International Journal of Engineering Research and General Science Volume 3, Issue 3, Part-2 , May-June, 2015
ISSN 2091-2730

COMPARISON OF STACKING AND boost SVM METHOD FOR KDD DATASET

NilufarZaman D.P. Gaikwad

AISSMS College of Engineering , Pune

nilufar.zaman@mescoepune.org dp.g@rediffmail.com

Abstract: The method of combining classifier can be done in various ways of which the most competent of them are Stacking and Voting method. Stacking is a way of combining multiple generalizers one after another where the output of the first classifier is considered as an input to the next one, whereas Voting method works on the principle of best result oriented generalizers. In this paper the author have used a large dataset i.e. KDD which helps in anomaly detection methods. In this paper the author has explained the paramount of Voting method over stacking method in the combination algorithm named as boostSVM for the above mentioned specified dataset. The classifiers used are Support Vector Machine (SVM) and AdaBoost where the AdaBoost algorithm boosts SVM to debase the error rates. SVM is mainly chosen as it provides a global maxima instead of local minima's. ROC curves are also being used to justify the results as it helps in evaluating the performance efficviently.

Keywords: Stacking, Voting, ROC, SVM, AdaBoost, prediction rules, hyperplane, support vectors, boostSVM.

I. INTRODUCTION

In this paper the author has used the classification methodology to find anomalies for the KDD dataset. In the dataset used in this paper two classes are being contemplated i.e. normal and anomaly. Classification is mainly used to determine whether a particular attribute belongs to normal or anomaly class i.e. if S= attribute

N= normal class &

A= anomaly class Then we need to find whether

$S \in N$ or $S \in A$

Here we will be comparing two methodologies for KDD dataset which are stacking and boostSVM method. Stacking is the method which combines various classifiers, step by step to increase its efficiency where the output of the first classifier is taken as an input to second classifier. In the voting method winner survive strategy is used in which the classifier with maximum accuracy or minimum error is selected as the output classifier and the accuracy level of the same is considered to be the final output. For the stacking and boostSVM method two classifiers are used which are SVM and AdaBoost. Boosting technique is used to create a highly accurate prediction rule by combining various weak prediction rules. It boosts other algorithm to provide more accurate results by reducing the error rates. In this paper AdaBoost[1] is used to boost SVM which as a result reduces the error rate shown in the results below. Support Vector Machine is used for classification of object depending on the number of divulge classes. It stratify the object by drawing hyperplane to discriminate the various classes. The data points that lie closest to the decision surface are called the support vectors. Normally we divide the data into two types, linear and nonlinear. Linear data are the data which can be easily separable by drawing a boundary in between whereas non linear are datasets which are in a slapdash format and cannot be separated by a single hyperplane. SVM deals with both the data types i.e. for linear data it uses LibSVM while for nonlinear data it uses kernel functions. Kernel functions help in the transformation that maps the original data to the new space. The main reason behind using SVM as a base classifier is that it gives a global maxima instead of multiple local minima's. The dataset used here is the KDD99 dataset which helps in intrusion detection and it is based on 1998 DARPA initiative. Intrusion detection helps in detecting the security issue signs by monitoring the events that occur in computer systems. It includes the procedure of identifying the set of malicious actions that modus vivendi the information resources. Normally for intrusion detection Systems we have two approaches : misuse detection and anomaly detection. Misuse detection basically works by pattern detection i.e.it matches the pattern between the captured network and attack signatures[2][3]. As soon as it detects any malicious

thing it immediately raise a trepidation. The main advantage of it is that it detects the familiar attacks easily, but faces problems for unknown errors. While anomaly detection works by behavioral identification[3][4]. It searches for the behavior that doesn't come seems normal and it establishes a model for all users and components in a network. If any aberration is being observed immediately a trepidation is being raised. The main advantage of this is that it doesn't require any known attacks to detect the anomaly, but it mainly finds problem in deciding what constitutes the attack and may give high false positive rate.[2][3][4] There are two classes involved in this dataset which are normal class and the anomaly class which includes four types:

a. DoS: In Denial of Service statutory users is being prevented by the assailant from using a service.

b. R2L: In Remote to Local assailant tries to gain access over the victim's machine.

c. U2R: In User to Root assailant have local access to a victims machine but tries to gain super user prerogative.

d. Probe: Here the assailant tries to get information about the target host.

II. STACKING:

Stacking is the method which uses the combination of generalizers rather than choosing any of the results with certain conditions. It is the method that takes the output guesses of generalizers as an input component in new space and then again generalizing it in the new space.

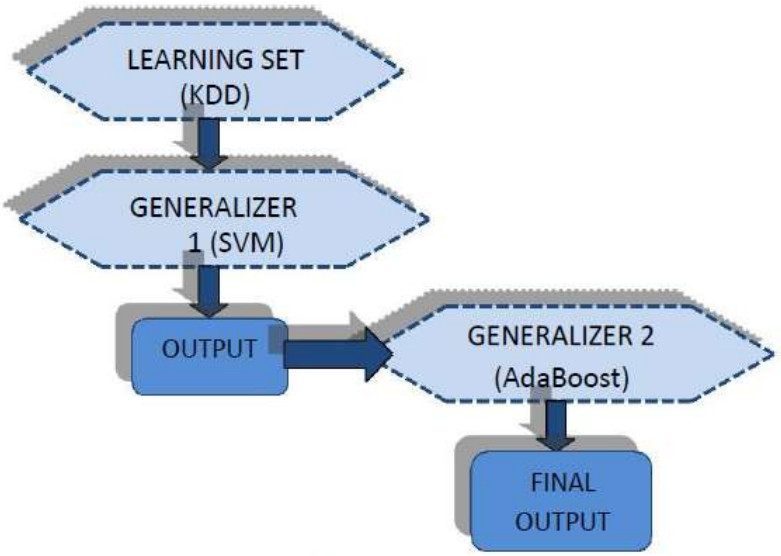

Fig1: Stacking

For Stacking method we first divide the whole dataset into n number of partitions. For all n partitions we again take two sets for each partition. Usually we consider both the sets to be disjoint. Suppose we have Sij where i denotes the number of partitions and j denotes the two disjoint sets. Stacking basically consists of two stages: Stage 1: Base learner, which learns from a dataset by using various models. A new dataset is being created by combining the outputs of the various models and the instance of that dataset is used for the prediction purpose. Stage 2: Stacking Model Learner takes the new dataset created by base learner and use it to obtain the final output.[5] For example, here the author has stacked AdaBoost with Support Vector Machine (SVM) to find the accuracy level of the dataset in the anomaly detection procedure. Here the AdaBoost is used as stage 1 classifier i.e. the base classifier whose output can be used as an input variable to SVM which is our stage 2 classifier i.e. the Stacking Model Learner. The Stacking Model Learner tries to learn from the data obtained from stage 1 and the ways of combining the predictions obtained from various models to achieve best accuracy level.[6] Though we know that AdaBoost method helps in boosting any algorithm to decrease the error rate or to increase the accuracy level, but for stacking method with KDD database , AdaBoost could neither increase the accuracy of SVM nor could decrease the error rate which is being clearly shown in the results whereas when we have used voting method for combining them we could find better results.

III. boostSVM

There are various approaches to combine classifiers at various levels:

a. Combiner approach: In this approach the main focus is on the way of combining the classifier results. The logic of the combiner decides the performance of the system.

b. Base Classifier approach: The base classifier design model for the ensemble is partly specified with bagging and boosting models but for combining the classifiers the logic used is not allied with any base classifiers.

c. Feature Level: At this level different feature subset is used is used for the classifiers. The dataset is divided here so that each classifier can get training over its own dataset. d. Manipulate output labels: The outputs received by the classifier can be manipulated by using error correcting codes (ECOC). There are more approaches of interest which includes miscellany classifier ensembles and also include certain clustering ensembles also. In this paper the author has used the fusion of label outputs. The logic of combining outputs depends on the information of individual classifiers. There are basically three types of classifier outputs:

1. The Abstract level: The classifier produces a class label which belongs to a feature space and each classifier output defines a vector(S) which can be mathematically written as: If Ci is the classifier, Li is the class label and F gives the feature space then Li ∈ F where i=1,2,.....m If S defines the vector, then the m classifier outputs define the vector as S=[S1,S2.....Sm] n ∈ Fm At this level, we didn't find any information about the guessed level nor are any alternative suggested which is the reason for calling this level as the ubiquitous one.

2. The Rank level: It is mainly used for large number of class labels. At this level the output of each classifier belongs to the feature space which is the probable reason of providing the correct labels. [8,9] Ci ∈ F

3. The Measurement level: Here each classifier produces a- dimensional vector [D] which provides output between 0 and 1.

D= [Ci,1, Ci,2,.....Ci,a] ^{m}

In this paper the combine classifier model is known as boostSVM which uses SVM as the base classifier and AdaBoost is used here to boost the classifier SVM which helps in mainly reducing the error rate and if possible increases the accuracy. LEARNING SET (KDD) GENERALIZER 1 GENERALIZER 2 (AdaBoost) (SVM) COMBINER (boostSVM) FINAL OUTPUT

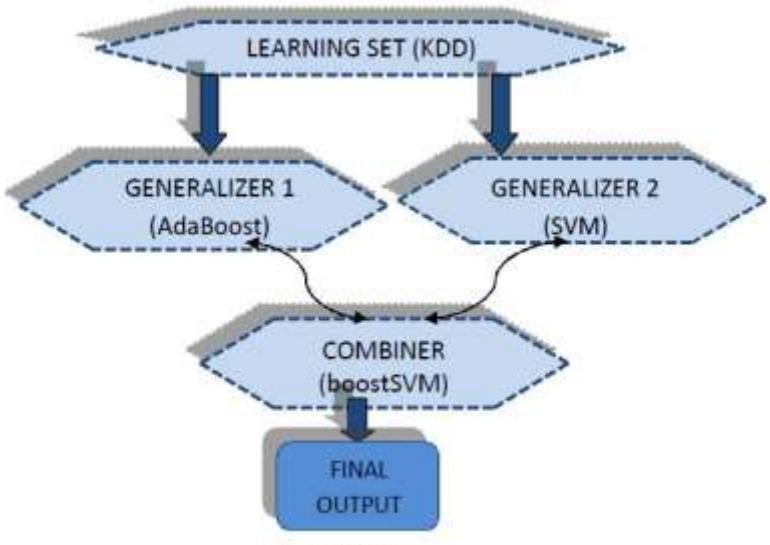

Fig2: boostSVM

Here we have introduced the majority vote technique which basically considers the classifier having more accuracy level but considering the average of probabilities. The Average of probabilities returns the mean of a probability distribution for the base classifiers. It considers the classifier which reduces the error more which is being clearly shown in the results.

IV. RESULTS

There are various ways of testing the dataset but the main aim is to train the classifier as much as possible and provide the maximum amount of data for testing but we also need to be careful about overtraining classifier. We may overtrain the classifier by providing the same dataset for testing and training. Suppose is the dataset of size C x a where

C = number of objects and contain a- dimensional feature vectors. Thus the various ways of making the best use of can be summarised as follows:

1. R-method: This Re-substitution method considers the same dataset for testing and training. Though this gives very good results but this is the condition which overtrain the classifier. This gives biased results so for understanding the classifier more we should ignore this method.

2. S-method: The splitting method or Hold out method splits the dataset into two parts and uses one part for training and one part for testing.

3. Random-method: This method overcomes the disadvantage of R-method by precipitating a random set which will be used for testing the classifier. This method helps in understanding the classifier more than its previous version. Though R-method may give more accuracy level but it gives biased predictions. This biasness is being minimized by the Random-method.

4. CV-method: Lastly we have the Cross Validation which is another technique for estimating performance. Suppose we have a dataset of size "P" which is divided into "K" subsets. Now from this K subset we use one for testing and the remaining "K-1" for training and we repeat the procedure till we reach the last subset.

In this paper cross validation method is being used to find the accuracy level of boostSVM as it helps the classifier to estimate the performance properly. The CV-method used here is 2-folded, i.e. the dataset is divided into two subsets i.e. set1 and set2 . Initially set 1 is used for training and set 2 is used for testing and for next step continues by taking set2 as training set and set1 as testing set. In this paper the error rate and the accuracy level are being checked.[14][15] The error rate is being determined by comparing the error by the classifier and the error by the labelled data

i.e. Error (C) = EC/EL

Where EC = misclassification by classifier (C) and

EL = misclassification of the labelled data

After finding the error the accuracy level is being determined by

Accuracy = 1 - Error (C)

The results for boostSVM compared with AdaBoost, SVM ans Stacking method is shown below: Classifier Correctly Classified ROC Relative absolute Root relative squared [CC] (%) error [RAE] (%) error [RRSE] (%) AdaBoost 94.3355 .9878 15.94 39.0935 SVM 94.0219 .936 12.0113 49.0128 Stacking 53.386 .5 100 100 boostSVM 94.0219 .995 13.9756 35.9053

Classifier	Correctly Classified [CC] (%)	ROC	Relative absolute error [RAE] (%)	Root relative squared error [RRSE] (%)
AdaBoost	94.3355	.9878	15.94	39.0935
SVM	94.0219	.936	12.0113	49.0128
Stacking	53.386	.5	100	100
boostSVM	94.0219	.995	13.9756	35.9053

Table 1: Accuracy and error for boostSVM

The graphical representation of the above result is shown below:

100 50 RRSE CC RAE 0 ROC ROC CC RAE RRSE

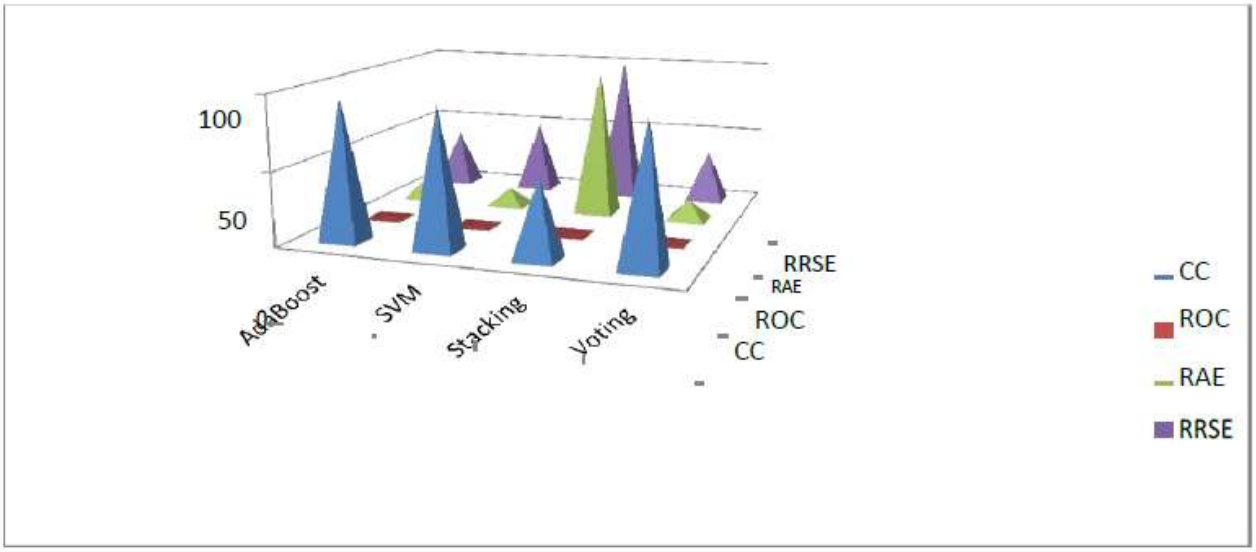

Fig 3: The graphical representation of the results

The ROC curves for verifying the result are shown below:

Fig 8: ROC for anomaly class for Stacking

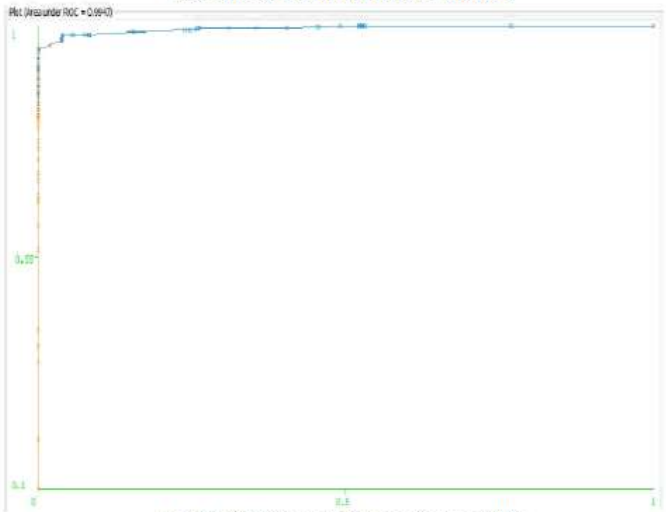

Fig 9: ROC for normal class for Stacking

Fig 10: ROC for anomaly class for boostSVM

Fig 11: ROC for normal class for boostSVM

The ROC (Receiver Operating Characteristics) or AUC (Area under the curve) shown above is the statistical attributes which helps in determining the active compounds in the dataset. The X-axis here helps in plotting the false positive rate whereas Y-axis quadrate to the true positive rate. The color in the ROC curves represents the threshold value and the compounds which exceeds the current threshold value is considered to be active. For a particular attribute if correct prediction is being made, then that attributes is prophesied as active one. The true positive values mentioned above can be identified using confusion matrix which helps in anticipating the performance of the algorithms. The confusion matrix consists of four quadrants which includes true positive, true negative, false positive and false negative. The true positive section is the most secure one as it correctly identified the object. Secondly, true negative is the portion which of negative cases which are classified correctly. Thirdly , false positive are the negative classes which are incorrectly classified. Finally, false negative positive cases which is incorrectly classified and it is the most dangerous one. Thus the accuracy can be increased by correctly classified objects or we can say if more values are there in positive diagonal then it is the more efficient classifier. False Negative(FN) False Positive(FP)

Fig 12: Efficiency of the classifier increases diagonally for the confusion matrix.

The confusion matrices for our results are shown below:

1. ***AdaBoost Confusion Matrix:***

a (normal)	b (anomaly)
12919(TP)	530 (FN)
897 (FP)	10846(TN)

2. ***SVM Confusion Matrix:***

a (normal)	b (anomaly)
13442 (TP)	7 (FN)
1499 (FP)	10244 (TN)

3. ***Stacking method Confusion Matrix:***

a (normal)	b (anomaly)
11743 (TP)	0 (FN)
13449 (FP)	0 (TN)

4. ***boostSVM Confusion Matrix:***

a (normal)	b (anomaly)
13442 (TP)	7 (FN)
1499 (FP)	10244 (TN)

V. CONCLUSION

Thus the result clearly shows that though both Stacking and boostSVM methods are the coherent method for combining classifiers for getting better results but for KDD dataset which can be considered as a big data as it contains a huge amount of information, boostSVM gives a much more better accuracy level than stacking method. The base classifier used in this paper is the SVM which is being boosted by AdaBoost . There are variegated reasons for selecting SVM which includes its way of avoiding overfitting problems. Secondly it doesn't contain local minima which increases its efficiency. Thirdly, it can deal with non-linear data also with the help of kernel functions. Lastly, it provides conjecture for the test error conditions. Though for nonlinear data it is being said that selecting kernel may become a difficult task but if we observe the objective function for logistic

regression it can be seen that in comparison for dealing with non-linear data, this difficult task is worth. The SVM for KDD dataset increases the area under the curve or the ROC to the maximum, i.e. 0.995 which is greater than both individually SVM and Adaboost. ROC is the statistical characteristics which easily help in identifying the instances correctly classified and our result which shows ROC for boostSVM is 0.995 which clearly indicates that it can very advantageously increases the viewing components.

REFERENCES:

[1] Freund, Y., Schapire, R.E.: A Decision-Theoritic Generalization of on-line Learning andan Application to Boosting. (1995)

[2] Panda, M., Patra, M.R.: Ensemble of Classifiers for Detecting Network Intrusion. In: InternationalConference on Advances in Computing, Communication and Control(ICAC3'09), pp. 510-515. (2009)

[3]Garcia-Teodoro,P., Diaz-Verdejo,J., Macia-Fernandez,G., Vazquez,E.: Anomaly-basednetwork intrusion detection: Techniques, systems and challenges.Computer & Security,Volume 28, Issues 1-2, pp. 18-28. (2009)

[4]Davis, J.J., Clark, A.J.: Data preprocessing for anomaly based network intrusion detection:A review. Computer & Security, Volume 30, Issues 6-7, pp 353-375. (2011)

[5] Graczyk, M., Lasota, T., Trawiński, B., Trawiński, K.: Comparison of Bagging, Boostingand Stacking Ensembles Applied to Real Estate Appraisal. In: ACIIDS'10 Proceedings ofthe Second international conference on Intelligent information and database systems: PartII Proceeding. Springer-Verlag Berlin, Heidelberg. (2010)

[6] Zhou, Z.-H.: Ensemble Learning, Encyclopedia of Biometrics, Volume 1, pp. 270-273,Berlin, Springer, ISBN: 978-0-387-73002-8 (2009)

[7] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics, 22:418–435, 1992.

[8] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:66–75, 1994.

[9] J. D. Tubbs and W. O. Alltop. Measures of confidence associated with combining classification rules. IEEE Transactions on Systems, Man, and Cybernetics, 21:690–692, 1991.

[10] W. H. E. Day. Consensus methods as tools for data analysis. In H. H. Bock, editor, Classification and Related Methods for Data Analysis, Elsevier Science Publishers B.V. (North Holland), 1988, pp. 317–324.

[11] R. Battiti and A. M. Colla. Democracy in neural nets: Voting schemes for classification. Neural Networks, 7:691–707, 1994.

[12] L. Lam and A. Krzyzak. A theoretical analysis of the application of majority voting to pattern recognition. In 12th International Conference on Pattern Recognition, Jerusalem, Israel, 1994, pp. 418–420.

[13] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: An analysis of its behaviour and performance. IEEE Transactions on Systems, Man, and Cybernetics, 27(5):553–568, 199

[14] Ludmila I. Kuncheva (2004). Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons, Inc..

[15] J. Kittler, M. Hatef, Robert P.W. Duin, J. Matas (1998). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(3):226-239.

[16] David H. Wolpert (1992). Stacked generalization. Neural Networks. 5:241-259

[17] Yasser EL-Manzalawy (2005). WLSVM. URL http://www.cs.iastate.edu/~yasser/wlsvm/.

[18] Chih-Chung Chang, Chih-Jen Lin (2001). LIBSVM - A Library for Support Vector Machines. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[19] Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996.

[20] Mohammad Khubeb and Shams Naahid, Analysis of KDD CUP '99 Dataset using Clustering based Mining, International Journal of Database Theory and Application, 6(5), 2013, 23-34.

[21] Mahbod Tavallaee,Ebrahim Bagheri,Wei Lu and Ali .A.Ghorbani, A Detailed Analysis of the KDD CUP '99 Dataset, Proc.of 200 9 IEEE Symposium on Computational Intelligence in Security and Defense Applications,978-1-4244-3764-1/09