

Ensemble Voting System for Anomaly Based Network Intrusion Detection

Mrutyunjaya Panda¹ and Manas Ranjan Patra²

¹ Department of ECE, Gandhi Institute of Engineering and Technology, Gunupur, Orissa-765022, India

Email: mrutyunjaya.2007@rediffmail.com

² Department of Computer Science, Berhampur University, Orissa-760007, India

Email: mrpatra12@gmail.com

Abstract— The growing dependence of modern society on telecommunication and information networks has become inevitable. Therefore, the security aspects of such networks play a strategic role in ensuring protection of data against misuse. Intrusion Detection systems (IDS) are meant to detect intruders who elude the “first line” protection. Data mining techniques are being used for building effective IDS. In this paper we analyze the performance of some data classifiers in a heterogeneous environment using voting ensemble system with the purpose of detecting anomaly based network intrusions. Experimental results using KDDCup 1999 benchmark dataset demonstrate that the voting ensemble technique yield significantly better results in detecting intrusions when compared to other techniques.

Index Terms— Intrusion Detection, Ensemble Learning, Voting Ensemble, ROC

I. INTRODUCTION

Computer networks are usually protected against attacks by Intrusion Detection Systems (IDS). The traditional prevention techniques such as user authentication, data encryption, avoidance of programming errors, and firewalls are only used as the first line of defense. But, if a password is weak and is compromised, user authentication cannot prevent unauthorized use. Similarly, firewalls are vulnerable to errors in configuration and sometimes have ambiguous/undefined security policies. They fail to protect against malicious mobile code, insider attacks and unsecured modems. Therefore, intrusion detection is required as an additional wall for protecting systems [1].

Intrusion detection attempts to detect computer attacks by examining various process data on the network. It is split into two categories, anomaly detection systems and misuse detection systems. Anomaly detection is an attempt to search for malicious behaviour that deviates from established normal patterns. Misuse detection is used to identify intrusions that match known attack scenarios. In this paper, we propose a scalable solution for detecting anomaly based network intrusion.

One of the most active areas of research in supervised learning has been to study methods for constructing good ensemble of classifiers. It has been observed that when certain classifiers are ensembled, the performance is phenomenal compared to the performance of the individual classifiers. Here, we propose a voting ensemble classifier algorithm which is tested and results are compared with other ensemble machine learning algorithms, including AdaBoost, MultiBoost, and Decorate with various base learner algorithms like J48, SMO, Rule Learner. KDDCup 1999 benchmark dataset is used for the experimentation and the results show that the proposed algorithm is promising and greatly outperforms existing methods, achieving high detection rate with low false alarm rate and more importantly take less time to build the model.

The outline of the paper is as follows. A review of the state-of-the-art on Intrusion Detection systems (IDSs) is given in Section 2. Section 3 introduces technical analysis of the various machine learning approaches. The proposed method is presented in Section 4. Various base learners used in this paper are explained in Section 5. The evaluation on the proposed algorithm is carried out in Section 6 by comparing it with other ensembles. The paper is concluded in Section 7.

II. RELATED WORK

In [2], the authors have proposed various feature reduction techniques like Principal component analysis (PCA), Linear Discriminate Analysis (LDA) and Independent Component Analysis (ICA) in order to build an efficient network intrusion detection model in terms of detection accuracy and computation time. PCA and ICA feature extraction approaches with Pareto-Optimal optimization is used in [3] to obtain a high performance intrusion detection system. The authors show that their proposed system outperforms standard SVM, PCA SVM and ICA SVM. In [4], the authors have proposed support vector machines (SVM) and neural networks (NN) for intrusion detection. An evolutionary support vector machine for intrusion detection is proposed

in [5]. In this, the authors have combined evolutionary programming into support vector machines. They concluded that their model is able to detect new attacks as well as experienced attacks. In [6], ensemble learning with various base learning algorithms for detecting rare attacks is proposed. In [7], the authors have proposed Bayesian approach in intrusion detection system. It consists of building a reference model and the use of a Bayesian classification procedure associated to unsupervised learning algorithm to evaluate the deviation between current and reference behaviour. The authors have evaluated various machines learning algorithm for detecting network intrusions in [8]. A novel ensemble of classifiers for micro array data classification is done in [9]. They used this method using Particle swarm optimization (PSO) and EDAs (Estimation of Distribution Algorithms) on four benchmark datasets to produce the best recognition rates. In [10], the authors have modeled an intrusion detection system using hybrid intelligent systems using Decision Trees and SVMs in order to maximize detection accuracy and minimize computational complexity.

III. TECHNICAL ANALYSIS

A. Ensemble learning


Ensembles of classifiers can perform better than any individual classifier; this performance advantage can be attributed to three key factors [11].

The learning procedure for ensemble algorithms can be divided into the following parts.

Constructing base classifiers/base models:

The main tasks at this stage are:

- *Data processing:* prepare the input training data for building base classifiers by perturbing the original training data, and
- *Base classifier constructions:* build base classifiers on the perturbed data with a learning algorithm as the base learner. In this work, we have used SMO, J48, and ZeroR as the base learners.

 *Voting:* The second stage of an ensemble method is to combine the base models built in the previous step into a final ensemble model. There are different types of voting systems, the frequently used ones are: weighted voting and un-weighted voting. In the weighted voting system, each base classifier holds different voting power. On the other hand, in the un-weighted system, individual base classifier has equal weight, and the winner is the one with most number of votes.

B. Boosting and Decorate

Boosting algorithms are a class of algorithms that have been mathematically proven to improve upon

the performance of their base models in certain situations.

AdaBoost has performed very well in practice and is one of the few theoretically motivated algorithms that have turned into a practical algorithm. However, AdaBoost can perform poorly when the training data is noisy, i.e. the inputs and outputs have been randomly contaminated [11]. Noisy examples are normally difficult to learn. More details can be found in [12], [13].

MultiBoosting

It is another method of the same category that can be considered as wagging committees formed by AdaBoost [14]. Wagging is a variant of bagging; bagging uses re-sampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses re-weighting for each training example, pursuing the effect of bagging in a different way.

Decorate

A new meta-learner DECORATE (Diverse Ensemble Creation by Oppositional Re-labeling of Artificial Training Examples) reported in [15] uses an existing “strong” learner (one that provides high accuracy on the training data) to build a diverse committee.

IV. PROPOSED METHODOLOGY

In this section, first we discuss on the disadvantages of the existing ensembles and then present the proposed approach which utilizes their positive aspects while subduing their weaknesses.

A. Analysis of Weakness of Existing Ensembles

The accuracy of boosting models remain the same after specific numbers of base models are established, because of the checking mechanism after each construction of base classifiers. The specific criterion in boosting stops further construction while its error rate equals to 0 or greater than 0.5. Therefore, if the sequential construction halts after building 6 base classifiers, same result will be obtained on evaluating over boosting models with any number greater than 6, because fundamentally these models are all identical. As our intrusion detection dataset consist of large amounts of intrusion classes, it helps the learning methods to generate a more precise classifier that fits exactly on the training dataset. The checking criterion seriously influences the diversity of boosted models by forbidding further construction of base models.

It was thought that using base learners can lead to a better performance, but in [16], “many could be better than all” theorem indicates that this may not be the fact. It was shown that after generating a set

of base learners, selecting some base learners instead of using all of them to compose an ensemble is a better choice.

It is also worth noting that the computational cost for building an ensemble comprising T base learners is roughly T times the cost of training a single learner. So, from the computational complexity point of view, training an ensemble is almost as efficient as training a single classifier.

B. Design of Voting Ensemble system

In this paper, we have used the un-weighted majority voting for detecting network intrusions.

Combining classifiers with this method is simple; it does not require any previous knowledge of the behaviour of the classifiers nor does it require any complex methodology to decide. It only counts the number of classifiers that agree in their decision and accordingly decides the class to which the input pattern belongs. This simplicity has a drawback, however; the weight of the decision of all the classifiers is equal, even when some of the classifiers are much more accurate than others. The voting ensemble is illustrated in Fig. 1.

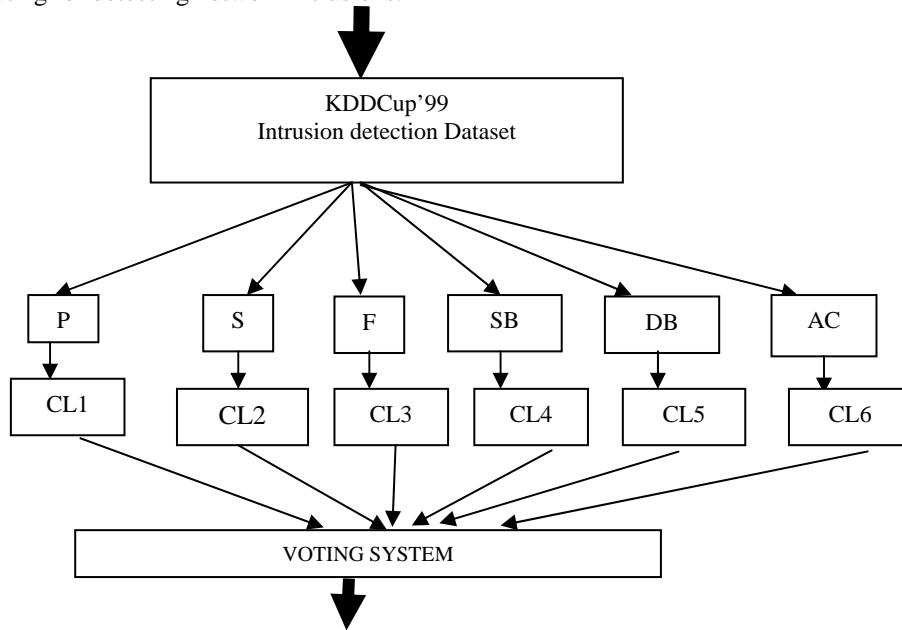


Figure 1. The framework for the Ensemble of Voting System with six features.

Where, CL_i refers to the i^{th} classifier, P for Protocol, S for Service, F for Flag, SB for Source Byte, DB for Destination Byte and AC for Attack Class.

- a. *Feature Selection:* In this proposed framework, six features were extracted from the full dataset. The features are normalized to [0, 1].
- b. *Classification:* Three Classifiers namely AdaBoost, MultiBoost, Decorate Meta classifiers were used with J48, SMO, ZeroR as base learners. The details are given in the next section.
- c. *Voting System:* Different results will be obtained from the different ensemble of classifiers by using different features extracted from the KDDCup'99 intrusion detection dataset, and then these results are put into the voting system. Each classifier has a weight to denote the contributions of the classifier to the voting system. For each class to be identified, a weighted sum of base learners can be calculated as:

$$V_i = \sum_{d=1}^N \alpha * w_d, \alpha = \begin{cases} 1, C_d = i \\ 0, otherwise \end{cases} \quad (1)$$

Where N is the number of classifiers, $i=1, 2... C$ is the class label, C_d is the predicted class label by the d classifier, and w_d is the weight of the d classifier. For a given unknown pattern, the final class to be classified is determined by maximizing $\arg \max_{j=1}^C V_j$.

V. BASE LEARNERS USED IN OUR EXPERIMENTATION

As mentioned above, we plan to design an ensemble machine learning classifier to address the network intrusion detection problem. The critical factor to achieve this goal is the selection of classifiers, since it is vital to achieve high accuracy from ensemble techniques. We have used Decision Trees (J48), SMO (Sequential Minimal Optimization), Rule Learning (ZeroR)

specifically for this problem. This process is illustrated in Fig.2.

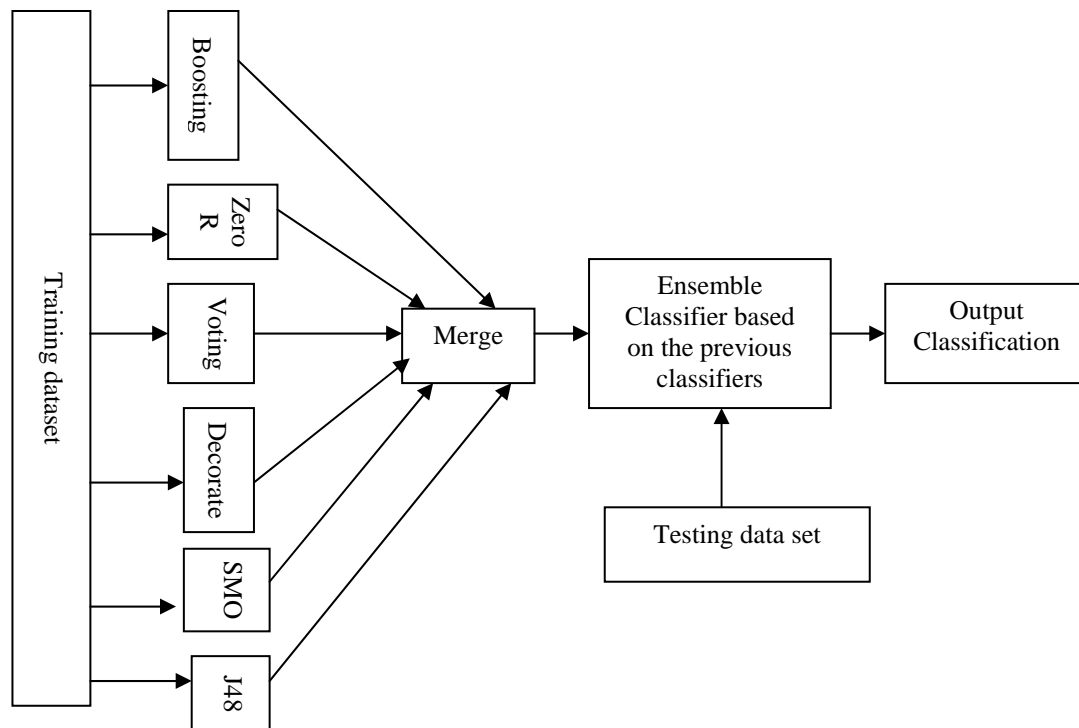


Figure 2. Proposed Ensemble process with various base learners used

VI. EXPERIMENTAL SETUP AND EVALUATION OF RESULTS

In this work, we have used five class classification methods to build our intrusion detection model. We have used a randomly selected subset of KDDCup'99 intrusion detection dataset, which contains 1000 instances from each class proportional to size, except that the smallest class is completely included. Full dataset is used for training and 10 fold cross validation for testing purpose. We have carried out our experiment on a Pentium 4 CPU 2.8GHz with 512MB RAM.

It can be observed from Table 1 that average accuracy of our proposed voting ensembles with ZeroR provides the best results in comparison to other ensemble classifiers already available. It can also be observed that it is very fast which takes only 0.03 second to build the network intrusion detection model.

We have also compared our five class classification results individually with other approaches used by different authors in building an efficient intrusion detection model in Table 2. In that, we could see that our proposed voting ensembles Voting+J48+ZeroR and Voting+AdaBoost+SMO are amongst the best to detect the normal classes. While detecting DoS

and U2R attacks, our method is the best in comparison to Hybrid DT+SVM, SOM IDS,

LAMSTAR IDS, SVM +Rocchio Bundling and SVM+DGSOT. However, the proposed methods provide better accuracy in detecting Probe and R2L attacks in comparison to Hybrid DT+SVM, SMO, AdaBoost+SMO, LAMSTAR IDS and SOM IDS. It is also proposed to compare the systems performance in terms of Receiver operating characteristics (ROC), which is a plot between detection rates (DR) with false positive rate (FPR), which is shown in Fig. 3. Compared to other measurements, ROC provides a visual tool for examining the trade off between the ability of a classifier to correctly identify positive cases that are incorrectly classified. At the same time, other evaluation metrics like root mean square error (RMSE), false negative rate (FNR) and F-Score are evaluated for different ensemble classifiers in Fig. 4. From, all these comparisons, it is imperative that our proposed voting ensemble classifiers with AdaBoost+48 and J48+ZeroR rule learner performs well in building an efficient network intrusion detection model.

Table.1. Performance Comparison of various Classifiers

Classifier	Avg. Accuracy (%)	Build Time (Seconds)
SVM [3]	95.56	Not Provided
PCA+SVM [3]	96.54	Not Provided
ICA+SVM [3]	87.14	Not Provided
PARETO OPTIMAL [3]	96.56	Not Provided
SMO [17]	75.97	1962.25
SVM Light [17]	88.55	24.10
ISVM [17]	88.12	26.0
Tree SVM [17]	85.99	28.2
Array SVM [17]	90.78	45.0
AdaBoost + SMO	96.63	230.52
MultiBoost + SMO	95.66	367.58
Vote+AdaBoost+J48 (ours)	97.38	0.28
Vote+AdaBoost+SMO(ours)	91.47	315.52
Vote+J48+ZeroR(ours)	97.47	0.03
Vote+MultiBoost+J48(ours)	96.97	0.63
Vote+SMO(ours)	96.97	42.06
Vote+Decorate+J48(ours)	97.28	2.86
SVM+Rochilo Bundling [18]	51.6	26.7
Clustering Tree +SVM [18]	69.8	13.18
ESVM [5]	96.4	Not Provided
ESVM with bootstrap [5]	95.6	Not Provided
Logistic [5]	89.8	Not Provided
Sigmoid [5]	80.4	Not Provided

VII. CONCLUSION

In this paper, after investigating voting ensemble of classifiers, it is observed that classifiers based on voting+J48+Rule learner and voting+AdaBoost+J48 performs efficiently in terms of high detection rate, low false positive rate, less time taken to build the model, high F-score, reasonably low RMSE in comparison to existing ensemble classifiers, at the expense of some what high false negative rate.

A serious deficiency of ensemble methods is its lack of comprehensibility, i.e., the knowledge learned by ensembles is not understandable to the user. Improving the comprehensibility of ensembles is an important yet largely understudied direction. Exploration of methodology towards this will be our future research direction.

REFERENCES

[1] Denning D., "An Intrusion Detection model", IEEE transaction on S/W Engineering, Vol.8E-13, No.2, pp.222-232, 1987.
 [2] V.Venkatchalam and S.Selvan, "Performance comparison of intrusion detection system classification using various feature reduction techniques", International journal of simulation, Vol.9, no.1, ISSN-1473-8031(print), ISSN-1473-804x (online).
 [3] Yu, Gu, Bo Zhou and Jiashu Zhao, "PCA-ICA ensembled intrusion detection system by Pareto-

optimal optimisation", Information Technology Journal, Vol.7, No.3, pp.510-515, 2008. ISSN-1812-5638.
 [4] Srinivas Mukkamala, Guadalupe Janoski, and Andrew Sung, "Intrusion detection using support vector machines and neural networks", in Proc. Of IEEE International conference on Neural Networks, IEEE Computer society Press, pp.1702-1707, 2002.
 [5] Sung-Hae Jun and Kyung-Whan Oh, "An Evolutionary support vector machine for intrusion detection", Asian journal Information Technology, Vol.5, No.7, pp.778-783, 2006.
 [6] M.Panda and M.R.Patra, "Ensemble learning for detection of rare attacks", in Proc. Of International conference on advances in computer, communication and control, India, pp.510-515, 2009, ACM Press, USA. ISBN: 978-1-60558-315-8.
 [7] M.Mehdi, S.Zair, A.Anou and M.Bensebti, "A Bayesian networks on intrusion detection system", Journal of computer Science, Vol.3, No.5, pp.259-263, 2007. ISSN: 1549-3636.
 [8] M.Panda and M.R.Patra, "Evaluating machine learning algorithms in detecting network intrusions", International Journal of Recent Trends in Engineering", Vol.1, No.1, pp.472-477, 2009. Academy Publisher, Finland.
 [9] Yuehui Chen, Yaou Zhao, "A novel ensemble of classifiers for micro array data classification", Applied soft computing, No.8, pp.1664-1669, 2008. Elsevier.
 [10] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan, and Johnson Thomas, "Modelling intrusion detection system using hybrid intelligent systems", Journal of network and computer applications, Vol.30, no.1, pp.114-132, 2007. Elsevier.
 [11] Dietterich, T.G., "Ensemble methods in machine learning", Lecture notes in computer science (LNCS), Vol.1857, pp.1-15, 2000.
 [12] N.C.Oza, "AdaBoost2: Boosting with noisy data", In F.Roli, J.Kittler, and T.windeatt (Edns.), Proc. Of the 5th international workshop on multiple classifier systems, pp.31-40, 2004. Springer-Verlag, Berlin.
 [13] Ratsch,G., Onoda, T., and Muller,K.R., "Soft margins for AdaBoost", machine Learning, Vol.42, pp.287-320, 2001.
 [14] Webb,G.I., "MultyiBoosting: A technique for combining boosting and wagging", Machine Learning, Vol.40,pp.159-196,2000.
 [15] P.Melville and R.J.Mooney, "Constructing diverse classifier ensembles using artificial training examples", in Proc. Of IJCAI, Acapulco, Mexico, pp.505-510, 2003.
 [16] Z-H Zhou, J.Wu and W.Tang, "Ensembling neural networks: many could be better than all"Artificial intelligence, Vol.137, No.1-2, pp.239-263, 2002.
 [17] John Mill and Atsushi Inoue, "support vector classifiers network intrusion detection", in Proc. Of 2004 IEEE international conference on fuzzy systems, WA, USA, Vol.1, pp.407-410, 2004. ISSN: 1098-7584.
 [18] Latifur Khan, Momouri Awad and Bhavani Thurasingham, "A new intrusion detection

system using SVM and hierarchical clustering”, the VLDB journal, Vol.16, pp.507-521, 2007. DOI-10.1007/s00778-006-0002-5.

[19] M. Panda and M.R.Patra, “Anomaly based network intrusion detection using boosting

support vector classifiers”, In Proc. Of 2009 IEEE international advance computing conference (IACC-09)”, India, 2009, pp.926-931. ISBN: 978-981-08-246-5. IEEE, USA.

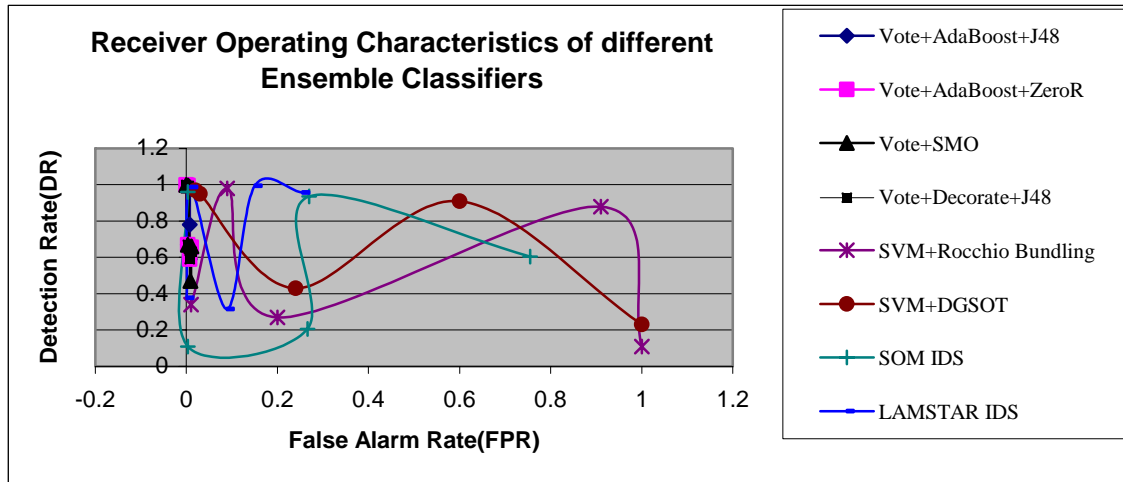


Figure 3. ROC comparison among different classifier ensembles

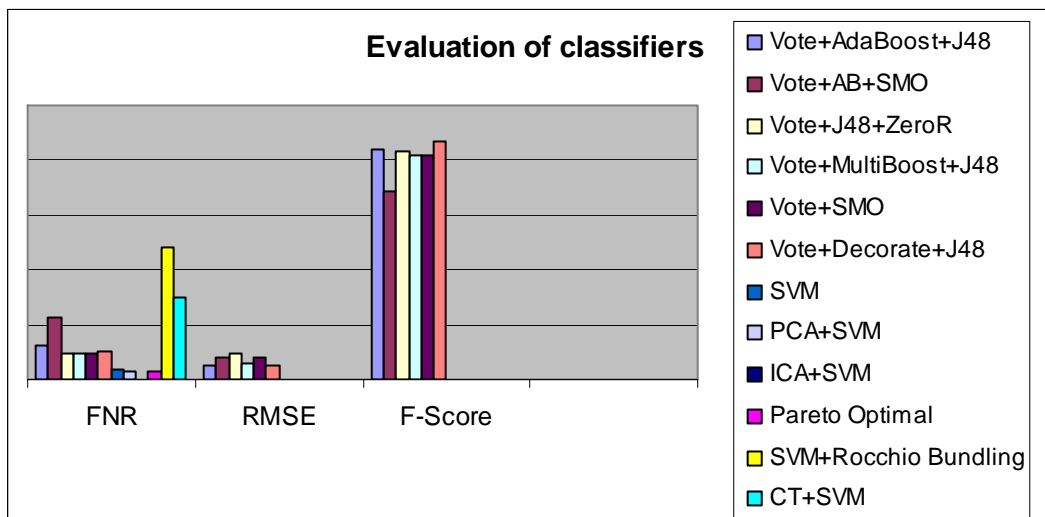


Figure 4. Comparison of Evaluation metrics for different Ensemble Classifiers

Table.2. Performance Evaluation of different Ensemble of Classifiers

	Hybrid DT+SVM [10]	SOM [19]	AB+SOM [19]	SVM + Rocchio [18]	SVM + DGSOT [18]	SOM IDS [2]	LAMSTAR IDS [2]	Vote + AB+J48 (ours)	Vote+J48+ZeroR (ours)	Vote + AB+SOM (ours)	Vote +Decorate+J48 (ours)
Normal	99.7	97.4	97.8	98.0	95.0	93.6	99.4	99.0	99.6	99.6	99.3
Probe	98.57	66.2	71.0	34.0	97.0	60.5	95.6	78.1	65.6	66.0	63.0
DoS	99.92	100	99.0	11.0	23.0	95.9	98.6	100.0	100.0	100.0	100.0
U2R	48.0	54.3	67.0	27.0	43.0	20.6	31.6	67.0	67.0	67.0	67.0
R2L	37.8	38.1	30.0	88.0	91.0	10.9	37.7	59.0	59.0	47.0	59.0