Network Intrusion Detection System based Experimental Study of Combined Classifiers using Random Forest Classifiers for feature selection

Nilesh B. Nanda
Student - Research Scholar (Computer Science) Gujarat Vidyapith, Ahmadabad-Gujarat (India)
nilideas@yahoo.co.in

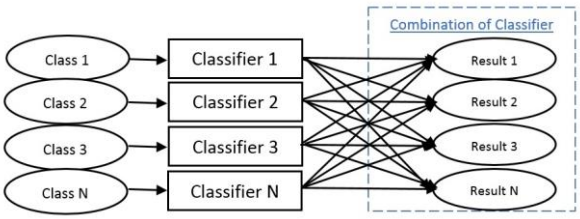
Dr. Ajay Parikh
Head Department of Computer Science Gujarat Vidyapith Ahmedabad-Gujarat (INDIA)
ajayparikh.gvp@gmail.com

***Abstract*—Most classifiers are producing excessive accuracies. In our research experiment, we tested and analyzed the performances of the combination of different classifiers. In this research work used k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree, Naive Baye. These Combined model implemented over Dos, Normal attacks. The detection of fraudulent attacks is considered as a classification problem. Experiments have been performed with different classification methods on KDDCup99 Dataset and compared combined classifiers using models accuracy and confusion matrix. Cross-validation means score used for accuracy. Remove noise data and feature selection we applied Random forest Classifier. We used anaconda navigator with python and R programming for implementation.**

Keywords*— network intrusion, support vector machine, decision tree, Decision Tree, detection.

I. INTRODUCTION

A group of classifiers is a set of classifiers whose personal forecasts are mixed in some way (typically by voting) to classify new examples. A standout amongst the utmost dynamic zones of research in machine learning has been to think about strategies for developing great groups of classifiers (Dietterich, 1997). The fascination that this point applies on machine learning specialists depends on the commence that combination are regularly substantially more exact than the individual classifiers that influence them to up. The majority of the exploration on classifier combination is concerned with creating groups by utilizing a machine learning calculation. Combination of classifiers are produced by controlling the preparation set, highlights the information, controlling the targets in the machine learning techniques for NIDS. The produced classifiers are regularly consolidated by soft voting or hard voting. Classification is one of the most troublesome tasks. In classification, classifiers are learned from a set of training instances with class labels, and instances are often represented by a set of attributes (tuple). Classifier performance and results are usually classification accuracy or confusion matrix, - score. Mostly network intrusions are the disturb of information security rules. At first, NIDS was implemented for computer-based that located in the datacenter to examine the internal interfaces [1]-[3], but with the evolution of computer networks, the focus gradually shifted toward network-based. Network intrusion detection system (NIDS) performs packet logging, real-time traffic analysis of IP network, and tries to discover if an intruder is attempting to break into the system [4]-[6]. Different Attacks on the network can be referred to as Intrusion. Intrusion means any set of fake activities that attempt to leak the security standards of the information. Network Intrusion detection is one of the enormous information security problems. NIDS (Network Intrusion Detection System) assist the host in resisting internal and external network attacks[1]. In this work, based on the current research topics in network intrusion detection, a new method for adaptive network intrusion detection using a combination of naive Bayes classifier, support vector machine, decision tree, random forest, and K-Nearest Neighbor Learning Algorithm Logistic regression is presented and can handle the above problem. It also explains the difficulty of data mining, such as processing of continuous attributes, coping with lack of attribute values, and noise reduction of training data, using random forest classifiers. This classifier will be evaluated on the NSL KDD dataset to identify attacks on the various attacks categories: Probe (information gathering), DoS (denial of service), U2R (user to root) and R2L (remote to local). The classifier's results are computed for comparison of feature reduction methods to show that the hybrid model is more efficient for network intrusion detection.



1.1 Combination of Classifier

This research work is organized as follows. Section I gives Introduction. Section II discusses the literature survey. Section III overviews the intrusion detection system and its classification. Section IV gives various data mining techniques for NIDS. Section V discusses the various datasets that are used to build a NIDS and the next section is in conclusion.

II. BACKGROUND

Various combined model algorithms have been used in the security area and machine-based learning methods. In this paper, compare the very famous mining algorithm with KDDCup train dataset.

The SVM is the best learning type of pattern algorithm for binary classification. It has been applied to information security for network intrusion detection. For anomaly intrusion detection, SVM has become one of the essential techniques and due to its good generalization of strength and the capacity to overcome the condition of dimensionality[13]. One of the main pleasures of using SVM for NIDS is its accuracy, speed, as the capability of detecting intrusions in real-time is essential [14][15].

Decision tree techniques are used to automatically learn intrusion signatures, pattern and perform the classification activities in computer network systems as usual or intrusive.

K-mean clustering was used to perform importance features extraction through clustering over data and in unsupervised manner cluster the whole KDD cup'90 dataset into parts.

the naïve Bayes model is a reduced Bayesian probability model[12]. The naïve Bayes classifier performs on a strong independence assumption [2,12]. Bias is the error due to groupings in the KDD'90 training data being very large. Variance is the error due to those groupings being too small.

The Voting Classifier is a meta-classifier for combining similar or conceptually different machine learning classifiers for classification via majority or plurality voting and implements "hard" and "soft" voting. In hard voting, predict the final class label as the class label that has been predicted most frequently by the classification models. In soft voting, predict the class labels by averaging the class-probabilities. The main advantage is to provide excellent accuracy, speed and real-time sensing of intrusions. It also has the ability to update training and signature pattern dynamically.

III. THE DIFFERENT TYPE OF NIDS ATTACKS

The KDD Cup '90 network intrusion detection dataset [7] based on the DARPA' 98 datasets is the only revised data set that is open to the public and provides the main data for researchers working in the field of intrusion detection. The details of the KDD dataset are described in the next section. KDD Dataset is generated using the simulation of a military network, which consists of three operating machines running different operating systems and traffic. The simulated period is several weeks. A regular TCP connection is created to profile what is expected of the military network and attacks fall into one of the following four categories:

- Denial of Service (Dos): Dos Attacker tries to slow the service server and send continuous garbage packets.
- Remote to Local (r2l): Attacker try to gain access to remote machine because they do not have rights to access or does not have control of same [13]-[14][15].
- User to Root (u2r): Attacker does not have super or root privilege on the machine, it has a local machine but does not has full rights.
- Probe: Attacker tries to get information from the remote host without knowing actual users.

Rate/Accuracy of Classification

The classification rate or accuracy is provided by the following relational character.

$$\text{ACCURACY} = \frac{TP + TN}{TP + TN + FN + FP}$$

But there is a problem with accuracy. This assumes the same cost for both kinds of errors. 99% accuracy can be excellent, good, ordinary, poor or frightening, depending on the problem.

A recall can be determined by isolating the percentage of the total number of perfectly classified positive samples by the total number of positive samples. The recall is given by the following relation.

$$\text{RECALL} = \frac{TP}{TP + FN}$$

To obtain the exactness worth, divide the entire range of adequately categorized positive examples by the entire range of foretold positive examples. High accuracy indicates that the instance labeled as positive is really positive (few FPs). Accuracy is given by the subsequent relation.

$$\text{ACCURACY} = \frac{TP}{TP + FP}$$

F-measure: In our work, we figure an F-measure which utilizes Harmonic Mean instead of Arithmetic Mean as it rebuffs the outrageous qualities more. The F-Measure will dependably be closer to the little estimation of Precision or Recall.

$$\text{F measure} = \frac{2 * Recall * Precision}{Recall + Precision}$$

V EXPERIMENTAL RESULTS

In the experiments, used standard NSL-KDD dataset. This dataset has several benefits in comparison with KDD'99 [10]:

1) redundant record is removed from the train set to eliminate the bias to the most common records using R Programming. The kddcup99 dataset has been used in this research of which 80% is treated as training data and 20% is considered as testing data.

2) In this dataset, we have used 42 attributes for each connection record including class label containing attack types. Train set dimension: 125973 rows, 42 columns and Test set dimension: 22544 rows.

3) Duplicate records in test sets are removed using R programming.

4) The number of records in the test and train datasets is reasonable.

5) For feature selection, we used the random forest classifier and selected 10 attributes.

In the experiment, subsets of training and test datasets are utilized. In [21] the NSL-KDD'99 dataset is analyzed using all experimental algorithm. The dataset is clustered into normal, DoS, Probe, R2L, and U2R attacks. It is shown that NSL-KDD dataset has reasonable accuracy in comparing with KDD99. The proposed method is implemented by the R Programming, Python, Jupyter notebook with Anaconda Navigator software and tested on NSL-KDD dataset. The number of training and testing datasets which are used for the experiments are shown in Tables and Graph.[13][14][15].

4.1 Training data describe in table format.

The number of training and test datasets used for the experiments.

Attack	Attack Class	Frequency Percent Train	Attack Class	Frequency Percent Test
DoS	45927	36.46	7458	33.08
Probe	11656	9.25	2421	10.74
R2L	995	0.79	2754	12.22
U2R	52	0.04	200	0.89
Normal	67343	53.46	9711	43.08

4.2 Attack Class Distribution.

In KDDcup dataset has total 42 attributes. Using random forest classifier we selected 10 attributes for experiment of combined classifiers. Selected attributes are src_bytes , Count , dst_host_diff_srv_rate, logged_in , service, dst_bytes,srv_count , dst_host_same_src_port_rate , dst_host_srv_count , dst_host_serror_rate and.

Two scenarios are used to investigate the performance of the proposed method is compared with the all combined models method: Scenario 1: in this experiment, just training datasets are used for the algorithm. Thus the training and test datasets are entirely separated from each other. Scenario 2: in this experiment, in training not only train dataset is used, but also a subset of the test dataset is used. Thus the training and test datasets are not entirely separated from each other.

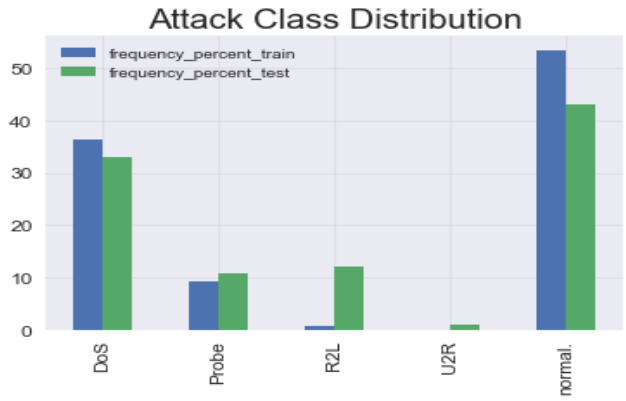


Fig: 1 Attack class bar plot

The experimental analysis combined classifier using the voting classifier.

Combined Classifier	Cross Validation Mean Score	Model Accuracy
Naive Baye & Decision Tree	0.99994803046	1.0
Naive Baye & KNeighbors	0.999339216232	0.999628766167
Naive Baye& LogisticRegression,	0.985796595808	0.986063881918
Decision Tree & KNeighbors	0.999962879373	1.0
Decision Tree & LogisticRegression,	0.99994803046	1.0
KNeighbors & Logistic Regression,	0.999428309707	0.999665889551

4.3 Evolution models Result.

For analysis of models accuracy, we created a two-target classes normal class and an attack class. In attack class list we consider DoS (0.0), Probe(2.0), R2L (3.0), U2R (4.0) and For normal class consider Normal (1.0).

Combined Classifier	Model Accuracy	Precision		Recall		F1-Score	
		0.0	1.0	0.0	1.0	0.0	1.0
Naive Baye & Decision Tree	0.835	0.85	0.74	0.94	0.50	0.90	0.60
Naive Baye & K-Neighbors	0.797	0.91	0.56	0.81	0.76	0.86	0.65
Naive Baye & Logistic Regression	0.773	0.84	0.54	0.87	0.47	0.85	0.51
Decision Tree & K-Neighbors	0.842	0.85	0.80	0.96	0.48	0.90	0.60
Decision Tree & Logistic Regression	0.835	0.85	0.74	0.94	0.50	0.90	0.60

K-Neighbors & Logistic Regression	0.797	0.91	0.56	0.81	0.76	0.86	0.65

4.4 Test model accuracy.

4.3 The simulated analysis of the NIDS methods of all the combined classifier of Naive Baye, Decision tree, and Support Vector Machine (SVM) is done using well define performance measuring parameters which are accuracy and Cross-Validation Mean Score. Here, table 4.3 shows accuracy result of the Evaluates models and Test models using a combination of Naïve Baye, SVM, Decision Tree, Logistic Regression and KNN algorithms. After analysis, a combination of it is found that the overall accuracy rate for Evaluates method is about 99.82% whereas the Test models are 99.94%. Decision tree accuracy is 100% during evaluates models and 99.83% during Test models. KNn algorithm accuracy was 99.99% whereas in the test was models 99.98%. So it is concluded that Evaluates models generate a more accurate result for network intrusion detection as compared to the Test method.

After analysis, a combination of Naive Baye & Decision Tree is found that the overall accuracy rate for evaluation method is about 99.994% whereas the test models accuracy is 100%. Naive Baye & KNneighbours overall accuracy rate for evaluation method is about 99.933% and test model accuracy is 99.962%. Naive Baye& Logistic Regression accuracy rate for evaluation method is about 98.579% and model accuracy is 98.606%. Decision Tree & KNneighbours accuracy rate for evaluation method is about 99.996% and model accuracy is 100%. Decision Tree & LogisticRegression accuracy rate for evaluation method is about 99.994% and model accuracy is 100%. KNneighbours & Logistic Regression accuracy rate for evaluation method is about 99.942% and model accuracy is 99.966%Table 4.4 Comparison of an accuracy rate of Evaluates models and test models with DOS attacks of classifiers model.

Table 4.3 Comparison of Cross-Validation Mean Score of Evaluates models and test models with DOS attacks of a combination of the classifier of SVM, Decision Tree, Naive Baye, KNN model, Logistic regression with the voting classifier.

Here, table 4.3 also shows the cross-validation mean score of evaluates the model for Naïve Baye with a decision tree and decision tree with a logistic regression algorithm which gives good results compared to other classifiers.

VI Conclusion

Network Intrusion Detection is becoming very challenging day by day. R Programming and Python can detect attacks in the network. In this paper compare and analysis of a various combination model like SVM, Decision tree, Naïve Baye, KNN, Logistic Regression models to improve the network intrusion detection system (NIDS) and after staring at ending that the execution of the hybrid version has considerably progressed the algorithm accuracy and as a conclusion it exhibits the significance of preprocessing in NIDS. Compared to the current methods, Evaluates model fairly improves the accuracy of Dos attacks. Hence conclude that the combined model of classifier proves to be an efficient classifier for DoS attacks. Using combined models like KNN and support vector machine or decision Tree and Naive Baye and other computational intelligence with other dataset technique which is a future work to be proposed to improve detection efficiency.

REFERENCES

[1] D.Dennin,(2007) "An intrusion-detection model", IEEE Transactions on Software Engineering.

[2] J. Frank, (2014) "Machine learning and intrusion detection: Current and future directions," in Proceedings of the National 17th Computer Security Conference, Washington, D.C.

[3] Lee, W., Stolfo, S., &Mok, K. (1999). A Data Mining Framework for Building Intrusion Detection Model.Proc. IEEE Symp. Security and Privacy, 120-132.

[4] Amor, N. B., Benferhat, S., &Elouedi, Z. (2014). Naive Bayes vs. Decision Trees in Intrusion Detection Systems.Proc. ACM Symp.Applied Computing, 420- 424.

[5] Mukkamala, S., Janoski, G., &Sung, A. (2012). Intrusion detection using neural networks and support vector machines. Paper presented at the International Joint Conference. on Neural Networks (IJCNN).

[6] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham,(2017) Principle Components Analysis and Support Vector Machine based Intrusion Detection System, IEEE.

[7] T.Shon, Y. Kim, C.Lee and J.Moon,(2015), A Machine Learning Framework for Network Anomaly Detection using SVM and GA, Proceedings of the 2015 IEEE.

[8] SandyaPeddabachigari, Ajith Abraham, CrinaGrosan, Johanson Thomas (2015). Modelling Intrusion Detection Systems using Hybrid Intelligent Systems. Journal of Network and Computer Applications.

[9] KyawThetKhaing (2010), Recursive Feature Elimination (RFE) and k-Nearest Neighbor (KNN) in SVM.

[10] NSL-KDD Dataset for Network-based Intrusion Detection Systems. Available at: http://nsl.cs.unb.ca/NSL-KDD.

[11] H. Liu and H. Motoda(1998), Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic.

[12] J.R. Quinlan,(2016) "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106.variant Firefly and Bk-NN Techniques", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 6 (2016) pp 4161- 4166.

[13] N. Nanda,A. Parikh (2017) "Classification and Technical Analysis of Network Intrusion Detection Systems," International Journal of Advanced Research in Computer Science,Volume 8, No. 4, May-June 2017.

[14] N. Nanda, A. Parikh (2018) "Network Intrusion Detection System: Classification, Techniques and Datasets to

Implement," International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248 Volume: 4 Issue: 3 106 – 109.

[15] Nilesh B. Nanda , Ajay Parikh (2018) "Experimental Analysis of k-Nearest Neighbor, Decision Tree, Naive Baye, Support Vector Machine, Logistic Regression and Random Forest Classifiers with Combined Classifier Approach for NIDS", International Journal of Computer Sciences and Engineering, E-ISSN: 2347-2693, Vol.-6, Issue-9, Sept. 2018, 940-943.

Author Profile

Nilesh Nanda pursed M.Phil of Computer Science from Gujarat Vidyapith, Ahmedabad, India in 2013 and currently pursuing Ph. D of Computer Science in the field of Network intrusion detection system from Gujarat Vidyapith. He is currently working as Computer Programmer in VVP Engineering College, Rajkot Gujarat (INDIA) Since 2000.

Dr. Ajay Parikh, Professor & Head, Department of Computer Science, Gujarat Vidyapith, Ahmedabad (Gujarat) INDIA. He pursued Master of Science from Gujarat University. He pursued M. Phil and Ph.D. of Computer from Gujarat Vidyapith. He has published 11 research papers in various conferences (National & international) has an excellent academic line of experience and published. His area of interest Machine Learning, Data Science, SOA, ICT Application in animal health care and Rural Development. He guided many Ph. D. and M.Phil Scholar students. He organized and participated in various seminars, Workshop, and training camps. He delivered lectures in refresher courses. He delivered Radio and TV talk. He delivered international and national lectures. He is a member of organizing a committee, program committee, international conference, workshop and reviewer in journals. He reviewed the research project. He has membership in professional and other bodies.