International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Anomaly based Network Intrusion Detection using Machine Learning Techniques.

Tushar Rakshe
Department of Electrical Engineering
Veermata Jijabai Technological Institute,
Matunga, Mumbai.

Vishal Gonjari
Department of Electrical Engineering
Veermata Jijabai Technological Institute,
Matunga, Mumbai.

***Abstract*— In the network communications, network intrusion is the most important concern nowadays. The booming contingency of network attacks is a devastating problem for network services. Various research works are already conducted to find an effective and efficient solution to prevent intrusion in the network in order to ensure network security and privacy. Machine learning is an effective analysis tool to detect any suspicious events occurred in the network traffic flow. In this paper, we developed a classifier model based on SVM and Random Forest based algorithms for network intrusion detection. The NSL-KDD dataset, a much improved version of the original KDDCUP'99 dataset, was used to evaluate the performance of our algorithm. The main task of our detection algorithm was to classify whether the incoming network traffics are normal or an attack, based on 41 features describing every pattern of network traffic. The detection accuracy more than 95 % was achieved using SVM and Random algorithms. The results of two algorithms compared and it is observed that Random Forest algorithm is more effective than Support Vector Machine.**

Keywords*— *Network Intrusion, Support Vector Machine, Random Forest, accuracy.

I. INTRODUCTION

Network Security maintenance is one of the major safety concerns for neutralizing any unwanted activities. It is not only for protecting data and network privacy issues but also for avoiding any hazardous situations. For decades, Network security is one of the major issues and different types of developed systems are being implemented. Network intrusion is an unauthorized activity over the network that steals any important and classified data. Also sometimes it's the reason of unavailability of network services. The unexpected anomaly occurs frequently and a great loss to internet cyber world in terms of data security, the safety of potential information's etc. Therefore, the security system has to be robust, dependable and well configured. Traditionally, network intrusion detection systems (NIDS) are broadly classified based on the style of detection they are using: systems relying on *misuse-detection* monitor activity with precise descriptions of known malicious behavior , while *anomaly-detection* systems have a notion of normal activity and flag deviations from that profile. Signature based detection system involves analyzing network traffic for a series of bytes or packet sequences known to be an anomaly Signature based type detection also has some disadvantages. A signature needs to be created for each attack and they are able to detect only those attacks. They are unable to detect any other novel attacks as their signatures are unknown to the detection scheme. Anomaly based NIDS operate based on the idea that the ambient traffic in a network collected over a period of time reflects the nature of the traffic that may be expected in the immediate future. Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to identify the intrusion. The normal usage patterns are constructed from the statistical measures of the system features, for example, the CPU and I/O activities by a particular user or program. The behavior of the user is observed and any deviation from the constructed normal behavior is detected as intrusion.

Now a days, Machine learning techniques are heavily being adapted and developed in intrusion detection to enhance the efficacy of the systems [1] and in other applications as well [2]. Suthaharan [3] in his work stated that due to the large size and redundant data in the datasets the computation cost of the machine learning methods increases drastically. They proposed ellipsoid-based technique which detects anomalies and side by side cleans the dataset. The research of [4] deals with intrusion detection technique which is a combination of k means clustering, neuro-fuzzy and radial basis support vector machine. In their technique, firstly k-means clustering is used to spawn the training subsets, on them various neuro fuzzy models are trained, after that a vector used by svm classification is generated and finally classification task is carried by radial SVM technique.

We propose a method that is based on the classification algorithm named as random forests and use it to detect the intrusions. Random forest is based on ensemble approach and is closely related to decision trees and nearest neighbor methods that are widely used in the task of intrusion detection. Random forest initiates with decision tree, which can be said to be a weak learner approach. A random forest creates a strong learner by combining trees which were stated as weak learners. Random forest works better than decision trees when the number of samples is more [5]. In random forests features are selected arbitrarily after each split, this ensures a higher classification power and greater efficiency. Moreover, this method overcomes the problem of over fitting and also it not only pertains the qualities possessed by decision trees, but by utilizing its paging mechanism and voting scheme it produces better results than decision trees mostly [6]. In this paper, we present a model that we implemented an intrusion detection system for classification of intrusion types which outperforms the support vector

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

machine method and the nearest centroid classification method in terms of accuracy, the detection rate and false alarm. An analysis has been performed for each type of attack mentioned in the dataset that has been utilized for this study.

2. NSL-KDD Dataset

The dataset to be used in this research is the NSL-KDD dataset [7] which is a new dataset for the evaluation of researches in network intrusion detection system. It consists of selected records of the complete KDD 99 dataset. NSL-KDD dataset solve the issues of KDD 99 benchmark and connection record contains 41 features. Among the 41 features, 34 features are numeric and 7 features are symbolic or discrete. The NSL-KDD training set contains a total of 22 training attack types; with an additional 17 types in the testing set only. Table I gives the description of NSL-KDD Dataset Features.

Table I: Description of NSL-KDD Dataset Features

Feature name	Variable type	Description
Duration	C	No. of seconds of the connection
Protocol_type	D	Type of protocol Eg.TCP,UDP ,ICMP
Service	D	Network service on the destination eg:http,telnet,etc
Flag	D	Normal or error status of the connection
src_bytes	C	Number of data bytes from source to destination
dst_bytes	C	Number of data bytes from destination to source
Land	D	1-connection is from the same host/port: 0-otherwise
Wrong_fragment	C	No. of 'wrong' fragments
Urgent	C	No of urgent fragments
Hot	C	The count of access to system directories, creation and execution of programs
Num_failed_logins	C	No. of failed login attempts
Logged_in	D	1-successfully logged in 0-otherwise
num_compromised	C	No. of compromised conditions
Root_shell	C	1-root shell is obtained;0 otherwise
Su_attempted	C	1-'su root' command attempted;0 otherwise
Num_root	C	No .of root accesses
num_file_creations	C	Number of file creation operations
Num_shells	C	No of shell prompts
Num_access_files	C	No. of write ,delete and create operations on access control files
Num_outbound_cmds	C	No. of outbound commands in an ftp session
Is_hot_login	D	1-the login belongs to the 'hot' list 0: otherwise
Count	C	No. of connections to the same host as the current connection in the past seconds
Srv_count	C	No of connections to the same host as the current connection in the past 2 seconds
serror_rate	C	% of connections that have 'SYN' errors to the same host
Srv_serror_rate	C	% of connections that have 'SYN' errors to the same service
Rerror_rate	C	% of connections that have 'REJ' errors to the same host
Srv_diff_host_rate	C	% of connections to different services and to the same host
Dst_host_count	C	No of connections to the same host to the destination host as the current connection in the past 2 seconds
Dst_host_srv_count	C	No of connections from the same service to the destination host as the current connection in the past 2 seconds
dst_host_srv_count	C	No. of connections from the same service to the destination host as the current connection in the past 2 seconds
Dst_host_srv_count	C	No. of connections from the same service to the destination host as the current connection in the past 2 seconds
Dst_host_same_srv_rate	C	% of connections from the same service to the destination host
Dst_host_diff_srv_rate	C	% of connections from the different services to the destination host
Dst_host_same_src_port_rate	C	% of connections from the port services to the destination host
Dst_host_srv_diff_host_rate	C	% of connections from the different hosts from the same service to destination host
Dst_host_serror_rate	C	% of connections that have 'SYN" errors to same host to the destination host
dst_host_srv_serror_rate	C	% of connections that have 'SYN' errors from the same service to the destination host
Dst_host_rerror_rate	C	% of connections that have 'REJ' errors from the same host to destination host
Dst_host_srv_rerror_rate	C	% of connections that have 'REJ' errors from the same service to the destination host

NSL – KDD Dataset Preprocessing:
Classification algorithms are not able to process NSL - KDD dataset in its current format.
Hence we need to preprocess the datasets before training the model.
Preprocessing contains below steps:

- Mapping symbolic features to numeric value.
- Implementing scaling since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [-1, 1].
- Attack names were mapped to one of the two classes, 0 for Normal, 1 for Attack.
- Missing values in data.

3. CLASSIFICATION MODEL

In general, the category of problems which contains data as well as the additional attributes that we want to predict comes under supervised learning approach. Under supervised learning approach the classification problem comes into

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

account when the instances belong to two or more classes and our intention is to forecast the class of the unlabeled instances. Under the category of supervised learning methods, a technique known as Support vector machines (SVM) holds its place for classification. This method is effective for high dimensional spaces, is memory efficient since it utilizes subset of training data points in the decision function called as support vectors, also it is adroit as for the decision function various kinds of kernel functions can be stated . If the count of features is bigger than the count of samples this technique is liable to give mediocre performance.

A) Support Vector Machine:

The SVM uses a portion of the data to train the system, finding several support vectors that represent the training data. These support vectors will form a SVM model. A basic input data format and output data domains are listed as follows

(Xi, Yi)……Xn, Yn)

Where

$X \in R^m$ and $Y \in \{0, 1\}$

(Xi, Yi)……. (Xn, Yn) is training data records, n is the numbers of samples m is the inputs vector, and y belongs to category of class '0' or class '1' respectively. On the problem of linear, a hyper plane can be divided into the two categories as shown in Figure.

The hyper plan formula is:

$$(w \cdot x) + b = 0$$

The category formula is:

$$(w. x) + b \geq 0 \text{ if } Yi = 1$$
$$(w. x) + b \leq 0 \text{ if } Yi = 0$$

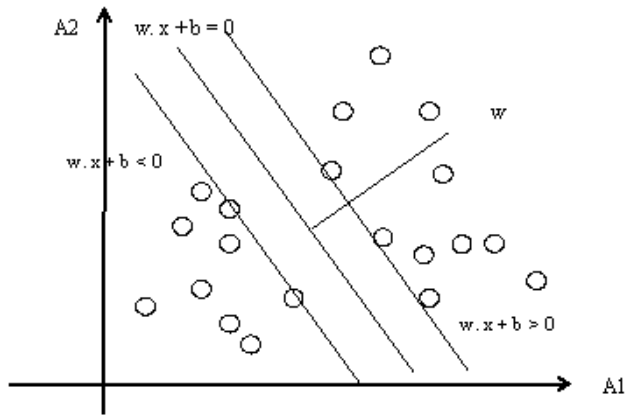

Figure 1-Classifiaction using of SVM

A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one "target value" (class labels: Normal or Attack) and several "attributes" (features).The goal of SVM is to produce a model which predicts target value of data instance in the testing set which is given only attributes. To attain this goal there are four different kernel functions.in this experiment RBF kernel function is used

The Formula for RBF Kernel Optimization function :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Finding vectors from training data is formulated as

Minimize : $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi$

B) Random Forest Classification Model:

The suggested classification model is comprised of random forest algorithm. In Random Forest, the method to build an ensemble of classifiers is to change the training set using the same strategy as bagging (Breiman, 1996). Bagging creates new training sets by resampling from the original data set n times, n being the number of samples in the original training set, randomly with replacement. This means the sample just being chosen will not be removed from the data set in the next draw. Hence, some of the training samples will be chosen more than once while some others will not be chosen at all in a new set. Bagging helps classification accuracy by decreasing the variance of the classification errors. In another words, it taps on the instability of a classifier. 'Instability' of a classifier means that a small change in the training samples will result in comparatively big changes in accuracy. The classifiers are combined by a majority vote and the vote of each classifier carries the same weight. In the case of a tie, the decision can be made randomly or by prescribed rules. Random Forest creates multiple trees using the impurity gini index (Breiman et al., 1984). However, when constructing a tree, Random Forest searches for only a random subset of the input features (bands) at each splitting node and the tree is allowed to grow fully without pruning. Since only a portion of the input features is used and no pruning, the computational load of Random Forest is comparatively light. In addition, in case a separate test set is not available, an out-of-bag method can be used. For each new training set that is generated, one-third of the samples are randomly left out, called the out-of-bag (OOB) samples. The remaining (in-the-bag) samples are used for building a tree. For accuracy estimation, votes for each sample are counted every time when it belongs to OOB samples. A majority vote will determine the final label. Only approximately one-third of the trees built will vote for each case. These OOB error estimates are unbiased in many tests (Breiman, 2001). The number of features for each split has to be defined by the user, but it is insensitive to the algorithm. Majority vote is used to combine the decisions of the ensemble classifiers.

The Algorithm:

The random forests algorithm (for both classification and regression) is as follows:

1. Draw n_{tree} bootstrap samples from the original data.

2. For each of the bootstrap samples, grow an unpruned classification or regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m_{try} of the predictors and choose the best split from among those variables. (Bagging can be thought of as the special case of random forests obtained when m_{try} = p, the number of predictors.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

3. Predict new data by aggregating the predictions of the n_{tree} trees (i.e., majority votes for classification, average for regression).

4. RESULT AND DISCUSSION:

The performance of all the classifiers was computed by utilizing a matrix known as confusion matrix. It is a standard metric for benchmarking the effectiveness and robustness of a classification algorithm. Using the confusion matrix, measures like accuracy, detection rate and false alarm rate have been computed which are the generic criteria for evaluating the performance of the IDS. These metrics have been utilized in a number of studies and they ensure a viable means of deciding the efficiency of the model for detecting the intrusions within systems. For a decent level of performance, the intrusion detection system (IDS) needs high accuracy and precision and conversely false alarm rate should be low. These terms are given by the following formulae:

Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$

Precision = $\frac{TP}{TP+FP}$

True positive rate (TPR) = $\frac{TP}{TP+FN}$

False positive rate (FPR) = $\frac{FP}{TN+FP}$

True negative rate TNR) = $\frac{TN}{TN+FP}$

False negative rate (FNR) = $\frac{FN}{TP+FN}$

Following figure represents a matrix known as confusion matrix. True positive (TP) indicates the number of instances having the class label of attack and were correctly classified as an attack. True negative (TN) indicates the number of instances having the class label of normal and were correctly classified as normal. False positive (FP) indicates the number of instances that have a label of being valid but have been incorrectly classified as intrusion. False negative (FN) indicates the number of instances that were having a label of intrusion but were incorrectly classified as normal by the IDS.

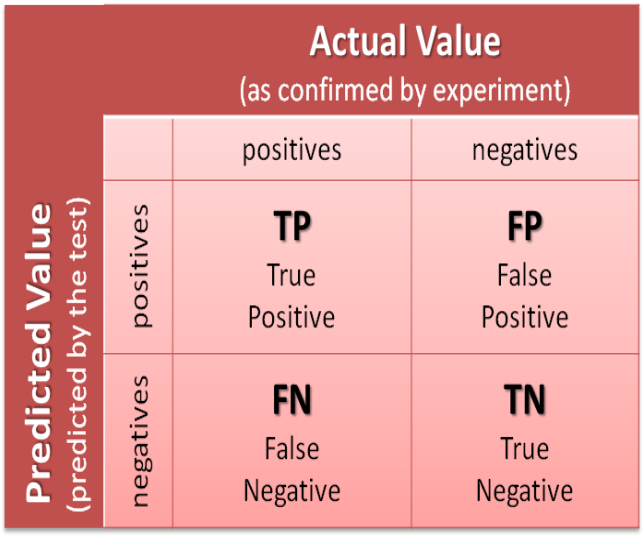

Fig. Confusion matrix

Experimental Analysis:

Following figure shows the prediction result of SVM and Random Forest method

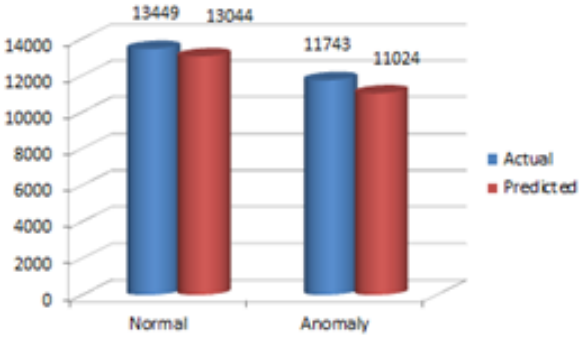

Figure 2- Prediction Result of Support Vector Machine

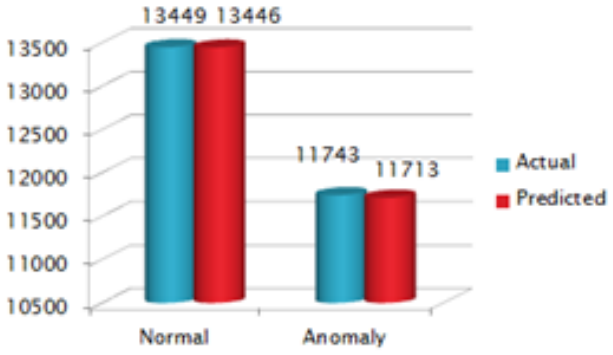

Figure 2- Prediction Result of Random Forest Prediction

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

Table II: Comparison of SVM and Random Forest classification model based on performance measure.

Algorithm	True Positive rate	False Positive rate	True Negative rate	False Negative rate
SVM	93.87%	3.01%	96.98%	6.12%
Random Forest	99.74%	0.026%	99.97%	0.24%

Algorithm	Accuracy	Precision
SVM	95.53 %	96.45%
Random Forest	99.86 %	99.96 %

5. CONCLUSION:

In this paper, we have scrutinized some new techniques for intrusion detection and evaluated their performance based on the benchmark KDD Cup 99 Intrusion data. An Intrusion Detection System that was able to assay the dynamic and complex nature of intrusion activities has been built. Random forests classification algorithm surpass the major classification methods support vector machine. After comparison of models, the proposed model resulted in the highest accuracy and detection rate values as well as the least false rate values. The results specify that the classification ability of the proposed model is inherently superior to the support vector machine model. Anomaly detection methods that are based on artificial intelligence are continuously alluring a lot of attention from the research community. The experimental result shows the efficiency of both Random forest and Support Vector Machine, which proves that the machine learning techniques can be successfully applied to the Anomaly Intrusion Detection System. The research work can be extended by applying various other soft computing techniques in Anomaly Intrusion Detection.

6. REFERENCES:

[1] S. S. Roy, V. M. Viswanatham - Classifying Spam Emails Using Artificial Intelligent Techniques. In International Journal of Engineering Research in Africa, vol. 22, pp. 152-161. Trans Tech Publications, 2016.

[2] S. S. Roy, D. Mittal, A. Basu, A. Abraham - Stock Market Forecasting Using LASSO Linear Regression Model. In Afro-European Conference for Industrial Advancement, pp. 371-381. Springer International Publishing, 2015.

[3] S. Suthaharan - An iterative ellipsoid-based anomaly detection technique for intrusion detection systems, In Southeast on, Proceedings of IEEE, pp. 1-6, 2012.

[4] A. Chandrasekhar, K. Raghuveer - Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers, In Computer Communication and Informatics (ICCCI), 2013 International Conference, pp. 1-7.

[5] S. Adusumilli, D. Bhatt, H. Wang, V. Devabhaktuni, P. Bhattacharya - A novel hybrid approach utilizing principal component regression and random forest regression to bridge the eripod of GPS outages, Neurocomputing, 2015.

[6] J. Ali, R. Khan, N. Ahmad, I. Maqsood - Random forests and decision trees, IJCSI International Journal of Computer Science Issues, vol. 9, no. 5, 2012.

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

[8] Breiman, L. (1996). Bagging predictors. Machine Learning, 24, 123−140.

[9] Breiman, L. (2001). Random forests. Machine Learning, 45, 5−32.

[10] Breiman, L., Freidman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Wadsworth.