ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 5, May 2019

Network Intrusion Detection by Evaluating Machine Learning Techniques

Archana Borse[1], Khushbu Nemade[2], Harshali Patil[3], Komal Patil[4]

U.G. Student, Department of Computer Engineering, SSBT's COET Bambhori, Jalgaon, Maharashtra, India[1-4]

ABSTRACT: Network traffic anomaly may indicate a possible intrusion in the network and therefore anomaly detection is important to identify and prevent the security attacks. The early re-search work in this area and commercially available Intrusion Detection Systems (IDS) are mostly signature-based. The current trend in anomaly detection is based on machine learning classification techniques. In this paper, we apply three different machine learning techniques with information entropy calculation on KDD-cup-99 data set and evaluate the performance of these techniques. It shows that, for this particular data set, most machine learning techniques provide higher than 90 percent precision, recall and accuracy.

KEYWORDS: Network Intrusion Detection System, machine learning techniques, precision, recall and accuracy

I. INTRODUCTION

Intrusion Detection is the act of detecting unwanted traffic on a network or device. A device or software application that ensures a network or systems for malicious activity or policy violations is called as Intrusion Detection System. An IPS(Intrusion Prevention System) is also a type of IDS that can prevent or stop unwanted traffic. e.g Fail2Ban, Snort, Bro, Suricata, Security Onion etc..

Network traffic anomaly detection is aimed to identify and prevent the attack. For that, the detection technique has to be very powerful. The early re-search work in this area and commercially provided Intrusion Detection Systems (IDS) are mostly signature-based. The drawback of signature based method is that the database signature needs to be updated as the new signatures become available and therefore it is not appropriate for the real-time network anomaly detection. Therefore, advanced machine learning techniques are required to detect new type of anomalies.

In current years, machine learning is becoming popular. To detect network anomaly using machine learning techniques has been researched by many people. In this paper, we use three techniques on a well-known data set KDD-cup-99 and evaluate the performance of those machine learning techniques in terms of precision, recall and accuracy. The three techniques adopted are: Naive-Bayes (NB), Decision Tree and hybrid approach (NBTree) which is a combination of both Naïve Bayes and Decision Tree algorithms. These algorithms are all evaluated using well-known metrics called precision, recall and accuracy.

II. RELATED WORK

Ren et al. [1] put forward Fuzzy C-Means clustering algorithm for intrusion detection. The algorithm was applied on six different subsets of KDD Cup 1999 data set with 5000 records each. The detection rate varies between 50.3% and 90.5% whereas the false positive rate ranges between 0.2% and 4.1%. Clustering methods are commonly used for anomaly detection. Syarif *et al.* [2] proposed and discussed five different anomaly detection techniques. The authors used NSL-KDD data set for the evaluation of clustering algorithms in network anomaly detection. Wang [3] also presented an improved K-Means algorithm to overcome the sensitivity problem of initial center selection. The basic idea was to choose the initial centers as decentralized as possible. The improved algorithm was applied on KDD Cup 1999 data set. The work done by Mulay *et al.* [4] involves multiclass intrusion detection using support vector and decision tree. Their work was also evaluated on KDD Cup'99 data set. Li and Wu [6] introduced an improved clustering algorithm based on information entropy and frequency sensitive discrepancy metric. These two metrics are used to place the initial centers of the clusters. The authors used KDD Cup 1999 data set. Their improved algorithm yields 98.3% detection rate in case of DOS attack. Zhu and Liao [7] recommended a SVM algorithm for intrusion detection based on space block and sample density. Their main contribution lies in developing an algorithm to reduce the sample size and thereby the learning speed. The SVM model works on the reduce sample set. The authors selected 100,000 records from DARPA data set and used Radial Basis Function (RBF) as the kernel function for the SVM. This improved SVM works with better accuracy and learning speed than the traditional SVM algorithm. One of the major issues in intrusion detection research is to find good labelled data sets which contain representatives of different types of intrusion and normal traffic data. Song et al. [8] in their research paper described how honey pots can be used effectively to collect information to prevent zero day attack can be eliminated. The authors evaluated in details two different honey pot systems which are deployed in Kyoto University, Japan. Using several honey pots, the researchers collected a very well-known data set called Kyoto 2006+. This data set uses 14 features which were part of the KDD Cup 99 data set – another popular and widely used data set for network security researchers.

III. DATA SET USED

The UCI machine learning repository [AN07] is one of the most universal archives for the machine learning community. The data set used for evaluation in this report is a subset of the KDD-Cup-99 data set for intrusion detection obtained from the UCI machine learning repository. It contains a standard set of data which includes a wide variety of intrusions simulated in a military network environment [OC99c]. KDD training data set contains approximately 4,900,000 single connection vectors. Each connection vectors contains 41 features and that are labeled as either normal or an attack, with having exactly one specific attack type. The attacks fall into one of the following categories:

- DOS attacks (Denial of Service attacks)
- R2L attacks (unauthorised access from a remote machine)
- U2R attacks (unauthorised access to super user privileges)
- Probing attacks

IV. PROPOSED SYSTEM

Classification is a classic data mining technique based on the concepts of machine learning. General applications of classification is used it as a tool to categorizes item in a set of data into one of predefined set of classes or groups. Classification—A Two-Step Process

1. Training: describing a set of predetermined classes. Each sample is supposed to belong to a predefined class, as determined by the class label attribute. The set of tuples used for model construction of training set. On training data we apply algorithms after that it constructs the model. The constructed model is represented as classification rules, decision trees, or mathematical formula.

2. Classification: for classifying future or unknown objects. Estimate accuracy of the model. The known label of test sample is compared with the classified result from the model. The percentage of test set samples that are correctly classified by the model is accuracy. To avoid over-fitting problem, testing data should be independent of training data.

The following figure shows the steps of classification.

DOI: 10.15680/IJIRCCE.2019. 0705036

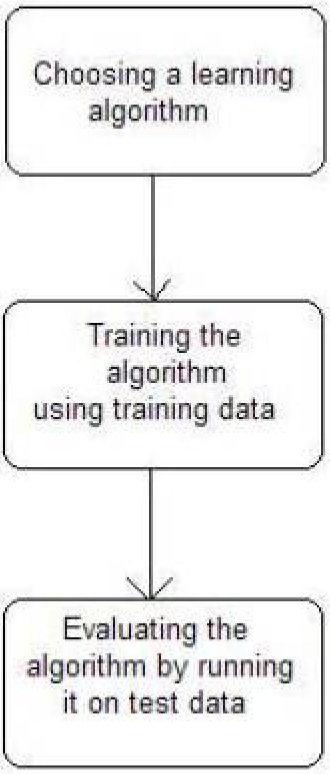
Choosing a learning algorithm
Training the algorithm using training data
Evaluating the algorithm by running it on test data

Fig. Classification Steps

The proposed system uses the following three algorithms :

Algorithm 1: Decision Tree Algorithm

```
Input: Training Dataset = D
Output: Decision Tree = T
DTBUILD (*D)
1. T= φ;
2. T= Create root node and label with splitting attribute;
3. T= Add arc to root node for each split predicate and label;
4. For each arc D= Database created by applying splitting predicate to D;
    • If stopping point reached for this path, then T= create leaf node and label with appropriate class;
    • Else T= DTBUILD (D); T= add T to arc;
```

Algorithm 2: Naive Bayes Algorithm

```
 INPUT :
   • Set of tuples = D
   • Each Tuple is an n dimensional attribute vector
   • X : (x1,x2,x3,. xn)
Let there be m Classes: C1, C2, C3Cm Naive Bayes classifier predicts X belongs to Class Ci iff
   • Maximum Posteriori Hypothesis
          P(Ci=X) > P(Cj=X)for1 <= j <= m; j <> i
   • P(Ci/X) = P(X/Ci) P(Ci) / P(X)
```

- Maximize P(X/Ci) P(Ci) as P(X) is constant

With many attributes, it is computationally expensive to evaluate P(X/Ci).
Naive Assumption of class conditional independence
P (X/Ci) = $\prod_{k=l}^{n} P(xk/Ci)$
P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…* P(xn/ Ci)

Algorithm 3: NBTree Algorithm (Mixed Approach)
NBTree algorithm is a hybrid between decision-tree algorithm and Naive Bayes classifiers.
The learned knowledge is represented in the form of a tree.
Input: a set of T labeled instances
Output: a decision-tree with Naive Bayes categorizers at leaves
1. For each attribute Xi, evaluate the utility u(Xi), of a split on attribute Xi. For continuous attributes, a threshold is also found out at this stage.
2. Let j = argmaxi(ui), i.e. the attribute with the highest utility.
3. If uj is not significantly better than the utility of the current node, create a Naive Bayes classifier for the current node and return.
4. Partition T according to the test on Xj. If Xj is continuous, a threshold split is used; if Xj is discrete, a multi-way split is made for all possible values.
5. For each child, call the algorithm recursively on the portion of T that matches the test leading to the child.

V. RESULTS

The first step is to find the number of instances of sample dataset using Naïve Bayes , Decision Tree and NBTree algorithm. In the next step of the experiment we will calculate the precision, recall and accuracy of three algorithms. For each algorithm we visualize the confusion matrix which shows the following widely used raw metrics.

1. True Positive (TP) : it means that the actual anomaly is predicted as anomaly
2. False Positive (FP) : it means that the actual anomaly is predicted as normal
3. True Negative (TN) : it means that the actual normal is predicted as normal
4. False Negative (FN) : it means that the actual normal is predicted as anomaly

Based on the above metrics we calculate the following measures :

- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)
- Accuracy = (TP+TN)/(TP+TN+FP+FN)

The following table shows the experimental results and performance of three machine learning algorithms calculated on various measures like precision, recall and accuracy. The results obtained for NBTree are much better than that of Naive Bayes (NB) and Decision Tree algorithms as can be seen from in table.

Algorithms	Precision	Recall	Accuracy
Naïve Bayes	88%	53%	55%
Decision Tree	95%	54%	60%
NBTree	90%	71%	70%

Table : Results

VI. CONCLUSION AND FUTURE WORK

This work presented three commonly using machine learning techniques for network traffic anomaly detection. We compared those methods and obtained the preliminary results using the sample data from KDD-cup-99 data set. The NBTree technique worked the best with an accuracy value of 70%.

Research would be interesting on real time sensitive data sets that have highest changes if getting intruder. Reduce the processing time when large amount of data set is provided.

REFERENCES

1. W. Ren, J. Cao, X. Wu, "Application of Network Intrusion Detection Based on Fuzzy C-Means Clustering Algorithm", *Proc. of the 3rd International Symposium on Intelligent Information Technology Application, 2009*.
2. Syarif I, Prugel Bennett A, Wills G., "Unsupervised clustering approach for network anomaly detection", *Networked Digital Technologies Communications in Computer and Information Science, vol. 293. Berlin Heidelberg: Springer, 2012, pp.135–45*.
3. S. Wang, "Research of Intrusion Detection Based on an Improved K-means Algorithm", *Proc. of the 2nd International Conference on Innovations in Bio-inspired Computing and Applications, 2011*.
4. S.A. Mulay, P. R. Devale, G.V. Garje, "Intrusion Detection System using Support Vector Machine and Decision Tree", *International Journal of Computer Applications, vol. 3, no. 3, 2010*.
5. L. Hu, T. Li, N. Xie, J. Hu, "False Positive Elimination in Intrusion Detection Based on Clustering", *Proc. of the 12th International Conference on Fuzzy Systems and Knowledge Discovery, 2012*.
6. H. Li and Q. Wu, "Research of Clustering Algorithm based on Information Entropy and Frequency Sensitive Discrepancy Metric in Anomaly Detection", *Proc. of the International Conference on Information Science and Cloud Computing Companion, 2013*.
7. G Zhu and J. Liao "Research of Intrusion Detection Based on Support Vector Machine", *Proc. of the International Conference on Advanced Computer Theory and Engineering, 2008*.
8. J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "*Statistical analysis of honeypot data and building of Kyoto 2006+ data set for NIDS evaluation*", *Proc. of the 1st Workshop on Building Analysis Data Sets and Gathering Experience Returns for Security*, 2011, pp. 29–36.
9. Z. Muda, W. Yassin, M.N. Sulaiman and N.I. Udzir, "A K-Means and Naïve Bayes Learning Approach for Better Intrusion Detection", *Information Technology Journal, 2011*.
10. X. Bao, T. Shu and H. Hau, "Network Intrusion Detection Based on Support Vector Machine", *Proc. of the International Conference on Management and Service Science, 2009*.
11. UCI Machine Learning Repository.