the classifier responsible for the region of the space to which the instance belongs.

(ii) In classifier fusion, the decisions from all members of the ensemble are combined in some manner to make the ensemble decision. Classifier fusion algorithms include combining rules, such as the average, majority vote, weighted majority vote, and the Borda Count, and more complex integration models, such as meta-classifiers. A meta-classifier is a second-level classifier generated from the outputs given by the base learners.

Methodology

Anomaly detection using Standard classifier

Support Vector Machine, (SVM) is a machine learning algorithms used for classification, regression and outlier detection. It is one of the most accurate and robust algorithms for classification and widely used in IDS as they provide high security and take less time to detect attacks [5], [10]. The major features of SVM according to [7] include:Deals with very large data sets efficiently, Multiclass classification can be done with any number of class labels, High dimensional data in both sparse and dense formats are supported, Expensive computing not required and can be applied in many applications like e-commerce, text classification, bioinformatics, banking and other areas. Even though SVMs are limited to making binary classifications, their superior properties of fast training,

scalability and generalization capability give them an advantage in the intrusion detection application. Finding cost-efficient ways to speed up or parallelize the multiple runs of SVMs to make multi-class identification is also under investigation.

Random Forest is an ensemble learning technique for classification and predictive modeling. It is also an approach to data exploration and generates many trees by using recursive partitioning then aggregate the results [11]. Each of the trees is constructed separately by using a bootstrap sample of the data and the bagging technique[12] is applied to combine all results from each of the trees in the forest. The method used to combine the results can be as simple as predicting the class obtained from the highest number of trees.

J48 [11] is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. C4.5 is a program that creates a decision tree based on a set of labeled input data. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. J48 classifier algorithms [13] are used to compare and build, using the information entropy process, a decision tree from a set of training dataset. These algorithms adopt a top down technique and inductively built the decision tree for classification. It's extremely efficient when handling large datasets. [14]. The extra features of J48 [15] includes accounting for missing values, decision trees pruning, continuous attribute value ranges and derivation of rules.To make actual decisions regarding which path of the tree to replace is based on the error rates used. The reserved portion can be used as test data for the decision tree to overcome potential overfitting problem (reduced-error pruning).

Instance based learners (IBL) are computationally simple and represent knowledge in the form of specific cases or experiences [16] . IBL rely on efficient matching methods to retrieve stored cases so they can be applied in novel situations. Instance based learners are also called lazy learners because learning is delayed until classification time, as most of the power resides in the matching scheme. IB1 [10] is an implementation of the k nearest neighbour based classifier where k is the number of neighbors. IB1 finds the stored instance closest according to a Euclidean distance metric to the instance to be classified and the new instance is assigned to the retrieved instance's class.

Bayesian reasoning provides a probabilistic approach for inference and is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data [17]. A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, bayesian networks have several advantages for data analysis [18]. First, the Bayesian networks encode the interdependencies between variables and hence they can handle situations where data are missing. Secondly, the Bayesian networks have the ability to represent causal relationships. Therefore, they can be used to predict the consequences of an action. Lastly, the Bayesian networks have both causal and probabilistic relationships; they can be used to model problems where there is a need to combine prior knowledge with data.

iii. ENSEMBLES CLASSIFIERS

Bootstrap is a meta learning algorithm that improves classification and regression models in terms of stability and classification accuracy. The algorithm takes bootstraps samples of objects and the classifiers are trained on each sample. The classifier votes are then combined by majority voting. A bootstrap sample is a statistical sample taken uniformly and with replacement, this means that the result sample set will contain duplicates [19]. Given a training dataset of size N, Bagging creates M base models, each trained on a bootstrap sample of size N created by drawing random samples with replacement from the original training set.

Boosting [20] is a machine learning meta-algorithm that built ensemble classifier by incrementally adding and iteratively learning weak classifiers with respect to a dataset to a final

strong classifier [21]. Bagging is better than boosting as boosting suffers from over fitting as it performs well only for the training data. While both can significantly improve accuracy in comparison to a single model, boosting tends to achieve greater accuracy[22]. However unlike bagging, boosting may also reduce the bias of the learning algorithm [9]

Stacking [11] or Stacked Generalization is a different technique of combining multiple classifiers. Unlike bagging and boosting that use a voting system, stacking is used to combine different learning algorithms e.g. decision tree, neural network, rule induction, naïve Bayes, logistic regression, etc. to generate the ensemble of classifiers. The Stacking produces an ensemble of classifiers in which, the base classifiers (level-0) are built using different training parameters. The outputs of each of the models are collected to create a new dataset which is related to the real value that it is supposed to predict. Then, the stacking model learner as (level-1) use the output from base classifier to provide the final output.

One of the issues in Stacking is obtaining the appropriate combination of base-level classifiers and the meta-classifier, especially in relation to each specific dataset. If only a small number of classifiers and algorithms will be used, this problem can be solved by a simple method, namely, exhaustive search, in a reasonable amount of time. However, it is difficult to determine the best Stacking configuration when the search space is large.

iv. Experiments and Results

Experiment Environment - Waikato Environment for Knowledge Analysis

Several data mining techniques which includes data cleaning and pre-processing, clustering, classification, regression, visualization and feature selection have been implemented in WEKA (Waikato Environment for Knowledge Analysis) [23]. Weka also offers some functionality that other tools do not, such as the ability to run up to six classifiers on all datasets, handling multi-class datasets which other tools continue to struggle with tools.

Experiment Data set - NSL-KDD dataset

The NSL-KDD dataset (Jain & Rana, 2016; Parsaei et al., 2016; Shahadat et al., 2017) which is derived from original KDD-99 and has eliminated some of its drawbacks is analysed. The simulated attacks in the NSL-KDD dataset fall in one of the following four categories Denial of service attack (Dos), Probe attacks, Remote-to-Local (R2L) attacks, and User-to-Root (U2R) attacks.

Figure 1 Flow Diagram of Main Steps in the Research Study

Experiment Setup

In the experiment, we apply full dataset as training set and 10-fold cross validation for the testing purposes. The available dataset is randomly subdivided into 10 equal disjoint subsets and one of them is used as the test set and the remaining sets are used for building the classifier. In this process, the test subset is used to calculate the output accuracy while the N_1 subset is used as a test subset and to find the accuracy for each subset. The process is repeated until each subset is used as test set once and to compute the output accuracy of each subset. The final accuracy of the system is computed based on the accuracy of the entire 10 disjoint subsets.

All experiments are performed using Windows platform with the following configuration Intel Core-i5 processor, 2.5GHz speed, and 8GB RAM.

Experiment 1

The experiment is conducted with two ensemble learning techniques, bagging and boosting and five classifier using 10-fold cross validation. The single classifier includes Bayes Net, IBK, Jrip and SVM and the results are illustrated in TABLE 1. The conducted experiments is evaluated according to four performance measures which are accuracy, false positive and execution time.

Experiment II

In this experiment, the researcher compaire eight different algorithms and SVM as a base learners and stacking as a multi classifier learner are used. We use various combinations of BayesNet, iBK, ANN, J48 and JRip. The classifications predicted by the base learners will be used as input variables into a stacking model learner. Each input classifier computes predicted classifications using cross validation from which overall performance characteristics can be computed. Then the stacking model learner will attempt to learn from the data how to combine the predictions from the different models to achieve maximum classification accuracy. The stacking algorithm experiment results are given in the Table 2

Experiment III

Intrusion detection performance using combination of four distinct classifiers based on stacking with SVM as a meta classifier and the results are illustrated in Table 3**.**

The conducted experiments will be evaluated according to four performance measures which are defined below:

i. The Classification Accuracy: is the percentage number of correctly classified instances (the number of correct predictions from all predictions made)
ii. Precision: is a measure of classifier exactness (used as a measure for the search effectiveness)
iii. Recall: is a measure of classifier completeness.
iv. F-Measure: also called F-Score, it conveys the balance between the precision and the recall.
v. Experiments' Results and Data Analysis

Experiment 1

Table 1: The performance of bagging and boosting with five classifier using 10-fold cross validation

Algorithm	Accuracy	False Positive	Execution time

	Single	Bagging	Boosting	Single	Bagging	Boosting	Bag	Boost
Bayes Net	95.7%	95.5%	99.3%	4.3%	4.5%	0.670%	6.8	6.5
IBK	99.2%	99.1%	99.3%	0.80%	0.90%	0.7%	0.25	6.5
Jrip	99.5%	99.5%	99.5%	0.5%	0.5%	0.5%	395	390
J48	99.5%	99.5%	99.6%	0.5%	0.5%	0.4%	29.7	6.47
SVM	95.4%	90.53%	90.6%	4.6%	9.4%	9.47%	1656.8	1643.8

Overall, all the algorithms achieved good results, with the highest accuracy being 99.6% and the lowest being 89.59%. Tables 3 show that Adaboost when implement with J48 as a weak classifier achieves the highest accuracy, which is 99.6%, with a false positive (FP) rate of 0.30%. On the other hand, the BayesNet Bagging algorithm achieves the highest FP rate of 4.5%. Unfortunately the computation time of the three ensemble classifiers are all very high; the slowest one is stacking followed in turn by boosting and bagging.

Table 1 show that the use of the bagging and boosting algorithms did not improve the accuracy significantly. Only the use of boosting on the BayesNet algorithm were able to improve the accuracy, by 3.6% respectively, while the others showed a less than 1% improvement. While the two ensemble algorithms failed to improve the accuracy, they succeed in reducing the false positive rates. Bagging was able to reduce the false positive rate by up to 0.1% and 0.02% when implemented with IBK and BayesNet respectively, boosting was able to reduce the false positive rate by up to 3.7% and 0.02% when implemented for Naïve Bayes and J48.

Many researchers compared the performance of bagging and boosting methods including some large-scale experiments [5], [22], [28]–[30]The overall agreement is that boosting reaches lower testing error have been crowned as the most accurate available off-the-shelf classifiers on a wide variety of datasets . Nonetheless, it is observed that boosting methods are sensitive to noise and outliers, especially for small datasets [31]. Bagging is effective with noisy data whereas boosting is quite sensitive to noise. Another benefit of bagging methods is that they are parallel in nature in both the training and classification phases, whereas the boosting method is sequential in nature.

Experiment II

Table 2: The performance of SVM as a base learners and stacking as a multi classifier learner with eight classifier using 10-fold cross validation

Stacking meta classifier	TP	FP	Precision	F measure	Execution time (sec)
SVM	96.4	0.029	96.1	96.1	762.33
SVM With Bayesian	98.9	0.7	98.9	98.9	21.8
SVM With RF	99.8	0.1	99.8	99.8	340.63
SVM With J48	99.8	0.1	99.8	99.8	40.1
SVM With boost	90.53	9.47	90.53	90.53	1643.8
SVM With bagging	90.6	9.4	90.6	90.6	1656.8
SVM With ANN	93.6	6.4	93.5	93.7	1057.8
SVM With IBK	95.7	4.3	95.7	95.7	2147.1
SVM With Jrip	97.1	2.79	97.1	97.1	985
SVM With oneR	91.77	8.23	91.77	91.77	876

Bayesian is a highly scalable classifier and performs well for classifying rough dataset like medical data. For NSL-KDD which is a preprocessed data set, Bayes Net is providing approximately the same results for accuracy, precision and recall of 98.9%. While its false positive rate i.e. in correctively classified instances is 0.7%. The time to build the model is moderate at 21.8 seconds.

Bagging and boosting Are ensemble machine learning algorithms used in integration with another classifier to improve its prediction power but when integrated with SVM its performance degrades. This is because SVM is a strong classifier, and as described by Khorshid et al., (2015) integrating SVM with Bagging or Boosting does not improve its performance. Similar results are found in our evaluation result as accuracy of both Ensemble with SVM is 90.53% and 90.6% respectively while the accuracy of individual SVM is 91.81%.

The JRip and OneR are both based on association rule mining. The JRip is a fast and ripper algorithm and OneR creates one rule for each attribute and then picks up rule with least error [32], [33]. Hence combining SVM with JRip gives high accuracy and sensitivity values i.e. 97.21% and 92.33% while with OneR, it provides accuracy and sensitivity values as 91.77% and 79.14%.

ANN [21] is a strong classifier and it is also a weak learner and it requires large data set to train classifier. Because of this stacking, both these algorithms do not give very good performance. It gives better performance than SVM as its accuracy, specificity, precision and recall is more than SVM.

The K - Nearest Neighbour (IBK) is a lazy trainer. This means it stores the training instances and do real work only at the time of classification [26]. and hence IBK gives strongly consistent results. However, equal weightage is given to each of the attributes. As a result, combining this algorithm with SVM gives moderately high rate of accuracy of. 95.79%. but slow in execution speed of 2147.1 sec.

The decision tree algorithms J48 and Random Forest provides high accuracy rate of 99.8 and a false positive rate of 0.1% and execution time of 40.1 seconds and 340.63 seconds respectively. Random Forest is giving the best performance in all evaluating parameters. NSL-kDD data set does not contain redundant records and it is easy for these classifiers to build their decision tree output and as a result combining them with SVM improves the overall performance of intrusion detection system.

Experiment III

Table 3. Intrusion detection performance using combination of four distinct classifiers based on stacking

Staking	TP	FP	PREC	RECAL	FMEASURE	ROC	TIME
	99.8	0.1	99.8	99.8	99.8	99.9	4757.87

CONCLUSION AND FUTURE WORK

Overall the application of bagging and boosting did not significantly improve the accuracy or reduce error rates. Only Stacking method was able to reduce the false positive rate by a moderately high amount. The key assumption that the errors due to the individual models are uncorrelated is unrealistic; in practice, the errors are typically highly correlated, so the reduction in overall error is generally small [28], [34]. Staking method took the longest execution time which is a drawback in its application in the intrusion detection field.

Although a lot of research on AI-based ensembles has been conducted, several research questions still remain unanswered, for example, how many base classifiers should be combined, how base classifiers should be combined, how to generate diverse set of base classifiers, how instances of training dataset should be partitioned to generate base classifiers, how feature space should be partitioned and in particular for ID quality training dataset among others.

REFERENCES

[1] S. Thaseen and C. A. Kumar, “Intrusion Detection Model using PCA and Ensemble of Classifiers,” vol. 16, no. 2, pp. 15–38, 2016.

[2] M. Albayati and B. Issac, "Analysis of Intelligent Classifiers and Enhancing the Detection Accuracy for Intrusion Detection System," *Int. J. Comput. Intell. Syst.*, vol. 8, no. 5, pp. 841–853, 2015.

[3] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[4] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Eng.*, vol. 30, no. 2011, pp. 1–9, 2012.

[5] M. M. H. Khorshid, T. H. M. Abou-el-enien, and G. M. A. Soliman, "A Comparison among Support Vector Machine and other Machine Learning Classification Algorithms A Comparison among Support Vector Machine and other Machine Learning Classification," no. August, 2015.

[6] M. Govindarajan, "Hybrid Intrusion Detection Using Ensemble of Classification Methods," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 2, pp. 45–53, 2014.

[7] S. Thaseen and A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, 2017.

[8] I. Czarnowski and J. Piotr, "An Approach to Machine Classification Based on Stacked Generalization and Instance Selection," 2016.

[9] G. Kumar and K. Kumar, "The Use of Multi-Objective Genetic Algorithm Based Approach to Create Ensemble of ANN for Intrusion Detection," *Int. J. Intell. Sci.*, vol. 2, no. October, pp. 115–127, 2012.

[10] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, and M. C. Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection A Comparative Analysis of SVM and its Stacking with other Classification Algorithm for Intrusion Detection," no. November, 2017.

[11] R. Pradhan, "Performance Assessment of Robust Ensemble Model for Intrusion Detection using Decision Tree Techniques," vol. 3, no. 3, pp. 78–86, 2014.

[12] S. L. Pundir and Amrita, "Feature Selection Using Random Forest in Intrusion Detection," *Int. J. Adv. Eng. Technol.*, vol. 6, no. 3, pp. 1319–1324, 2013.

[13] J. H. Assi and A. T. Sadiq, "NSL-KDD dataset Classification Using Five Classification Methods and Three Feature Selection Strategies," vol. 7, no. 1, pp. 15–28, 2017.

[14] M. K. Gambo and A. Yasin, "Hybrid Approach for Intrusion Detection Model Using Combination of K-Means Clustering Algorithm and Random Forest Classification," *Ijes*, vol. 6, no. 1, pp. 93–97, 2017.

[15] Dubb Shruti and Sood Yamini, "Feature Selection Approach for Intrusion Detection System," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 2, no. 5, pp. 47–53, 2013.

[16] S. V. Lakshmi, T. E. Prabakaran, and D. Ph, "Application of k-Nearest Neighbour Classification Method for Intrusion Detection in Network Data," vol. 97, no. 7, pp. 34–37, 2014.

[17] A. A. Ramaki, M. Khosravi-Farmad, and A. G. Bafghi, "Real time alert correlation and prediction using Bayesian networks," *2015 12th Int. Iran. Soc. Cryptol. Conf. Inf. Secur. Cryptol.*, vol. 978, pp. 98–103, 2015.

[18] R. Kaur and M. Sachdeva, "International Journal of Advanced Research in An Empirical Analysis of Classification Approaches for Feature Selection in Intrusion Detection," vol. 6, no. 9, 2016.

[19] S. Kovac, "Suitability analysis of data mining tools and methods," p. 53, 2012.

[20] S. S. Dongre and K. K. Wankhade, "Section V with concluding conclusion in Section VI. II. data mining for IDS," vol. 2, no. 4, pp. 488–492, 2012.

[21] M. Amini, "Effective Intrusion Detection with a Neural Network Ensemble Using Fuzzy Clustering and Stacking Combination Method," vol. 1, no. 4, pp. 293–305, 2014.

[22] I. Journal, F. Technological, A. Patel, and R. Tiwari, "Bagging Ensemble Technique for Intrusion Detection," vol. 2, no. 4, pp. 256–259, 2014.

[23] M. Revathi, "Network Intrusion Detection System Using Reduced," *J. Comput. Sci.*, vol. 2, no. 1, pp. 61–67, 2000.

[24] N. Shahadat, I. Hossain, A. Rohman, and N. Matin, "Experimental Analysis of Data Mining Application for Intrusion Detection with Feature reduction," pp. 209–216, 2017.

[25] A. Thesis, "Using Support Vector Machines in Anomaly Intrusion Detection by," 2015.

[26] A. Jain and J. L. Rana, "Classifier Selection Models for Intrusion Detection System (Ids)," *Informatics Eng. an Int. J.*, vol. 4, no. 1, pp. 1–11, 2016.

[27] M. R. Parsaei, S. M. Rostami, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," vol. 7, no. 6, pp. 20–25, 2016.

[28] M. P. Sesmero, A. I. Ledezma, and A. Sanchis, "Generating ensembles of heterogeneous classifiers using," vol. 5, no. February, pp. 21–34, 2015.

[29] A. Shrivastava, M. Baghel, and H. Gupta, "A Review of Intrusion Detection Technique by Soft Computing and Data Mining Approach," no. 3, pp. 224–228, 2013.

[30] C. Science, "A Hybrid Approach to improve the Anomaly Detection Rate Using Data Mining Techniques," no. July, 2015.

[31] Y. Wahba, E. ElSalamouny, and G. ElTaweel, "Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction," *Ijcsi*, vol. 12, no. 3, pp. 255–262, 2015.

[32] J. Hussain and S. Lalmuanawma, "Feature Analysis, Evaluation and Comparisons of Classification Algorithms Based on Noisy Intrusion Dataset," *Procedia Comput. Sci.*, v

[33] J. Song, "Feature Selection for Intrusion Detection System Jingping Song Declaration and Statement," p. 132, 2016.

[34] M. Govindarajan, "Evaluation of Ensemble Classifiers for Intrusion Detection," vol. 10, no. 6, pp. 1045–1053, 2016.