Performance Analysis of various classifiers using Benchmark Datasets in Weka tools

Nishi Rani[#1], Ravindra Kr. Purwar[*2]

[#]*USICT, GGSIPU, DELHI*
Delhi, India

Abstract*—Intrusion occurs in the network due to redundant and irrelevant data that cause problem in network traffic classification. These kinds of data slow down the network and create difficulties in detecting cyber attacks. Intrusion detection system monitors the network for malicious activities. For network intrusion detection many data mining and machine learning techniques exist in literature but their efficiency has always remain a challenge. In this paper, various classification techniques of weka tool have been studied over a number of datasets like KDD cup 99 dataset, NSL KDD dataset and Kyoto 2006 dataset which can reflect current network stages. KDD cup 99 dataset contains 42 features and can classify intrusion in five classes, NSL KDD is the filtered version of KDD and able to classify intrusion in two classes and Kyoto 2006 data set contains labels as normal (no attack), attack (known attack) and unknown attack which can reflect current stages of the network. We have analyzed the performance of the classification techniques with respect to time and accuracy.*

Keywords *—Intrusion detection, KDD cup 99 dataset, NSL KDD dataset, Kyoto 2006 dataset, weka.*

I. INTRODUCTION

Intrusion is a kind of attack that can occur within the network or between hosts in an organization or it can occur within the organization. Intrusion detection falls under the network security management for. networks and systems (hosts). For detecting intrusions in the networks, it is required to use various machine learning techniques.

These machine learning techniques run various algorithms to detect intrusions in the network packets. In [8], authors described about the features of KDD Cup 99 dataset and also described the traffic data it conatins. Likewise, authors in [6] described about the features of NSL-KDD and the traffic data it contains and also described that NSL KDD has removed the redundant records those were available in KDD Cup 99. In [18] authors gives detailed description of Kyoto 200 dataset and described the values used to analyze the classification.

The datasets contains huge amount of network traffic data, which has very large size, it has been a challenge to detect intrusion in real time.

However, the current researches say that J48 gives accurate results but the time taken by J48 algorithm is very high. Likewise, SMO (Sequential minimal optimization) also gives accurate results but it also takes huge amount of time.

In [11] authors used SOM-based radial basis function (RBF) network for intrusion detection. Results aims at optimizing the performance of the recognition and classification of novel attacks for intrusion detection.

Authors in [11] applied Swarm Intelligence and Ant colony optimisation for intrusion detection and the performance was analyzed.

In [13] author compared four machine learning algorithms i.e., J48, BayesNet, OneR and Naive Bayes (NB) for Intrusion detection and results shows that J48gives more accuracy than other thee algorithms.

In [14] author used clustering algorithms- k-means, mixture of spherical gaussians, SOM (self organizing map) algorithm and investigated multiple centroid-based unsupervised clustering algorithms for intrusion detection. Results presents a simple and effective self-labelling heuristic for detecting attacks and normal clusters of network traffic data.

Furher in [15] authors evaluated Least Square Support Vector Machine based Intrusion Detection System (LSSVM-IDS) using three data sets, namely KDD Cup 99, NSL-KDD and Kyoto 2006 dataset. It has been observed that LSSVM-IDS achieve better accuracy and lower computational cost. Mutual Information (MI) is one of the promising measure in the realm of variable dependence estimation. In [16], a supervised filter-based feature selection algorithm has been proposed, namely Flexible Mutual Information Feature Selection (FMIFS).

M.A. Ambusaidi et al. [16] worked on unsupervised feature selection method for intrusion detection based on MIFS (Mutual Information Feature Selection) and compared its performance against Laplacian score method. The results show that

feature selection works better than laplacian score in terms of classification accuracy. To demonstrate the proposed feature selection algorithms' efficiency, three datasets of intrusion detection have been used for assessment and to compare the performance against Laplacian score method. These three datasets are KDD Cup 99, NSL KDD, Kyoto 2006+ datasets.

Shailendra Sahu et al. [17] used J48 decision tree algorithm to classify the network packet that can be used for network intrusion detection system and results shows that Kyoto 2006 data set can be able to detect unknown attacks. For training and testing purpose 134665 network instances have been used. The rules generated works with correctness of 97.2% for detecting the connection.
Methods proposed in [17] used J48 decision tree algorithm to classify the network packet that can be used for network intrusion detection system and results shows that Kyoto 2006 data set is able to detect unknown attacks and J48 classification technique is an efficient technique.

II. Overview of Classification Techniques

A number of classification techniques are available in weka tool. The classification methods which have been used in this paper are summarized below-

1. BayesNet classifier - Bayesian Network [13] is a statistical model that represents a set of r variables which are random and conditional dependencies through a directed graph that is acyclic (DAG).It represents a probabilistic relationship and based on these relationships it finds out the classes of the network traffic coming. Here, nodes represents random variables and edges shows conditional dependencies.

2. Naive Bayes classifier – Naive bayes are one of a probabilistic classifiers based on Bayes' theorem with strong i.e. naive assumptions among the features and these assumptions are independent. In 1960 [16], it was described under a name into the text retrieval community.

3. Sequential Minimal Optimization (SMO)- It is used in SVM (Support vector machine). It is generally used for solving problems related to quadratic programming. It is implemented by the [12] LibSVM which is a tool used for training of support vector machine. It is an iterative algorithm which picks the multiplier and continue to optimize them until convergence.

4. IBK- It refers to K-nearest neighbor technique.[17] It is instance based algorithm and when k=1, this means object is simply assigned to single nearest neighbor class.

5. J48- It is C4.5 decision tree based algorithm [13]. It is developed as an extension of ID3 algorithm [13] of Ross Quinlan. It is also referred as a statistical classifier and It has ranking #1 in top 10 Data mining algorithms [7]. It is an open source java implementation algorithm available in weka.

6. Random Forest- It is decision tree based algorithm. [18] It is operated by constructing a multitude of decision trees during training time and output is the class that classify. In terms of intrusion detection, the class is anomaly and normal in which anomaly refers to an attack.

III. Overview of Benchmark Datasets

Currently, there are only few number of datasets available publicly for evaluation of intrusion detection. The brief description of these dataset is given in the Table 1 below.

TABLE I : Datasets used for Intrusion Detection

Dataset Name	Attributes	Sample Traffic Data	Classes can be identified
KDD Cup 99 [10]	42	More than 100,000	5
NSL KDD [14]	42	More than 100,000	2
Kyoto 2006 [18]	24	More than 200,000	3

The most popular dataset is KDD cup99 dataset, it is widely used to evaluate the performance of the Intrusion detection systems [15]. It contains 10% training data with approximate five million data connection records and it contains test data with approximate two millions data connection records. KDD is able to identify five classes one as normal and remaining four as different types of attacks (DOS, Probe, U2R, R2L).
The second dataset used is NSL KDD which is a revised version of KDD cup 99 dataset. KDD cup 99 dataset contains large amount of redundant records which are filtered out in NSL KDD dataset. It contains approximately 126,000 and 22,600 training and test connections respectively. It can be classified into two classes anomaly and normal.
The third dataset used is Kyoto 2006 dataset. This dataset covers over three years of real traffic data collected from honeypots [6] and regular servers that are deployed at Kyoto University. It consists of approximately 50,000,000 normal sessions, more

than 40,000,000 attack sessions and more than 420,000 sessions that were unknown attacks. Each connection in this dataset contains 24 features. For our experiments, we have taken samples of data of the 2006 November 01,02,03,04 days.

IV. Experimental Results and Analysis

To demonstrate the performance effectiveness of each classification algorithm, it has been executed for each dataset for fifteen iterations. Number of iteration used are based on the statistics available in the literature. On each iteration, the time parameter required to build each model of the algorithm taken. If the algorithm has to build 10 models then 10 multiply by time to build one model will be the total time.

KDD Cup 99

	Time in seconds					
Algorithm	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Naïve Bayes	4	38.99	4.22	100	126.14	9.49
BayesNet	404.37	24.4	34.07	26.97	25.64	38.1
SMO	610.36	610.29	1019.41	614.59	608.99	606.21
IBK	0.3	0.14	0.14	0.13	0.14	0.31
J48	110.45	102.89	104.81	108.63	110.2	108.29
RandomForest	813.81	475.87	461.95	419.38	436.85	420.24

Fig 1 : Execution time in seconds of different classifiers for KDD Cup 99 for iterations (Run 1-Run 6)

KDD Cup 99

	Time in seconds								
Algorithm	Run 7	Run 8	Run 9	Run 10	Run 11	Run 12	Run 13	Run 14	Run 15
Naïve Bayes	4.17	4.13	4.11	4.14	4.12	4.29	4.41	4.67	4.09
BayesNet	24.7	28.38	25.54	24.24	24.59	25.11	24.02	25.37	24.49
SMO	613.14	614.31	606.79	591.79	586.58	814.68	695.61	795.76	769.17
IBK	0.11	0.72	0.16	3.67	0.2	0.14	0.19	5.24	0.42
J48	111.7	227.95	109.89	109.94	109.44	112	107.95	109.6	108.45
RandomForest	422.12	576.84	502.52	428.54	424.67	423.95	422.33	422.02	421.17

Fig 2 : Execution time in seconds of different classifiers for KDD Cup 99 for iterations (Run 7-Run 15)

The table shown below i.e. Table II is showing min, mean and standard deviation calculated by time in seconds taken by each iteration on each algorithm on KDD cup 99 dataset. Also Correctly classified instances represents the percentage of correct classifications.

TABLE II : Performance Analysis for KDD Cup 99 Dataset using different classifiers

Algorithm	Min	Mean	Standard Deviation	Correctly classified instances
Naive Bayes	4	21.398	38.59044	92.78%
Bayes Net	24.02	51.99933	97.56301	99.67%
SMO	58.58	677.1787	122.4577	99.92%
IBK	0.11	0.800667	1.521289	99.11%
J48	102.89	116.8127	30.83659	99.96%
Random Forest	419.38	471.484	104.2069	99.87%

NSL KDD

	Time in seconds					
Algorithm	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Naïve Bayes	1.95	1.25	1.4	1.34	1.26	1.46
BayesNet	9.67	8.53	8.56	8.69	8.46	8.66
SMO	3005.46	3222.25	2826.34	2125.21	2122.21	2568.45
IBK	0.14	0.09	0.19	0.13	0.14	0.14
J48	61.77	50.41	49.7	49.31	49.55	51.63
RandomForest	135.76	135.2	134.81	135.06	135.68	135.65

Fig 3 : Execution time in seconds of different classifiers for NSL KDD for iterations (Run 1-Run 6)

NSL KDD

	Time in seconds								
Algorithm	Run 7	Run 8	Run 9	Run 10	Run 11	Run 12	Run 13	Run 14	Run 15
Naïve Bayes	1.39	1.42	1.42	1.25	1.37	1.31	1.31	1.44	1.42
BayesNet	8.58	9.05	8.52	8.44	9.02	9.02	8.53	9.45	8.99
SMO	2566.13	2339.14	3100.54	2423.89	2122.21	2008	2001.11	1589.95	1589.54
IBK	0.16	0.17	0.13	0.16	0.17	0.14	0.16	0.17	0.2
J48	49.13	49.2	48.8	48.91	49.3	49.2	49.37	49.02	49.54
RandomForest	134.73	135.15	135.32	135.6	135.25	135.24	135.54	136.53	135.1

Fig 4 : Execution time in seconds of different classifiers for NSL KDD for iterations (Run 7-Run 15)

The table shown below i.e. Table III is showing min, mean and standard deviation calculated by time in seconds taken by each iteration on each algorithm on NSL KDD dataset. Also Correctly classified instances represents the percentage of correct classifications.

TABLE III : Performance Analysis for NSL KDD Dataset using different classifiers

Algorithm	Min	Mean	Standard Deviation	Correctly classified instances
Naive Bayes	1.25	1.399333	0.167821	90.38%
Bayes Net	8.24	8.811333	0.376067	97.17%
SMO	1989.54	2427.375	434.5332	98.43%
IBK	0.09	0.152667	0.027115	99.74%
J48	48.8	50.32267	3.243551	99.78%
Random Forest	134.73	135.3813	0.44089	99.92%

Kyoto 2006						
	Time in seconds					
Algorithm	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Naïve Bayes	1.27	1.05	4.55	3.43	1.05	1.24
BayesNet	8.34	7.08	6.96	6.99	7.37	7.15
SMO	2982.43	2177.45	3423.08	2809.05	2198.32	2809.05
IBK	1.73	3.49	0.19	0.07	0.07	0.07
J48	23.94	23.12	22.64	22.42	22.3	22.59
RandomForest	921.43	901.02	828.09	755.43	762.2	788.58

Fig 5 : Execution time in seconds of different classifiers for Kyoto 2006 for iterations (Run 1-Run 6)

Kyoto 2006									
	Time in seconds								
Algorithm	Run 7	Run 8	Run 9	Run 10	Run 11	Run 12	Run 13	Run 14	Run 15
Naïve Bayes	1.05	1.03	1.09	3.2	1.79	5.47	1.78	5.16	10.07
BayesNet	7.21	7.22	7.29	7.09	7.39	7.16	7.84	7.12	7.12
SMO	2668.2	2809.05	2668.2	2989.15	2265.23	2787.19	2265.23	2787.19	2198.07
IBK	0.07	0.07	0.06	0.06	0.06	0.07	0.06	0.06	0.07
J48	22.44	22.5	23.02	22.85	22.4	22.84	22.84	22.29	22.63
RandomForest	432.1	450.09	432.1	432.1	248.89	432.1	654.45	633.39	432.1

Fig 5 : Execution time in seconds of different classifiers for Kyoto 2006 for iterations (Run 7-Run 15)

The table shown below i.e. Table IV is showing min, mean and standard deviation calculated by time in seconds taken by each iteration on each algorithm on Kyoto 2006 dataset. Also Correctly classified instances represents the percentage of correct classifications.

TABLE IV: Performance Analysis for Kyoto 2006 Dataset using different classifiers

Algorithm	Min	Mean	Standard Deviation	Correctly classified instances
Naive Bayes	1.03	2.882	2.551602	94.11%
Bayes Net	6.96	7.288667	0.359561	97.26%
SMO	2177.45	2655.793	364.9395	98.55%
IBK	0.06	0.413333	0.952393	99.46%
J48	22.29	22.72133	0.422237	99.98%
Random Forest	248.89	606.938	210.7887	99.23%

In the above results, it is shown that these popular algorithms are able to find out more than 90% of correctly classified instances and correspondingly we have shown the min time taken by the algorithm.

V. CONCLUSION

In recent studies, it has been shown that all the classifiers are efficient in terms of results however In our research, we have shown the efficiency of the classifiers in terms of accurate results and execution time also. It has been concluded in this paper that IBK (Instance based K-nearest neighbor) classification technique takes minimum deviation in terms of execution time and also it is able to achieve better results approximately 99% with all three datasets i.e. KDD Cup 99, NSL KDD and Kyoto 2006.

REFERENCES

[1] Altman, N.S. (1992), An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician. 46 (3) : 175-185. DOI: 10. 1080/00031305. 1992. 10475879.

[2] Pearl, Judea (2000), Casuality:Models, Reasoning, and Inference. Cambridge University Press. ISBN 0-521-77362-8. OCLC 42291253.

[3] Russell, Stuart, Norvig, Peter (2003) [1995], Artificial Intelligence : A Modern Approach(2nd ed.), Prentice Hall. ISBN 978-0137903955.

[4] Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[5] Wei Pan and Weihua Li, A hybrid neural network approach to the classification of Novel attacks for intrusion detection, ISPA 2005 LNCS 3758 no 564-575, 2005.

[6] Jungsuk Song, Hiroki Takakura and Yasuo Okabe, Cooperation of Intelligent Honeypots to Detect Unknown Malicious Codes, WOMBAT workshop on Information Security Threat Data Exchange (WISTDE 2008), The IEEE CS Press, Amsterdam, Netherlands, 21-22 April 2008.

[7] Umd.edu- Top 10 Algorithms in Data Mining. Knowl Inf syst (2008) 14:1-37. DOI 10.1007/s10115-007-0114-2.

[8] Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome (2008). The Elements of statistical learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[9] Naeem Seliya, Clustering based network intrusion detection, International Journal of Reliability, Quality and Safety Engineering, Vol. 14, No. 2 (2007) 169-187.

[10] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, A Detailed analysis of the KDD cup 99 dataset, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)

[11] C. Kolias, G. Kambourakis and M. Maragoudakis, Swarm intelligence in intrusion detection: A survey, Computers and Security 30 (2011) 625-642.

[12] Chang, Chih-Chung, Lin, Chih-Jen (2011), LIBSVM :A Library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2(3)

[13] 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, ISSN 2250-3153, Jan 2012.

[14] S. Revathi and Dr. A. Malathi, A Detailed analysis on NSL-KDD dataset using various machine learning techniques for Intrusion detection, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 12, December-2013

[15] M.A. Ambusaidi, Priyadarsi Nanda, Building an intrusion detection system using a filter-based feature selection algorithm, IEEE transactions on computers, Vol no 65, November 2014.

[16] M.A. Ambusaidi, Xiangjian He* and Priyadarsi Nanda, Unsupervised feature selection method for intrusion detection system, IEEE Trustcom/BigDataSE/ISPA, 2015.

[17] Shailendra Sahu, B M Mehtre, Network intrusion detection system using J48 decision tree, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015. Hyderabad, India. 10-13 August, 2015.

[18] Jungsuk SONG, Hiroki Takakura and Yasuo Okabe, Description of Kyoto University Benchmark Data, Academic Center for Computing and Media Studies (ACCMS), Kyoto University