International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.13
June-2014,
Pages:2890-2894

www.semargroup.org,
www.ijsetr.com

Performance Evaluation of Different Classifiers for Detection of Attacks in Unauthorized Accesses

MYA THIDAR MYO WIN[1], KYAW THET KHAING[2]

[1]Faculty of Information and Communication Technology, University of Technology, Yatanarpon Cyber City, Pyin Oo Lwin, Myanmar, E-mail: myathidarmyowin@gmail.com.

[2]Dept of Hardware, University of Computer Studies, Yangon, Myanmar, E-mail: kyawthetkhaing.ucsy@gmail.com.

Abstract: Intrusion Detection (ID) is the most significant component in Network Security System as it is responsible to detect several types of attacks. Classification of Intrusion detection, according to their features into either intrusive or non intrusive class is a widely studied problem. The aim of this paper is to investigate the performance of various classifiers for intrusion detection data. The features of KDD Cup '99 attack dataset are reduced for each class of attacks performed manual feature selection; using our domain knowledge with analyzing the nature of the attack. In this paper, we study and analysis of four machine learning algorithms Random Forest, J48, k nearest neighbor and Naïve Bayes of data mining for the task of detecting intrusions and compare their relative performances. Based on this study, it can be concluded that Random Forest is the most suitable associated algorithm than the other three algorithms with high true positive rate (TPR) and low false positive rate (FTR) and high accuracy.

Keywords: Intrusion Detection, Classifiers, KDD'99 Dataset, U2R, R2L.

I. INTRODUCTION

The field of information security has evolved rapidly in recent years because of the swift growth and widespread use of electronic data processing, and also of business conducted through the Internet and other computer networks (LAN, WAN, etc.). These application areas make networks an attractive target for abuse and thus an area of vulnerability. At the same time, the tools of the intruder and the hacker have improved substantially. In order to both combat the growing number of attacks and to maintain critical information services, both academic and industry groups have been developing systems to monitor networks and to raise alarms over suspicious activities. These systems are called Intrusion Detection Systems (IDS) [1]. Intrusion Detection is defined as the problem of identifying individuals who are using a computer system without authorization and those who have legitimate access to the system but are abusing their privileges. An Intrusion Detection System (IDS) gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).

An IDS is designed to detect unscrupulous activities that compromise the confidentiality, integrity, or availability of network or computer systems and to analyze what happens or what has happened to indicate that the computer has been misused. Machine learning is a valuable tool for intrusion detection that offers a major opportunity to improve quality of IDs. Generally, there are two types of detecting an intrusion; misuse detection and anomaly detection. In misuse detection, an intrusion is detected when the behavior of a system matches with any of the intrusion signatures. In the anomaly based IDs, an intrusion is detected when the behavior of the system deviates from the normal behavior. Intrusion Detection Systems (IDS) can also be categorized as host-based IDSs and network-based IDSs according to the target environment for detection. Host-based IDSs usually monitor the host system behavior by examining the information of the system, such as CPU time, system calls and command sequences. Network-based IDSs, on the other hand, monitor network behavior usually by examining the content (e.g., payload) as well as some statistical attributes of network traffic [2].

First we performed feature selection with domain knowledge of the nature of attacks to effectively detect different classes of attacks. Our proposed feature selection is based on the nature of each attack that is to understand which feature is most important for each attack. Second, we present the application of machine learning to intrusion detection. We analyse four learning algorithms Random Forest, J48, K Nearest Neighbor and Naïve Bayes for the task of detecting intrusions and compare their relative performances. These algorithms provide the efficient results for intrusion detection data .There is only available data set is

KDD data set for the purpose of experiment for intrusion detection. The rest of the paper is organized as follows: Section II presents an overview of related works. Section III gives the features within the KDD data set and Section IV gives overview classifiers in intrusion detection field. Section V discusses the performance evaluation of our system when applied to the KDD 99 data.

II. RELATED WORKS

Jayshri R.Patel [3] presented performance of four selected decision tree classification algorithms for ranked intrusion detection data is evaluated and investigated. From the experiment & result analysis it is very clear that the performance of Random Forest is better as it correctly identifies more number of instances than other. In [4], Classification of intrusion detection is done based on various machine learning algorithms like J48, Naïve bayes, OneR and BayesNet. They find the Decision tree algorithm J48 most suitable with high positive rate and low false positive rate. Dr. Pfahringer [5] uses different standard learning algorithms such as C5 (trees, rules, and boosted trees), Ripper, naive bayes, nearest neighbor, a back-propagation neural network and a radial-basis function neural network. Then the experiments involving Ripper, nearest neighbor, and neural networks were cancelled because of extreme runtime requirements. Of the four remaining algorithms, all variants of C5 performed much better than naive bayes.

The authors have used four different learning algorithms to produce a set of classifiers for this evaluation experiment [6]. The J48 algorithm, which we already have introduced in the example, has produced one pruned and one unpruned decision tree classifier. Both sub tree rising and reduced error pruning was applied on the first classifier. One of WEKA's nearest neighbor implementations, called IBk, has been used to produce one classifier based on one neighbor (IB1) and another classifier bas ed on ten neighbors (IB10). Mitchell [7] argues that this algorithm is known to perform comparably with decision tree and neural network learning in some domains. This makes it an interesting algorithm to use in the experiments concerning the evaluation of classifiers using a measure function. Panda and Patra [8] have compared the performance of Naïve Bayes with the Neural Network approach and found its suitability in building an intrusion detection model.

A. Kdd'99 Dataset and Properties

KDD Cup '99 intrusion detection datasets [9] which are based on DARPA '98 dataset provides labelled data for researcher working in the field of intrusion detection and is the only labelled dataset publicly available. The details of KDD dataset are given in the subsequent section. The KDD dataset is generated using a simulation of a military network consisting of three target machines running various operating systems and traffic. Finally, there is a sniffer that records all network traffic using the Tcpdump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of the four categories:

- Denial of Service (Dos): Attacker tries to prevent legitimate users from using a service.
- Remote to Local (R2L): Attacker does not have an account on the victim machine, hence tries to gain access.
- User to Root (U2R): Attacker has local access to the victim machine and tries to gain super user privileges.
- Probe: Attacker tries to gain information about the target host.

There are 41 features for each connection, which are detailed in Table I. Specifically, "a connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source IP address to atarget IP address under some well-defined protocol". Features are grouped into four categories:

- Basic Features: Basic features can be derived from packet headers without inspecting the payload.
- Content Features: Domain knowledge is used to access the payload of the original TCP packets. This includes features such as number of failed login attempts.
- Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval.
- Host-based Traffic Features: Utilize a historical window estimated over the number of connections instead of time. Host-based features are designed to access attacks, which span intervals longer than 2 seconds. KDD DataSet Feature (Summarized From[9])

III. CLASSIFIERS FOR INTRUSION DETECTION SYSTEM

Intrusion detection can be considered as classification problem where each connection record is identified as normal or intrusive based on some existing data. Classification for intrusion detection is an important challenge because it is very difficult to detect several attacks, as the attackers are continuously changing their attack patterns. Various classification algorithms can be used for the classification of intrusion data such as Random Forest, J48, K Nearest Neighbor and Naïve Bayes.

Random Forests: Random Forest(s) are one of the most successful tree based classifier. It has proven to be fast, robust to noise and offers possibilities for explanation and visualization of its output. In the random forest, a large number of classification trees are grown and combined. Statistically speaking two elements serve to obtain a random forest re-sampling and random split selection. Re-sampling is done here by sampling multiple times with replacement from the original training data set. Thus in the resulting samples, a certain event may appear several times, and other events not at all. About $2/3^{rd}$ of the data in the training sample are taken for each bootstrap sample and the remaining one-third of the cases are left out of the sample.

TABLE I: KDD Data Set Feature (Summarized From [9])

No	Features Name	No.	Features Name
1	duration	22	is guest login
2	protocol	23	count
3	service	24	srv_count
4	flag	25	serror rate
5	source bytes	26	srv serror rate
6	Destination bytes	27	rerror rate
7	land	28	srv rerror rate
8	wrong	29	same srv rate
9	urgent	30	diff srv rate
10	hot	31	srv diff host rate
11	failed logins	32	dst_host _count
12	logged in	33	dst_host _srv_count
13	# compromised	34	dst_host_same_srv_rate
14	root shell	35	dst_host_diff_srv_rate
15	su attempted	36	dst_host_same src_port rate
16	# root	37	dst host srv diff host rate
17	file creations	38	dst host serror rate
18	# shells	39	dst host srv serror rate
19	# access files	40	dst host rerror rate
20	# outbound cmds	41	dst host srv rerror rate
21	is hot login		

The main features of random forests algorithm [10] are listed as follows:

- It is unsurpassable in accuracy among the current data mining algorithms.
- It shows efficient performance on large data sets with many features.
- It can give the estimate of what features are important.
- It has no nominal data problem and does not over fit.
- It can handle unbalanced data sets.

J.48: J48 is an open source Java implementation of the C4.5 algorithm of the WEKA data mining tool. C4.5 is based on the ID3 algorithm developed by Ross Quinlan [11], with additional features to address problems that ID3 was unable to deal. In practice, the J48 is a Decision tree classifier algorithm. In this algorithm for classification of new item, it first needs to create a decision tree based on the attribute values of the available training data. It discriminates the various instances and identifies the attribute for the same. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign it to the target value that we have obtained.

K-nearest neighbor (KNN): K-nearest neighbor (KNN) is one of the most common methods among memory based induction. Given an input vector, KNN extracts k closest vectors in the reference set based on similarity measures, and makes decision for the label of input vector using the labels of the k nearest neighbors. Pearsons coefficient correlation and Euclidean distance have been used as the similarity measure. When we have an input X and a reference set D = d1, d2, …, dN, the probability that X may belong to class cj , P(X, cj) is defined as follows:

$$P(X, c_j) = \sum_{di \in kNN} Sim(X, di) P(di, c_j) - b_j \quad \textbf{(1)}$$

Where Sim(X, di) is the similarity between X and di and bj is a bias term [12].

Naïve Bayes: Naïve Bayesian classification is called naïve because it assumes class conditional independence. That is, the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is made to reduce computational costs, and hence is considered naïve. The major idea behind naïve Bayesian classification is to try and classify data by maximizing P(Xj|Ci)|P(Ci) (where i is an index of the class) using the Bayes theorem of posterior probability[13].

IV. EXPERIMENTS AND RESULTS

We have used an open source machine learning framework WEKA [Waikato Environment for Knowledge Analysis] written at University of Waikato, New Zealand [14].The input data for weka classifiers is represented in .ARFF [Attribute Relation Function Format], consisting of the list of all instances with the values for each instance separated by commas. We perform our experiments with the benchmark KDD 1999 intrusion data set [9]. The raw data from the KDD 99 is first partitioned into four groups (input data set), DoS attack set, Probe attack set, R2L attack set and U2R attack set. The features of KDD Cup '99 attack dataset are reduced for each class of attacks performed manual feature selection; using our domain knowledge with analyzing the nature of the attack. The reduced feature set is classified with Random Forest classifier, k-nearest neighbor and Naive Bayes. And then compare with the detection of all 41 features with these classifiers.

In the experiments, we randomly selected 9711 normal connections and each attack connections form the training set. For test set, we randomly selected 9711different normal connections and different attacks connections. In the experiments, we use True Positive Rate, calculated as the correctly classify percentage of intrusions detected, and False Positive Rates (FPR), calculated as the percentage of normal connections falsely classified as intrusions, as criteria for evaluation. In the experiments, we randomly selected 9711 normal connections and each U2R and R2L attack connections from NSL-KDD training and testing dataset. We analyse four learning algorithms Random Forest, J48, K Nearest Neighbor and Naïve Bayes for the test of detecting intrusion and compare their relative performances. The experimental results shows that Performance Evaluation of four classification models, Random Forest and K Nearest Neighbor have much better performance than other three

methods and it is also observed that the overall performance of Random Forest and K Nearest Neighbor classification has increased their performance using feature reduction a notable improvement in their classification, means the classification accuracy increases better after feature selection.

TABLE II: Selected Features Based On Attack Nature for Proposed Attack

No.	Attack	Selected Feature
	U2R	
1	Buffer_Overflow	1,2,3,4,5,6,13,14,16,17,18,19,21
2	Loadmodule	1,2,3,4,5,6,10,17
3	Perl	1,2,3,4,5,6,14 ,17,18
4	Rootkit	1,2,3,4,5,6,10,12,13,14,17
	R2L	
1	ftp_write	1,2,3,4,5,6,10,12, 17,18,19
2	guess-password	1,2,3,11
3	warezmaster	1,2,3,4,5,6,10,12, ,22
4	warezclient	1,2,3,4,5,6,10,12,22
5	snmpget	1,2,3,4,5,6,11,12,23,24,29,36
6	snmpguess	1,2,3,4,5,6,11,12,29,36
7	Imap	1,2,3,4,5,6,10,13,14,16,17,18,19,21
8	Phf	1,2,3,4,5,6,10,12,14,18

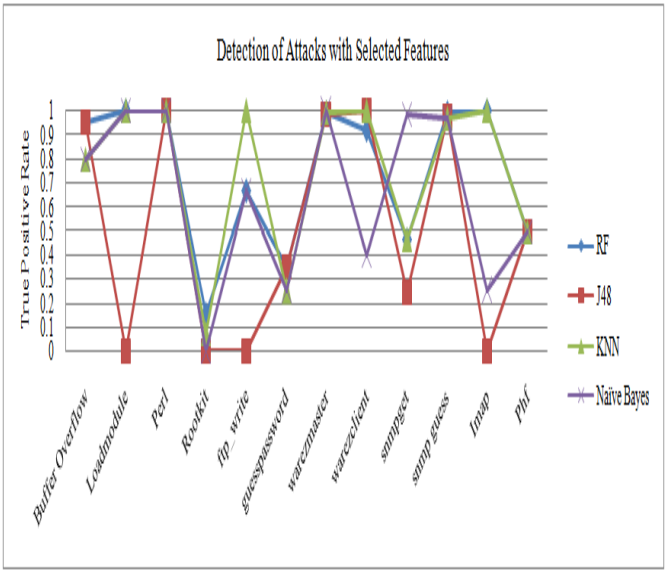

Fig.1. Detection of Attacks with Selected Features.

Fig 1 and 2 show that detection of the true positive rate of attacks in unauthorized accesses with selected features and all features using different classifiers, In Table III and IV show that the percentage of performance comparison of correctly classified in U2R and R2L attack with selected features and all 41 features. It show that the detection of attack in unauthorized accesses using random forest, J48, k-nearest neighbor and Naïve Bayes. The result compare the percentage of correctly classify instances with selected features and all features of each attack. These detection results only using the selected attributes almost remain the same or even become better than those using all the 41 features. This shows that many of the 41 attributes are irrelevant and only a smaller set of attributes is required to extract from raw network traffic for detection of individual attacks. In Table III and IV, results are generated using dataset in weka environment by selecting different classification algorithms. Based on this, random Forest better than other classification algorithms.

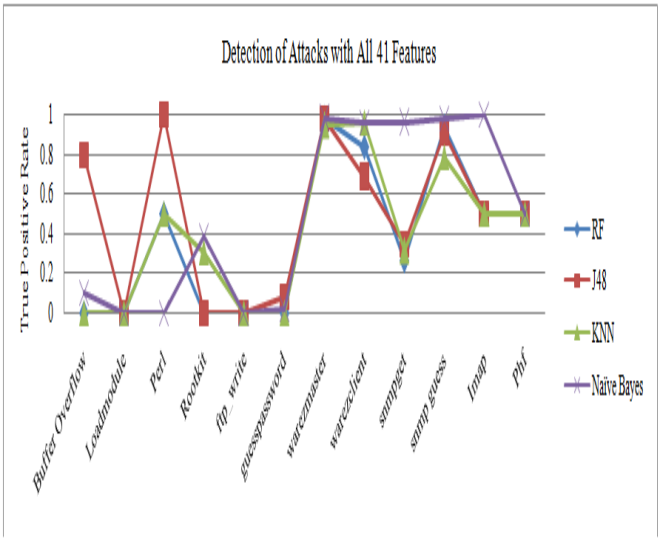

Fig.2. Detection of Attacks with All 41 Features.

TABLE III: Performance Evualation of correctly classified in each U2R attacks with selected features

Categories of Attack	Classifiers			
	Random forest	J48	KNN	Naïve Bayes
Buffer_Overflow	99.9692	99.9383	99.9692	94.6974
Loadmodule	100	99.9794	100	96.8393
Perl	100	100	100	98.1674
Rootkit	99.8869	99.8663	99.8972	97.0794
ftp_write	99.9897	99.9691	100	97.5088
guess-password	99.3375	99.3375	99.2254	98.9094
warezmaster	99.9214	99.892	99.9214	94.4319
warezclient	99.9795	100	100	97.2576
snmpget	99.194	99.2144	99.2654	90.8989
snmpguess	99.9799	99.9398	99.9297	83.2045
Imap	100	99.9588	100	98.4148
Phf	99.9897	99.9794	99.9897	97.9203

TABLE IV: Performance Evaluation of correctly classified in each R2L attacks with all 41 features

Categories of Attack	Classifiers			
	Random forest	J48	KNN	Naïve Bayes
Buffer_Overflow	99.7945	99.9383	99.7945	98.8799
Loadmodule	99.9794	99.9794	99.9794	99.1043
Perl	99.9897	100	100	99.8867
Rootkit	99.8663	99.8663	99.9074	95.7528
ftp_write	99.9691	99.9691	99.9691	97.797
guess-password	98.9807	99.0623	98.9807	98.3896
warezmaster	99.8822	99.892	99.7349	92.566
warezclient	99.9589	99.9178	99.9897	90.7251
snmpget	99.2246	99.3266	99.3674	92.3375
snmpguess	99.8595	99.7791	99.4579	95.8538
Imap	99.9794	99.9794	99.9794	99.3309
Phf	99.9897	99.9897	99.9897	99.9279

V. CONCLUSION

This paper draws the conclusions on the basis of implementations performed using various data mining algorithms. Different classifiers have different knowledge regarding the problem. Feature relevance is performed by analyzing the nature of selected attack. It analyses the involvement of each feature to classification and a subset of features are selected as relevant features.

VI. ACKNOWLEDGMENT

I would like to thank my supervisor and all of my teachers for their helpful comments in improving our manuscript. We would like to thank the anonymous reviewers for their thorough reviews, and constructive suggestions which significantly enhance the presentation of the paper.

VII. REFERENCES

[1] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance", Technical Report, James P. Anderson Co., Fort Washington, PA, April 1980.

[2] K.Nageswara rao, D.RajyaLakshmi, T.Venkateswara Rao. "Robust Statistical Outlier based Feature Selection Technique for Network Intrusion Detection", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[3] Jayshri R.Patel, "Performance Evaluation of Decision Tree Classifiers for Ranked Features of Intrusion Detection", Journal of Information, Knowledge and Research in Information Technology, ISSN: 0975 – 6698| NOV 12 TO OCT 13 | VOLUME – 02, ISSUE – 02.

[4] Yogendra Kumar Jain and Upendra, "An efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction" International Journal of Scientific and Research Publications, Vol. 2, Issue 1, Jan. 2012, ISSN 2250-3153 [2012].

[5] Pfahringer, B. (2005, 1 1), "Winning the KDD99 Classification Cup: Bagged Boosting". Vienna, Austria.

[6] Lavesson, Niklas and Davidson, Paul, "A multidimensional measure function for classifier performance", in Proc. Of 2nd IEEE international conference on intelligent systems, June2004 , pp.508-513.

[7] Mitchell, T. M., "Machine Learning", International Edition, McGraw-Hill Book Co., Singapore, 1997, ISBN: 0-07-042807-7.

[8] Panda, Mrutyunjaya and Patra, Manas Ranjan , " Network Intrusion Detection using Naïve Bayes", International journal of computer science and network security, Dec'30-2007, pp.258-263.

[9] KDD-CUP 1999 Data, http://kdd.ics.uci.edu/ databases /kddcup99 / kddcup99.html.

[10] L. Breiman, "Random Forests", Machine Learning, 45(1):5-32, 2001.

[10] J. R. Quinlan, "Induction of Decision Trees," Machine Learning, Volume 1, pp. 81- 106, 1986.

[11] Xinguo Lu, Xianghua Peng, Ping Liu, Yong Deng, Bingtao Feng, Bo Liao, "A Novel Feature Selection Method Based on CFS in Cancer Recognition", 2012 IEEE 6th International Conference on Systems Biology (ISB), August 18–20, 2012.

[12] Naïve Bayes Classifier, Question and Answers

[13] Weka tool [online] Available http:// www.cs.wai kato. ac.nz /ml/weka.