ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Detecting Anomaly Based Network Intrusion Using Feature Extraction and Classification Techniques

Janu Gupta
Department of Computer Science and IT
University of Jammu
Jammu, India

Jasbir Singh
Department of Computer Science and IT
University of Jammu
Jammu, India

Abstract: Computer networks are vulnerable to many kinds of cyber-attacks. It is the responsibility of network administrator to protect the data and resources under threat. Network administrators use various techniques like encryption, firewall, intrusion detection system (IDS), etc. to protect the data and resources of the organizations. Intrusion detection is a topic of research and a lot of work has been done in this field. In the present investigation various machine learning classification techniques were applied to the KDD'99 dataset for building anomaly based intrusion detection system. KDD'99 dataset is used as a benchmark in intrusion detection literature.

Keywords: Machine Learning; Intrusion Detection; Classification; Feature Extraction; KDD'99 dataset.

I. INTRODUCTION

Most of the organizations today have realized the importance of the security of data and resources of their organization over internet. Due to which they have installed devices like firewalls and intrusion detection systems. Firewalls prevent outsiders attack to the network and don't signal an attack from inside the network. An IDS is a device or software application that monitors a network or system for malicious activity or policy violations. Any detected activity or violation is reported to the network administrators. IDSs are of two types: Network Intrusion Detection System(NIDS) and Host Intrusion Detection System(HIDS). A system that monitors important operating system files is called HIDS while a system that analyses incoming network traffic is called NIDS. It is also possible to classify IDS by their detection approach namely signature based and anomaly based detection. Signature based IDS detect intrusions by looking for specific patterns. Although signature based IDS can detect known attacks, they are unable to detect new attacks for which signature is not available. Anomaly based IDS was primarily designed to detect new attacks due to rapid development of malware and cyber attacks. In this paper we presented our model of anomaly based intrusion detection system using various classifiers and we compare their accuracies, training and prediction times. The paper isorganised as follows: section II describes the related work, section III explains the proposed methodology. The results of the experiment is described in the section IV and conclusion is given in section V.

II. RELATED WORK

The rapid growth in the use of internet technologies in various fields have made the network an alluring target for misuse. To detect the anomalous behaviour and misuse the intrusion detection systems are used. Many researchers are engaged in this field of work to analyse current problems that exist in this area.

Ionut Indre et al. [1] touches on the area of cyber security, intrusion detection, intrusion prevention and artificial intelligence. They developed a system capable of detecting and preventing malicious connections using applied concepts of machine learning. They select and extract features that can lead to accurate classification of malware and intrusion.

Ch.Ambedkar et al. [2] presents a detailed analysis of probe attacks by applying various machine learning techniques like naive bayes, svm, multilayer perceptron, decision tree, etc. They used KDD'99 dataset to build the model. In this paper they proposed three layer architecture for detection of probe attacks. PCA is used for dimension reduction. They also removed duplicate samples from the dataset.

Mohammad Khubeb Siddiqui et al. [3] presents an analysis of 10% of KDD'99 training dataset based on intrusion detection. They have focussed on establishing a relationship between the attack types and protocols used by hackers, using clustered data. Analysis of data is performed using K-means clustering. They have used Oracle 10g data miner as a tool for the analysis of dataset and build 1000 clusters to segment the 494021 records. The experiment reveals many interesting results about the protocols and attack types preferred by the hackers for intruding the network.

Mr.Kamlesh et al. [4] analyses various approaches for intrusion detection system using KDD'99 dataset. They explain the various supervised and unsupervised approaches to detect network intrusion.

Alok Pandey et al. [5] presents various TCP/IP attacks that are common in computer networks. Due to design flaws and faulty implementation of TCP/IP protocol suite various vulnerabilities have been reported. Computer networks act as transportation system for different types of attacks or intrusions. They explain various tools and defense mechanisms to identify and mitigate such attacks.

Computer networks are generating large volume of data traffic which cannot be analysed by most of the network intrusion detection system. Muhammad Asif Manzoor et al. [6] developed a system based on support vector machine. They have used Apache Storm framework; which is a real-time distributed stream processing framework. They tested this system on KDD'99 dataset.

III. METHODOLOGY

The proposed methodology is as shown in following figure:

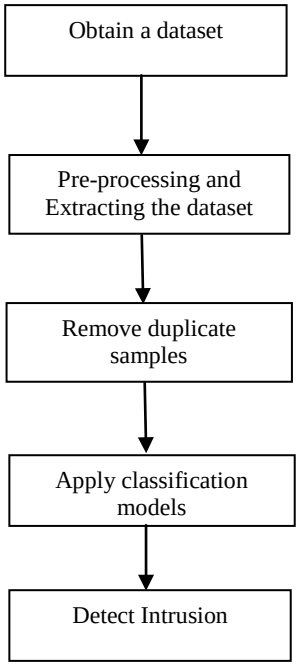

Fig. 1 Methodology

The dataset used for testing our proposed model is KDD'99 [7], [8] network intrusion dataset. Since this dataset is quite large we will use another version of this dataset which is 10% of KDD'99 dataset. This dataset is also used for testing and evaluation of network intrusion detection. The dataset contains both normal and malicious connections. Training dataset contains 23 types of attacks while testing dataset contains 15 more types of attacks in addition to attacks present in training dataset. The attacks are divided into four categories: Denial of Service(DoS), probe, Remote-to-Local(R2L), unauthorized-to-root(U2R). The training dataset contains a total of 494021 samples while the testing dataset contains a total of 311029 samples. The dataset contains 38 numerical features while 3 features are categorical. We converted the categorical features to numeric. Thus we have a total of 41 numerical features describing various connections.

Next we applied various pre-processing techniques and extracted the dataset so that various features can be interpreted by the various classifiers in their training and testing phases. Then we found that there are many duplicate samples in the training and testing datasets. We know from literature that performance of classifiers is biased to samples which are duplicates or are much more in number than the rest class of samples. Now our network data is ready to fed to classifiers. We will analyse the performance of various classifiers on our dataset. The main classifiers used are :

k-nearest neighbours: The principle behind nearest neighbour [10] methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbour learning), or vary based on the local density of points (radius-based neighbour learning). The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbours-based methods are known as non-generalizing machine learning methods, since they simply "remember" all of its training data. Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular.

MLP: Multilayer Perceptron (MLP) [10] is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X = x_1, x_2, ..., x_m$ and a target y, it can learn a non-linear function approximation for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy. MLP requires tuning a number of hyper parameters such as the number of hidden neurons, layers, and iterations. MLP is sensitive to feature scaling.

Decision Trees: Decision Trees (DTs) [10] are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Some of the famous decision tree algorithms are: ID3, C4.5, C5.0, CART, etc. but we are using CART here. Simple to understand and to interpret. Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. Able to handle both numerical and categorical data. Other techniques are usually specialized in analysing datasets that have only one type of variable. See algorithms for more information. Able to handle multi-output problems. Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret. Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Naive Bayes: Naive Bayes methods [10] are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

SVM: Support vector machines (SVMs) [10] are a set of supervised learning methods used for classification, regression and outlier detection. Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples. Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Different Kernel functions can be specified

for the decision function. Common kernels are provided, but it is also possible to specify custom kernels. If the number of features is much greater than the number of samples, the method is likely to give poor performances.

Random Forests: In random forests [10], [11] each tree in the ensemble is built from a sample drawn with replacement from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.

Adaboost: The core principle of AdaBoost [10] is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights w_1, w_2, ...,w_N to each of the training samples. Initially, those weights are all set to $w_i = \frac{1}{N}$, so that examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence.

IV. RESULTS

A. *Experimental Setup*

All the experiments were performed on 2.5 GHz Intel Core i5 processor with 4 GB of RAM. Python 2.7.12 [9] and scikit-learn 0.18.1 [10] is installed and the operating system used is Ubuntu 16.04.

B. *Evaluation and Analysis*

The KDD'99 dataset is loaded in pandas framework in python and various pre-processing techniques is applied like conversion of categorical attributes to numerical attributes obtaining 41 numerical attributes excluding the class label for various connections. The class label of various sample connections is either normal or type of attack name. Class labels are converted in two categories: normal connections and malicious connections then duplicate samples are removed from the dataset. Feature scaling is then applied to bring values of each feature in the range from 0 to 1. Then we fed our training dataset to various classifiers for training and after performing same pre-processing on test dataset we tested our test dataset against trained models. Performance of various classifiers applied is summarized in table I.

TABLE I: ACCURACY, TRAINING AND PREDICTION TIME

Classification Technique	Accuracy	Training Time(seconds)	Prediction Time(seconds)
Decision Tree	.949	1.032	.003
MLP	.9246	20.59	.004
KNN	.9278	82.957	13.14
Linear SVM	.9259	78.343	2.11
Passive Aggressive[12]	.9151	.275	.001
RBF SVM	.9167	99.67	2.457
Random Forest	.9405	1.189	.027
Adaboost	.9352	29.556	.225
GaussianNB	.9435	.244	.006
MultinomialNB	.9171	.429	.001
Quadratic Discriminant Analysis	.9323	1.305	.0019

Accuracy ranges from 0 to 1 where 0 signifies 0% accuracy and 1 signifies 100% accuracy.

Figure 2 shows the comparison of accuracy of all the classifiers used in the experiment. Figure 3 shows the comparison of training and prediction time of all the classifiers used in the experiment.

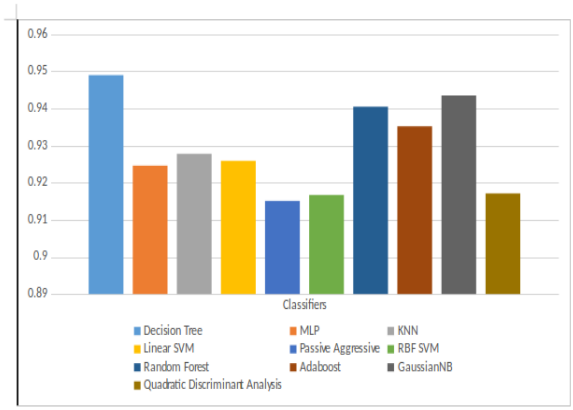

Fig.2 Comparison of accuracies of classification techniques.

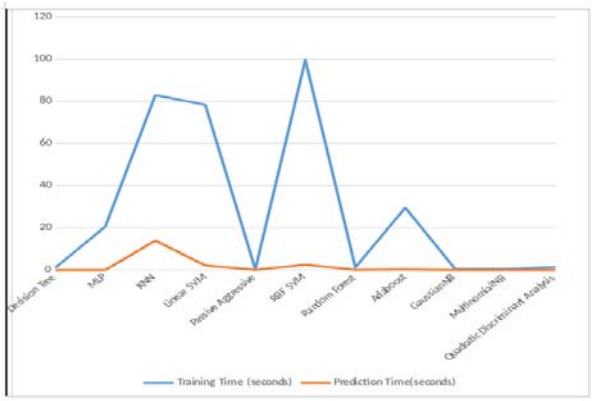

Fig. 3 Comparison of training and prediction time

The confusion matrix of all the classification techniques is obtained. Test dataset after removing duplicates contains a total of 7730 samples. The confusion matrix of various classifiers is as shown in table II.

TABLE III: CONFUSION MATRICES

Classification Techniques	True Positives	True Negatives	False Positives	False Negatives
Decision Tree	4649	2702	279	100
MLP	4728	2419	562	29
KNN	4726	2446	535	23

Linear SVM	4723	2434	547	26
Passive Aggressive	4701	2282	699	48
RBF SVM	4726	2360	621	23
Random Forest	4677	2560	421	72
Adaboost	4676	2553	428	73
GaussianNB	4642	2651	330	107
MultinomialNB	4732	2357	624	17
Quadratic Discriminant Analysis	4677	2530	451	72

From the above table we obtained various metrics for comparing the performance of the classification techniques applied which are described in the table III.

TABLE IIIII: METRICS OBTAINED FROM CONFUSION MATRICES

Classification Techniques	Accuracy	Error Rate	Sensitivity	Specificity	Precision
Decision Tree	95.09	4.90	97.89	90.64	94.34
MLP	92.46	7.54	99.56	81.15	89.38
KNN	92.78	7.22	99.52	82.05	89.83
Linear SVM	92.59	7.41	99.45	81.65	89.62
Passive Aggressive	90.34	9.66	98.99	76.55	87.06
RBF SVM	91.67	8.33	99.52	79.17	88.39
Random Forest	93.62	6.38	98.48	85.88	91.74
Adaboost	93.52	6.48	98.46	85.64	91.61
GaussianNB	94.35	5.65	97.75	88.93	93.36
MultinomialNB	91.71	8.29	99.64	79.07	88.35
Quadratic Discriminant Analysis	93.23	6.77	98.48	84.87	91.20

Figure 4 shows the comparison of metrics of all the classifies used in the experiment.

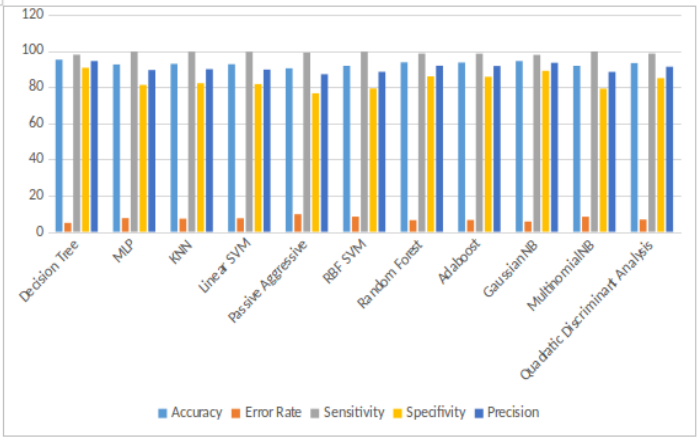

Fig. 4 Comparison of metrics of classification techniques

V. CONCLUSION

After analyzing the results obtained, it was found that various classification techniques can be used effectively to detect network intrusions. Accuracy for all the classification techniques used is more than 90%. Among the classification techniques, the decision tree classifier gives the highest accuracy of almost 95%. Also the training and testing time for decision tree classifier is quite good. In future a hybrid intrusion detection system integrating signature as well as anomaly based detection mechanism will be developed so that known and unknown attacks can be found effectively.

VI. REFERENCES

[1] I. Indre and C. Lemnaru, "Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things", IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP), 2016.

[2] Ch. Ambedkar and V. Kishore Babu , "Detection of probe attacks using machine learning techniques", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) Volume 2, Issue 3, March 2015, pp. 25-29.

[3] Mohammad Khubeb Siddiqui and Shams Naahid, "Analysis of kdd cup 99 dataset using clustering based data mining", International Journal of Database Theory and Application, Vol.6, No.5 (2013), pp.23-34.

[4] Mr. Kamlesh Lahre, Mr. Tarun Dhar, Diwan Suresh,Kumar Kashyap, Pooja Agrawal , "Analyze different approaches for IDS using KDD 99 data set", International Journal on Recent and Innovation Trends in Computing and Communication, ISSN 2321 8169, Volume: 1 Issue: 8 645 651, 2013.

[5] Alok Pandey , Dr. Jatinderkumar and R. Saini, "Attacks and defense mechanisms for TCP/ IP based protocols", International Journal of Engineering Innovation and Research, Volume 3, Issue 1, ISSN: 2277 5668, 2014.

[6] Muhammad Asif Manzoor, Yasser Morgan, "Real-time Support Vector Machine Based Network Intrusion Detection System Using Apache Storm", IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016.

[7] (1999) The UCI KDD Archive University of California [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[8] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, "A detailed analysis of the KDD cup 99 data set" , Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2009.

[9] (2017) Python 2.7 official documentation [Online]. Available: https://www.python.org/

[10] (2016) Scikit-Learn Machine learning in python documentation [Online]. Available: http://scikit-learn.org/documentation.html

[11] Jiong Zhang and Mohammad Zulkernine, "Network intrusion detection using random forests", IEEE Transactions on Systems, Man, and Cybernetics, Volume: 38, 2008.

[12] Koby Crammer ,Ofer Dekel,Joseph Keshet, Shai Shalev-Shwartz, Yoram Singer, "Online passive-aggressive algorithms", Journal of Machine Learning Research 7 (2006) 551585, 2006.