Intrusion Detection Using Error Correcting Output Code Based Ensemble

Shaza Merghani AbdElrahman and Ajith Abraham[1,2]
Faculty of Computer Science & Information Technology, Sudan University of Science and Technology, Khartoum, Sudan
seza_mer@yahoo.com
[1]Machine Intelligence Research Labs (MIR Labs),Scientific Network for Innovation and Research Excellence, WA, USA
[2]IT4Innovations - Center of excellence, VSB - Technical University of Ostrava, Czech Republic
ajith.abraham@ieee.org

Abstract-**Intrusion Detection System is an essential part in computer security. Researchers have proposed many methods but most of them suffer from low detection rates and high false alarm rates. In this paper, we try to tackle the class imbalance problem, increase detection rates for each class and minimize false alarms in intrusion detection system. We test the performance of seven classifiers using Bagging and AdaBoost ensemble methods. We proposed a new hybrid ensemble for intrusion detection based on Error Correcting Output Code (ECOC) approach.**

Keywords-Intrusion Detection, ensemble, Error Correcting Output Code (ECOC)

I. INTRODUCTION

Intrusion Detection Systems is an essential part in computer security. It aims to protect the network and computer system from any intrusions that compromise the integrity, confidentiality, or availability of resources[1]. It must be able to identify all suspicious patterns and abnormal traffic by monitoring, detecting and respond to the unauthorized system activities [2], [3]. Therefore, the functions of intrusion system include monitoring and analyzing both user and system activities, analyzing system vulnerability, assessing file and system integrity, ability to recognize attacks and abnormal activities, and tracking user policy violations [4].

There are two intrusion detection systems based on the used detection strategy: Signature/Misuse and Anomaly based intrusion detection. In Signature/Misuse intrusion detection, alarms generated based on specific attack or activity signatures of intrusive activities. This involves the monitoring of network traffic in search of direct matches to known patterns of attack (signatures). The advantage of misuse detection is it's high accuracy for all known attack however, it detect only patterns that follow a predefined intrusions and undetected those aren't trained on them. Anomaly based intrusion detection, which uncovers user/system behavior patterns that are far and have some deviations from normal. Anomaly detection has ability for detecting novel or unknown attacks however; it suffers from high false alarm rate and low detection rate [5] [6].

In this paper, to tackle the class's imbalance problem, increase detection rates for each class and minimize false alarms in intrusion detection systems, we have tested the performance of seven classifiers using Bagging and AdaBoost ensemble methods and proposed new hybrid ensemble. This ensemble based on Error Correcting Output Code (ECOC)approach, which is one of the multiclass binary classification methods.

II. RELATED WORKS

Several approaches and many data mining techniques have been investigated for design an ensemble for Intrusion detection system such as Neural Network (NN), Support Vector Machines (SVM), Decision trees (DTs) and Random Forest, etc.

Mukkamala et al. [7] introduced an ensemble that combining soft and hard computing techniques. They combined three classifiers: Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and Multivariate Adaptive Regression Splines (MARS). Their results showed that the ensemble of classifiers outperformed the individual ones. To obtain more accurate intrusion detection system, Peddabachigari et al. [8] proposed a hierarchical hybrid system by integrating different learning models and an ensemble approach combined SVM and DTs as base classifiers.

Chebrolu et al. [6] identified the important features using Markov blanket model and decision tree and introduced a hybrid architecture ensemble model for intrusion detection. They combined Bayesian Network (BN) and Classification and Regression Tree (CART) as base classifiers. An ensemble that combined Support vector machine, artificial neural network and random forest as base classifier was introduced in [9]. Their results showed that ensemble gives better results. To improve the performance, Hu et al. [10] proposed an intrusion detection system based on AdaBoost ensemble technique. They used a decision stump as weak classifiers and built an ensemble from the weak classifiers for both continuous and categorical features.Dongre and Wankhade [11]used boosting technique with adaptive sliding window and hoeffing tree.

Three-layer hierarchy ensemble was developed by Chou et al. [12] to solve uncertainty and increase diversity between classifiers. The ensemble involved three groups of classifiers each of them uses different sub set of

978-1-4799-7633-1/14/$31.00 2014 IEEE

features and different machine learning algorithms. They used Back propagation network (BPNN), fuzzy KNN and Naïve Bays as base classifiers. Govindarajan and Chandrasekaran [13]proposed an ensemble model using arcing (adaptive re-sampling and combining) technique with RBF, SVM and hybrid RBF-SVM.Recently, Chaurasia and Jain [14] enhanced the detection rate by combining neural networks (NN) and K- nearest neighbor (KNN) with bagging ensemble approach. Govindarajan [15] proposed homogenous and heterogeneous ensemble models. The homogenous ensemble used bagging technique with RBF and SVM. And the other heterogeneous ensemble used arcing technique with hybrid models of SVM-RBF. He concluded that, ensemble approach improves the accuracy and heterogeneous models gives better results than the homogenous ones.

III. Methods Used

A. *Meta learning ensemble Methods*

Ensemble is a combination of multiple classifiers so as to improve the generalization ability and increase the prediction accuracy. The most popular combining techniques are the meta-learning methods such as boosting and bagging (Bootstrap Aggregating).

In boosting, each classifier is dependent on the previous one, and focuses on the previous one’s errors. Examples that are misclassified in previous classifiers are chosen more often or weighted more heavily. Whereas, in bagging, each model in the ensemble votes with equal weight. Bagging promotes model variance and improves the accuracy of one model by using multiple copies of the model. It trains each model in the ensemble using a randomly drawn subset of the training set, average of misclassification errors on different data splits gives a better estimate of the predictive ability of a learning method [16].

B. *Multi-class Binary Classification Methods*

The basic idea of multi-class binary classification is decomposing multi-class problem into a set of binary problems. The ensemble is made of binary classifiers and then the output of ensemble is outputs combinations of those classifiers as multi-class classifier. There are several approaches introduced for representing multi class into binary class depending on the decomposition or integration approach such as one-against-all (OAA), One-against-one (OAO) and error correcting output code (ECOC) [17].

One-against-all (OAA):

In this approach, for k class problem, create k binary classifiers that distinguish each class from all others. For each instance, run all classifiers, return classifier with highest score.

One-against-one (OAO):

In OAO approach (also known as All-pairs), for k class problem, Create k(k-1)/2 classifiers that distinguish each pair of classes. For each instance, run all classifiers, return classifier with majority voting.

Error correcting output code (ECOC):

In this approach, for k class's problem, each class assigned a binary string of length k (codeword). Each bit position in the string corresponds to output of specific classifier. For each instance, apply each classifier to compute new codeword and return the class with closest codeword.

IV. Experiments and Analysis

A. *Dataset Description*

In this paper, we used dataset related to intrusion detection prepared by MIT Lincoln Lab [1]. The data set consists of 41 attributes and one class label. 24 attack types classified into four main classes: Dos (Denial of Service), R2L (Unauthorized Access from a Remote Machine), U2R (Unauthorized Access to Local Super User (root)) and Probing. The data is highly imbalance, the training set divided as follows: 1000 are normal data, 1000 are probe, 3002 are Dos, 27 are U2R and 563 are U2L.Table 1explains the number of samples in training and testing sets per class.

Table 1.IDS Data set

Class	Training Set	Testing Set
Class 1 (normal)	1000	1400
Class 2 (probe)	500	700
Class 3 (Dos)	3002	4202
Class 4 (U2R)	27	25
Class 5 (U2L)	563	563
Total	5092	6890

B. *Experimental Results*

Firstly, in our experiments we implemented multi class classification and tested the performance of seven selected classifiers (Naïve Bays (NB), Multi Layer Perceptron (MLP), Support Vector Machine (SVM), Radial Basis Function Network (RBF), J48, Random Tree (RT), and Random Forest (RF)) and compare their result when applied homogenous ensemble methods such as Bagging and AdaBoost. The obtained results depicted in figure 1 in term of accuracy. However, as we dealing with imbalance class problem, the overall accuracy biased to the majority class regardless the minority class with lower samples, which leads to poor performance on the minority class. This is clearly illustrated in Figure 2 where all high detection rates (True positive rates) are for the majority classes (class 3, class 1) using different classifiers. This also true for the other measures such as precision, f-measure and ROC.

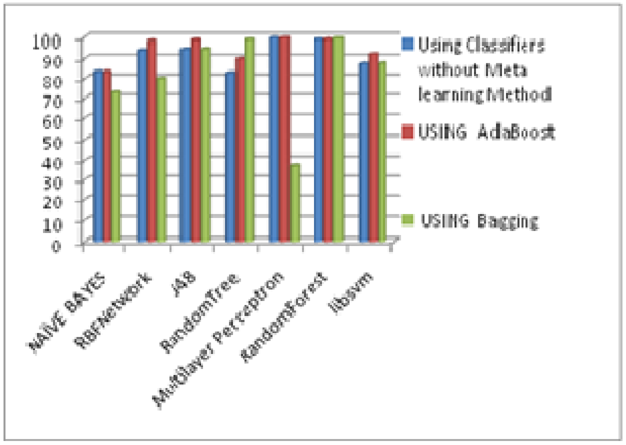


Figure 1. The performance of classifiers interm of accuracy using multi class classification

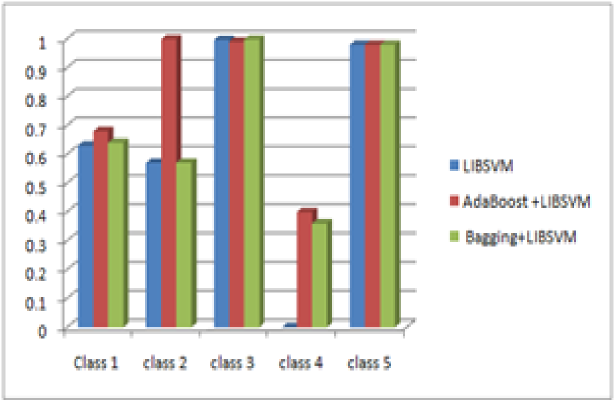
From these results, we can conclude the following: In terms of accuracy (as depicted in Figure 1):

- Multi layer perceptron has the best performance using the classifier alone or within AdaBoost ensemble.
- Random Forest (RF) has the second highest accuracy and its performance always stable up to 98.5% with all experiments.
- There is observed improvement in performance using Bagging with Random Tree.
- The worst result obtained when applying Bagging with Multi layer perceptron, which is a negative result for ensemble caused by over fitting.

In term of detection rates (as depicted in Figure2):

- All classifiers have rather good detection rates for class 3, class 1, class 2 and class 5, which are the majority classes (classes by large numbers of samples) respectively, except SVM which has a lowest detection rate for class 1 and MLP have a clear degradation in performance with all classes when using it with bagging.
- All classifiers have rather bad detection rates for class 4, which is a class with lowest number of samples. The highest detection rate for class 4 obtained by MLP/AdaBoost(64%). The lowest detection rate for class 4 obtained by SVM and when using bagging with MLP. Which indicatethat these methods are very sensitive for class imbalance or for the minority classes (classes by the lowest numbers of samples).

Figure 2a. Detection rate for each class using SVM and Meta learningMethods



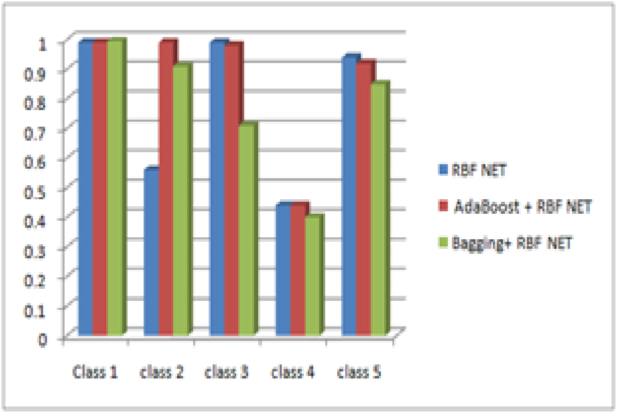


Figure 2b. Detection rate for each class using RBF and Meta learningMethods

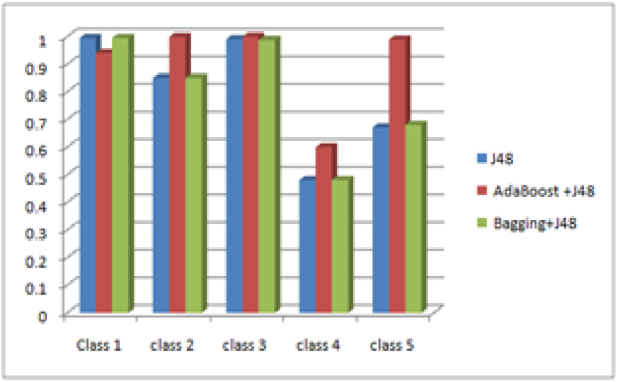


Figure 2c. Detection rate for each class using J48 and Meta learning Methods

Figure 2d. Detection rate for each class using Random Forest and Meta learningMethods

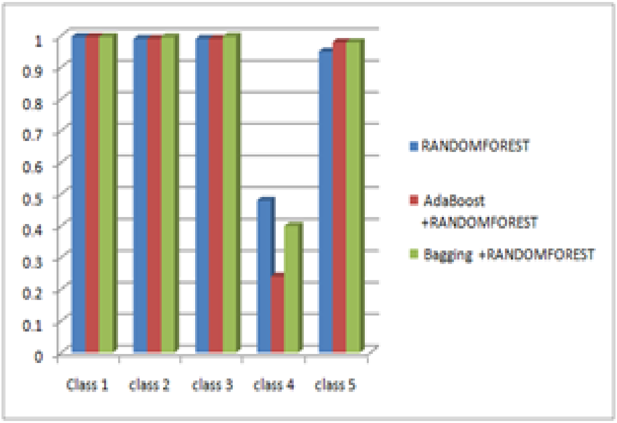


Figure 2d. Detection rate for each class using Random Forest and Meta learning Methods

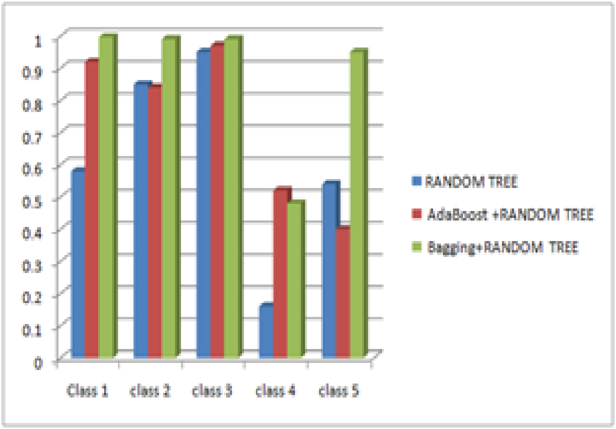


Figure 2e. Detection rate for each class using Random Tree and Meta learning Methods

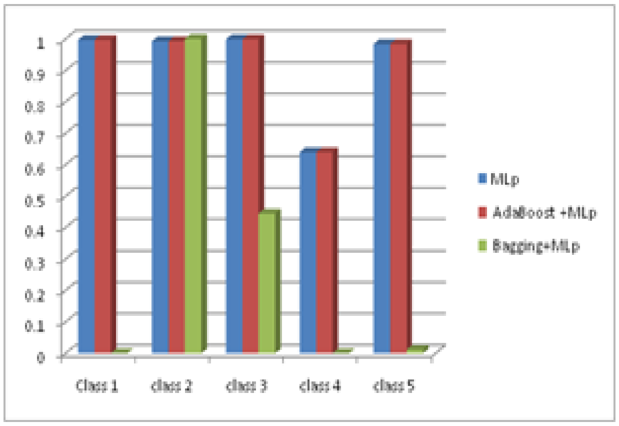


Figure 2f. Detection rate for each class using MLP and Meta learning Methods

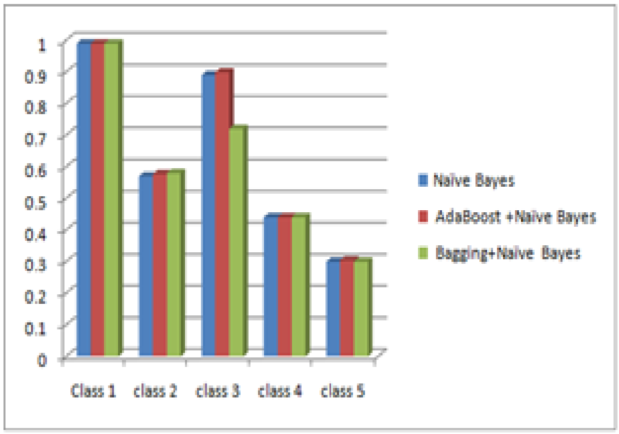


Figure 2g. Detection rate for each class using Na ïve Bayes and Meta learning Methods

Secondly, to tackle the class's imbalance problem, increase detection rates for each class and minimize false alarms, we suggest an ensemble model based on Error-Correct Output Codes. In which, the multiclass problem decomposes into several binary sub-problems, and trains a standard classifier for each class. The constructed model must distinguish the samples of a single class (positive class) from all samples in remaining classes (negative class).

Also here, we tested the performance of seven selected classifiers (Na ïve Bays (NB), Multi Layer Perceptron (MLP), Support Vector Machine (SVM), Radial Basis Function Network (RBF), J48, Random Tree (RT), Random Forest (RF)) and compare their result when applied homogenous ensemble methods such as Bagging and AdaBoost for the five classes. Figure 3 depicts the false positive rates for all classifiers per class.

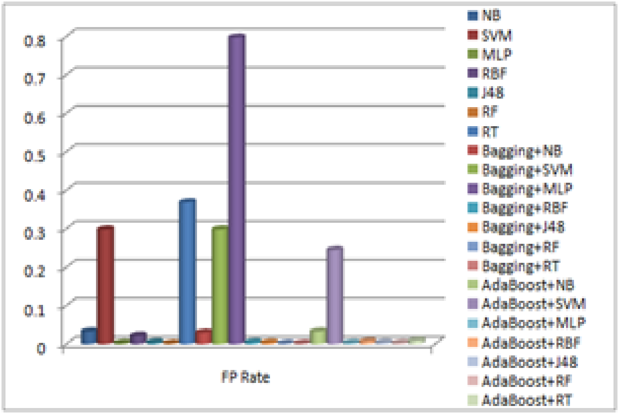


Figure 3a. False Positive rates for class 1

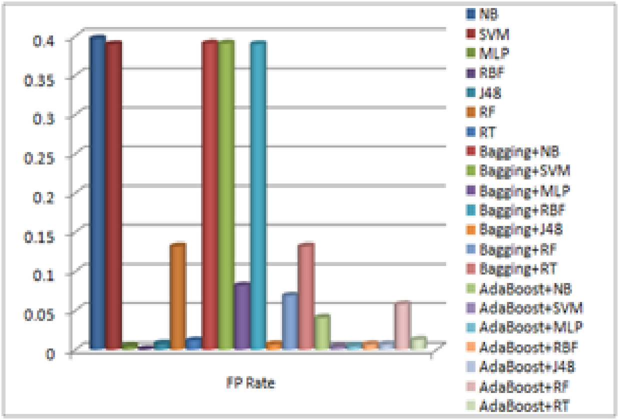


Figure 3b. False Positive rates for class 2

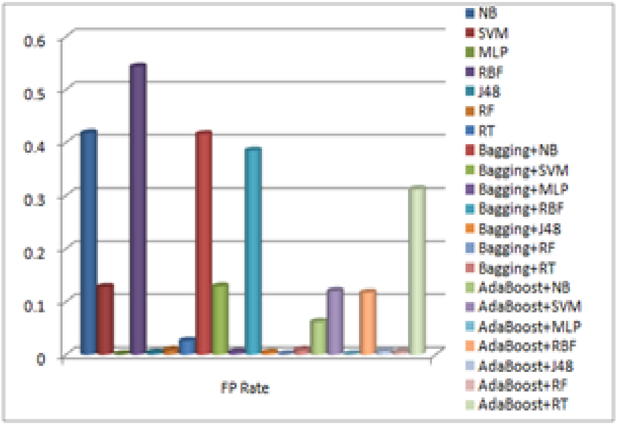


Figure 3c. False Positive rates for class 3

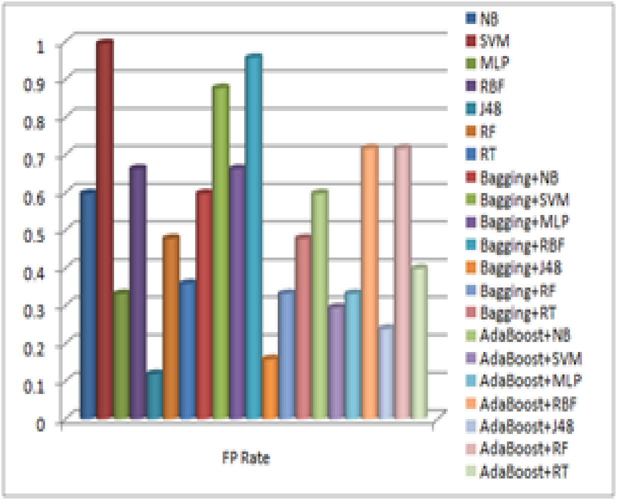


Figure 3d. False Positive rates for class 4

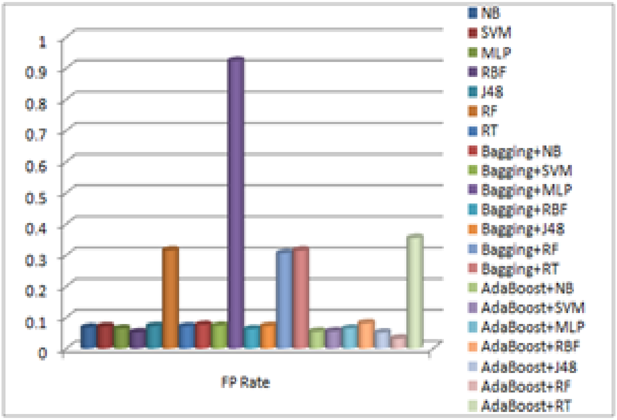


Figure 3e. False Positive rates for class 5

Then, the best classifier for each class has been selected to build the proposed ensemble. In coding phase, each class assigned a code word of length 5, where each bit in a code word corresponding to output of one classifier. In decoding phase, the weighted hamming distance function has been used. It works by measures the differences between the output and code words and then assigned the class with the closest codeword (has lowest difference) and highest weight.

Table 2. The performance of the ensemble

Class	*Detection rate (TP rate)*
Class 1	0.997
Class 2	0.994
Class 3	0.999
Class 4	0.92
Class 5	0.97
Accuracy	0.997

The best results for the ensemble obtained when using bagging with RF for class 1, MLP for class 2, MLP for class 3, J48 for class 4 and adaboost with RF for class 5 (the best classifiers with low false alarms for each class as depicted in figure 3). The ensemble performance outperforms the individual classifiers and used meta learning methods. It improves the accuracy and clearly reduces false alarm rates even for those minorities' classes. The performance of the ensemble depicted in Table 2.

V. CONCLUSION

In this paper, we have investigated the performance of seven selected classifiers (Naïve Bays, Multi Layer Perceptron, Support Vector Machine, Radial Basis Function Neural Network, J48, Random Tree and Random Forest and compare their results when applied them into homogenous ensemble using metalearning methods such as Bagging and AdaBoost. We have presented new hybrid ensemble model based on Error-Correct Output Codes for intrusion detection. The experimental results show that, the proposed ensemble improves the overall accuracy (99.7%). Also, it increases detection rates and reduces false alarm rates even for those minorities' classes.

References

[1] C. Grosan, C. Martin-Vide and A. Abraham, "Evolutionary Design of Intrusion Detection Programs," *International Journal of Network Security*, vol. 4, pp. 328-339, 2007.

[2] N. Padhy, R. Pantigrahiand P. Mishra, "The survey of data mining applications and feature selection," *Asian Journal of Computer Science And Information Technology*, vol. 2, pp. 68-77, 2012.

[3] D. Pahuja and R. Yadav, "Outlier Detection for different application: Review," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, 2013.

[4] S. Gore A. S. Ashoor, "Importance of Intrusion Detection System (IDS)," *International Journal of Scientific & Engineering Research*, vol. 2, no. 2229-5518, pp. 1-4, January 2011.

[5] K. K.R. and A. Indra, "Intrusion Detection Tools and Techniques A survey," *International Journal of Computer Theory and Engineering*, vol. 2, pp. 901-906, 2010.

[6] S. Chebrolu, A. Abraham and J. P. Thomas, "Feature Deduction and ensemble Design of Intrusion Detection Systems," *Computers and Security*, 2004.

[7] S. Mukkamala, A. H. Sung and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *Journal of Network and Computer Applications*, 2004.

[8] S. Peddabachigaria , A. Abraham, C. Grosanc and J. Thomas,"Modeling intrusion detection system using hybrid intelligent systems," Journal of Network and Computer Applications, 2005.

[9] Riyad.A.M and M.S Irfan Ahmed, "An Ensemble Classification Approach for Intrusion Detection," *International Journal of Computer Applications*, vol. 80, pp. 37-42, October 2013.

[10] W. Hu and S. Maybank, "AdaBoost-Based Algorithm for Network Intrusion detection," *IEEE Transactions on Systems Man and Cybernetics*—Part B: Cybernetics, vol. 38, no. 2, pp. 577-583, 2008.

[11] [S. S. Dongre and K. K. Wankade, "Intrusion Detection System Using New Ensemble Boosting Approach," *International Journal of Modeling and Optimization*, vol. 2, pp. 488-492, August 2012.

[12] T. Chou, J. Fan, S. Fan, and K. Makki, "Ensemble of Machine Learning Algorithm for Intrusion Detection," IEEE, 2009.

[13] M. Govindarjan and RM. Chandrasekaran, "Intrusion Detection using an Ensemble of Classification Methods," in World Congress on Engineering and Computer Science, vol. 1, San Fracisco, USA, 2012.

[14] S. Chaurasia and A. Jain, "Ensemble Neural Network and K-NN Classifiers for Intrusion Detection," *International Journal of Computer Science and Information Technology*, vol. 5, pp. 2481-2485, 2014.

[15] M. Govidarajan, "Hybrid Intrusion Detection Using Ensemble of Classification Methods," *I.J.Computer Network and Information Security*, vol. 2, pp. 45-53, 2014.

[16] A. Fernando, E. Barrenechea, H. Business, F. Herrera and M. Galar, "A Review on ensembles for the class Imbalance Problem," *IEEE Transactions on Systems Man and Cybernetics: Part C: Applications and Reviews*, vol. 42, 2012.

[17] T. H. Oong and N. A. M. Isa, "One-against-all ensemble for multiclass pattern classification," *Applied Soft Computing*, vol. 12, pp. 1303-1308, 2012.