ISSN:0975-9646

Anomaly Detection in Network using Genetic Algorithm and Support Vector Machine

[1]Prashansa Chouhan and [2]Dr.Vineet Richhariya

[1]*M.tech(CSE),LNCT Affiliated to RGPV Bhopal*
[2]*HOD, CSE LNCT Affiliated to RGPV Bhopal*

Abstract- An anomaly is a abnormal activity or deviation from the normal behaviour .Anomaly detection is the process of removing these abnormal or anomalous behaviour from data or services. In this paper, we present a novel method for the detection of anomaly in network. The proposed detection algorithm, is a hybrid algorithm. It is combination of two algorithm genetic and SVM. Experimental results demonstrate to be superior than existing k-mean algorithm. One of the most common problems in existing K means detection techniques is that one must specify the clusters in advance and further the algorithm is very sensitive of noise, mixed pixels and outliers. The definition of means limit the application to only numerical variables. It is data driven with relatively few assumptions on the distributions of underlying data. This paper investigates the performances of genetic algorithm (GA) with support vector machine (SVM) classification method for detecting different types of network attacks. . Genetic based feature selection algorithm reduces the 41 features of the KDD cup dataset into 9 important features by applying fitness value as a threshold and then these 9 features are used for classification using support vector machine. In this work GA and SVM have been implemented and tested on KDD CUP 1999 dataset. Our method has more accurate as compare to existing once.

Keywords-Anomaly detection techniques, clustering, CAD, genetic and classification based technique.

1. INTRODUCTION

The number of hacking and intrusions incidents is increasing year on year as technology rolls out. Maintaining a high level security to ensure safe and trusted communication of information between various organizations becomes a major issue. So Intrusion detection system (IDS) has become a needful component in terms of computer and network security [11]. An Intrusion Detection system (IDS) is a device or a software product that analyzes the coming traffic on network for a malicious activities (or intrusion) and raises an alarm when intrusion detected. The aim of IDS is to detect illegal and improper use of system resources by unauthorized users by monitoring network traffic and audit data. An intrusion can be defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of resources on system [12].

Anomaly detection refers to detecting patterns in a given data set that do not conform to an established normal behavior. The patterns thus detected are called anomalies and translate to critical and actionable information in several application domains. Anomalies are also referred to as outlier, surprise deviation etc [9].

Most anomaly detection algorithms require a set of purely normal data to train the model and they implicitly assume that anomalies can be treated as patterns not observed before. Since an outlier may be defined as a data point which is very different from the rest of the data, based on some measure, we employ several detection schemes in order to see how efficiently these schemes may deal with the problem of anomaly detection. The statistics community has studied the concept of anomaly quite extensively [5]. In these techniques, the data points are modeled using a stochastic distribution and points are determined to be outliers depending upon their relationship with this model. However with increasing dimensionality, it becomes increasingly difficult and inaccurate to estimate the multidimensional distributions of the data points.

2. RELATED WORK

A. Supervised Approaches

In this approach, a predictive model is developed based on a training dataset (i.e., data instances labeled as normal or attack class). Any unseen data instance is compared against the model to determine which class it belongs to. There are two major issues that arise in supervised anomaly detection. First, the anomalous instances are far fewer in number compared to normal instances in the training data. Issues that arise due to imbalanced class distributions have been addressed in the data mining and machine learning literature [6]. Second, obtaining accurate and representative labels, especially for the anomaly class is usually challenging. A number of techniques have been proposed that inject artificial anomalies in a normal dataset to obtain a labeled training dataset [7]. Other than these two issues, the supervised anomaly detection problem is similar to building predictive models. We now discuss some of the most common incremental supervised anomaly detection approaches. The authors propose a new anomaly detection algorithm that can update the normal profile of system usage dynamically [2]. The features used to model a system's usage pattern are derived from program behavior. A new program behavior is inserted into old profiles by density-based incremental clustering when system usage pattern changes. It is much more efficient compared to traditional updating by re-clustering. The authors test their model using the 1998 DARPA BSM audit data, and report that the normal profiles generated by their algorithm are less sensitive to noise data objects than profiles generated by the ADWICE algorithm. The method improves the quality of clusters and lowers the false alarm rate.

B. Semi-supervised Approaches

In semi-supervised approach, the training data instances belong to the normal class only. Data instances are not labeled for the attack class. There are many approaches used to build the model for the class corresponding to normal behavior. This model is used to identify anomalies in the test data. Some of the detection methods are discussed in the following.

ADWICE (Anomaly Detection With fast Incremental Clustering) uses the first phase of the BIRCH clustering framework [6] to implement fast, scalable and adaptive anomaly detection[3]. It extends the original clustering algorithm and applies the resulting detection mechanism for analysis of data from IP networks. The performance is demonstrated on the KDD99 intrusion dataset as well as on data from a test network at a telecom company. Their experiments show good detection quality (95%) and acceptable false positives rate (2.8 %) considering the online, real-time characteristics of the algorithm. The number of alarms is further reduced by application of the aggregation techniques implemented in the Safeguard architecture.

It is important to increase the detection rate for known intrusions and also to detect unknown intrusions at the same time [4]. It is also important to incrementally learn new unknown intrusions. Most current intrusion detection systems employ either misuse detection or anomaly detection. In order to employ these techniques effectively, the authors propose an incremental hybrid intrusion detection system. This framework combines incremental misuse detection and incremental anomaly detection. The framework can learn new classes of intrusion that do not exist in data used for training. The framework has low computational complexity, and so it is suitable for real-time or on-line learning. The authors use the KDDcup99 intrusion dataset to establish this method.

C. Unsupervised Approaches

Unsupervised detection approaches do not require training data, and thus are most widely applicable. These techniques make the implicit assumption that normal instances are far more frequent than anomalies in the test data. If this assumption is not true, such techniques suffer from high false alarm. Most existing unsupervised anomaly detection approaches are clustering based. Clustering is a technique to group similar objects. It deals with finding structure in a collection of unlabeled data. Representing the data by fewer clusters necessarily leads to the loss of certain finer details, but achieves simplification. In anomaly detection, clustering plays a vital role in analyzing the data by identifying various groups as either belonging to normal or to anomalous categories. There are many different clustering based anomaly detection approaches in the literature. [1] describe on collective anomaly detection and clustering anomaly which are generated due to variety of abnormal activities such as credit card fraud detection, mobile phone fraud, banking fraud, cyber attack etc. an important aspect as the nature of anomaly. In existing paper introduced the concept of collective anomaly for network traffic analysis. It's used the variant of k-mean and x-mean algorithm for clustering network traffic and detects DOS attack. **[8]** describe on genetic algorithm and classification algorithm for anomaly detection intrusion detection system using soft computing techniques to offer effective security through the provision of detection accuracy, fast processing time, ability to adapt and exhibit fault tolerance. In this paper, intelligent algorithms for intrusion detection are proposed which detect the network attacks as normal or anomaly based attacks by performing effective preprocessing and classification. This system uses a new genetic algorithm approach for pre-processing and Modified J48 classification algorithm to identify the intended activities[10]. The new genetic based feature selection algorithm proposed in this paper is helpful to identify the important features needed to classify the normal and anomaly records. For this, we propose a new genetic based feature selection algorithm which reduces the 41 features of the KDD Cup data set into 9 important features by applying the fitness value as a threshold. Moreover, we perform classification using a modified decision tree algorithm which has been developed by enhancing the existing J48 decision tree algorithm. 99 dataset suffers from major weakness due to the presence of redundant records. These to redundant records reduce the detection rate and accuracy. KDD'99 dataset has 41 features with classes labeled as either normal or anomaly with specific attack type.

3. PROBLEM FINDING

A major disadvantage of K means is that one must specify the clusters in advance and further the algorithm is very sensitive of noise, mixed pixels and outliers. The definition of means limit the application to only numerical variables. It is data driven with relatively few assumptions on the distributions of underlying data.

Based on our survey of published papers on incremental anomaly detectors, we observe that most techniques have been validated using the KDD99 intrusion datasets in an offline mode. However, the effectiveness of an ANIDS based on incremental approach can only be judged in a real-time environment. The clustering techniques that are used by anomaly detectors need to be faster and scalable when used on high dimensional and voluminous mixed type data. We overcome these all problem through hybrid algorithm.

4. PROPOSED SOLUTION AND ALGORITHM

We focus on a Machine Learning Model using a modified **Support Vector** Machine **(SVM)** that combines the benefits of supervised and unsupervised learning. Moreover, we provide a preliminary feature selection process using **GA** to select more appropriate packet fields. Now, we discuss our hybrid algorithm steps which are as follow:

Step 1 - firstly load kdd dataset.

Step 2- Data preprocessing

Here process all data from database. KDD CUP‟99 database has 41 features such as dst_bytes, src_bytes etc. Since SVM classification uses only numerical data for testing and training, so text features are needed to be converted into numerical values. Therefore, we have

assumed some numerical values for different text features, like „protocol_type" feature „tcp" as 3, „udp" as 7, and „icmp" as 9 etc. as shown in table.

Transformation Table for translating the Text data to numeric data in KDD cup'99 Data Set

TYPE	CLASS	NO.
Attack/ Normal	Attack	1
	Normal	0
Protocol Type	TCP	3
	ICMP	9
	UDP	7
Flag	OTH	1
	REJ	2
	RSTO	3
	RSTOS0	4
	RSTR	5
	S0	6
	S1	7
	S2	8
	S3	9
	SF	10
	SH	11
Services	Auth	1
	Bgp	2
	Courier	3
	csnet_ns	4
	Ctf	5
	Daytime	6
	Discard	7
	Domain	8
	domain_u	9

Step 3- Feature Selection Algorithm

In this work, Genetic algorithm based approach is proposed to select the optimal features from the overall 41 features. The selected features discriminate in predicting class during classification for anomaly and misuse.

The steps of the algorithm are as follows:

1. Generate random population of n chromosomes (dataset suitable solutions for the problem)
2. Evaluate the fitness f(x) = k (x) / sqrt(k+k(k-l) x) where k is a random number and x represents the chromosome in the population
3. Create a new population by repeating following steps until the new population is complete,

a) Select two parent chromosomes from a population according to their Fitness (the better fitness, the bigger chance to be selected).

b) With a crossover probability the parents form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

c) With a mutation probability mutate new offspring at each locus (position in chromosome).

d) Place new offspring in a new population.

4. Use new generated population for a further run of algorithm
5. If the end condition is satisfied, stop and return the best solution in current population
6. Go to step 2.

Step 4- Selected feature

The main reason for selecting KDD Cup 99 dataset is that currently, it is the mostly used comprehensive data set that is shared by many researchers. In this dataset, 41 attributes are used in each record to characterize network traffic behavior. Among this 41 attributes, 38 are numeric and 3 are symbolic. Features present in KDD data set are grouped into three categories and are discussed below.

A. Basic Features: Basic features comprises of all the attributes that are extracted from a TCP/IP connection. These features are extracted from the packet header and includes src bytes, dst_ bytes, protocol etc

B. Content Features: These features are used to evaluate the payload of the original TCP packet and looks for suspicious behavior in the payload portion. This includes features such as the number of failed login attempts, number of file creation operations etc. Moreover, most of the R2L and U2R attacks don't have any frequent sequential patterns. This is due to the fact that DoS and Probing attacks involve many connections to some host(s) in a very short duration of time but the R2L and U2R attacks are embedded in the data portions of the packets, and generally involves only a single connection. So to detect these kinds of attacks, content based features are used.

c. Traffic Features: These include features that are computed with respect to a window interval and are divided into two categories

i) "Same host" features: These features are derived only by examining the connections in the past 2 seconds that have the same destination host as the current connection, and compute statistics related to protocol behavior, service etc.

ii) "Same service" features: These features examine only the connections in the past 2 seconds that have the same service as the current connection. The above two types are called "time based traffic features".

Using the genetic algorithm, the following 9 features have been selected. It is observed that the feature selection algorithm proposed in this paper has selected only the most contributing attributes from the 41 features. These 9 features are used by the classification algorithm for effective classification of the dataset. Such as protocol types, service, src_bytes, dst_bytes, flag, diff_srv_rate, dst_host_srv_count, dst_host_error_rate, dst_host_srv_error_rate.

Step 5- Classification algorithm

We have divided the behavior of user into two classes namely attack and normal, where the behavior of user is the collection of different attacks belonging to the five classes such as

1 Normal-- Normal

2 DoS-- apache2, back, land, mailbomb, neptune , pod, processtable, smurf, teardrop, udpstrom

3 Probe -- ipsweep, mscan, nmap, portsweep, saint, satan

4 R2L-- ftp_write, guess_passwd, imap, multihop, named, phf, sendmail, spy, snmpgetattack, snmpguess , warezclient, warezmaster, worm, xlock, xsnoop

5 U2R-- buffer_overflow, httptunnel, loadmodule, perl, ps, rootkit, sqlattack, xtern

The aim of our SVM experiment is to differentiate between normal and attack behavior of user. In our experiments normal data are classified as -1 and all attacks are classified as +1.
Basic input data design and output data areas are given as follows:

(xi , yi),…, (xn ,yn), x Rm ,y {+1,-1}

where (xi , yi),…, (xn ,yn) are a train data, n is the numbers of samples, m is the inputs vector, and y fits in to category of +1 or -1 respectively. On the problem of linear, a hyper plan can divided into the two categories. The hyper plan formula is:

(w. x) + b =0

The category formulae are:
(w.x) + b ≥ if yi = +1
(w.x) + b ≤ if yi = -1

Step-6 classification result
Step 7- Anomaly detect.

5. Result Analysis

All the experiments were performed using an i3-2410M CPU @ 2.30 GHz processor and 4 GB of RAM running windows 7. discussed genetic+svm hybrid algorithms is implemented using java language in NETBEANS tool. For generation of bar chart, weka (Waikato Environment for Knowledge Analysis) data mining tool was used. Proposed as well as existing k-mean clustering algorithms were applied one by one in both the proposed framework. At last, comparative study was prepared for both frameworks.

5.1Detection Rate

Detection rate refers to the percentage of detected attack among all attack data, and is defined as follows:

$$Detection\ rate = \frac{TP * 100}{TP + FN}$$

Where,
TP (True Positive) = Number of anomalous methods
FP (False Positive) = Number of normal methods that are mistaken for the anomalous.

5.2Precession

Precision can be defined as the exactness of the approach and it can be calculated as:-

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

Where,
TP (True Positive) = Number of anomalous methods
FP (False Positive) = Number of normal methods that are mistaken for the anomalous.

5.3Recall

The measure of the completeness of the approach is called *Recall*. Recall can be calculated using given below formula:-

$$Recall = \frac{TP}{TP + FN}$$

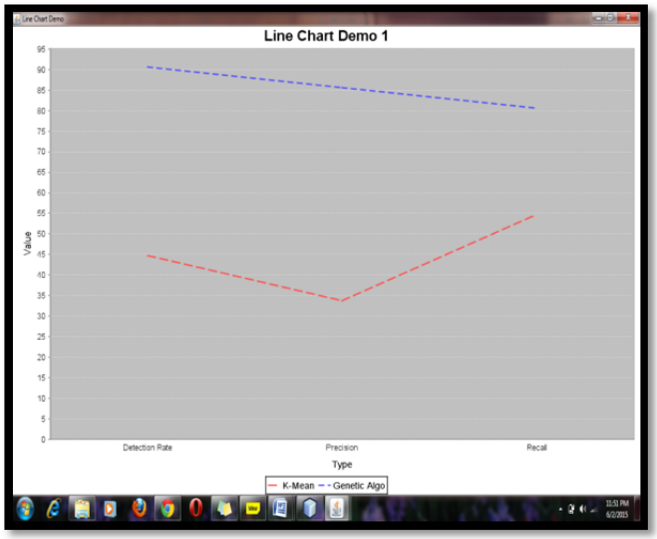
Where,
TP (True Positive) = Number of anomalous methods
FN (False Negative) = Number of anomalous methods that are mistaken for the normal.

Confusion metrics

Actual traffic label	Normal	Attack
Normal	TN	FP
Attack	FN	TP

Comparison of result with different techniques

Algorithm Name	Detection Rate	Precession	Recall
K-mean	44.75	33.75	54.75
GA+SVM	90.55	85.55	80.55



LINE CHART OF RESULT OF DIFFERENT TECHNIQUES

Here, we analyze that our algorithm is better than existing once. Because accuracy, detection rate, precession and recall result is better than existing algorithm.

6. Conclusion

The intrusion detection systems (IDS) have evolved considerably since the 1980s and are now indispensable in today's communication networks. Modern intrusion detection systems are based almost exclusively on signature detection strategies due to their remarkable successes in detection of most already known attacks. However, the appearance of new and sophisticated attacking methods is constantly and quickly making network systems again vulnerable even though they are supposedly protected by the well-established signature detection systems. Although the new approach of anomaly detection has been invented to overcome this drawback, up until now, the reported performances are still far from satisfactory, precisely because of its high false alarm rates. Definitely further research on anomaly detection is needed.

REFERENCES

1. Mohiuddin Ahmed, Abdun Naser Mahmood, "Network Traffic Analysis based on collective anomaly detection" 2014 IEEE 9th Conference on Industrial Electronics and Applications.
2. F. Ren, L. Hu, H. Liang, X. Liu, and W. Ren, "Using density-based incremental clustering for anomaly detection," in Proceedings of the 2008 International Conference on Computer Science and Software Engineering. Washington, DC, USA: IEEE Computer Society, 2008, pp. 986–989. [Online]. Available: http://dx.doi.org/10.1109/CSSE.2008.811
3. K. Burbeck and S. Nadjm-tehrani, "ADWICE - anomaly detection with real-time incremental clustering," in Proceedings of the 7th International Conference on Information Security and Cryptology, Seoul, Korea. Springer Verlag, pp. 4007-424, 2004.
4. A. Rasoulifard, A. G. Bafghi, and M. Kahani, Incremental Hybrid Intrusion Detection Using Ensemble of Weak Classifiers, in Communications in Computer and Information Science. Springer Berlin Heidelberg, November 23 2008, vol. 6, pp. 577–584. [Online]. Available: http://10.1007/978-3-540-89985-3
5. M. V. Joshi, I. T. J. Watson, and R. C. Agarwal, "Mining needles in a haystack: Classifying rare classes via two-phase rule induction," SIGMOD Record (ACM Special Interest Group on Management of Data), Vol. 30, No. 2, pp. 91-102, 2001.
6. J. Theiler and D. M. Cai, "Resampling approach for anomaly detection in multispectral images," in Proc. SPIE, pp. 230–240, 2003.
7. B.Senthilnayaki, K.Venkatalakshmi, A. Kann an, "An Intelligent Intrusion Detection System Using Genetic Based Feature Selection and Modified J48 Decision Tree Classifier" 2013 Fifth International Conference on Advanced Computing (ICoAC)
8. P. Laskov, C. Gehl, S. Kr¨uger, and K.-R. M¨uller, "Incremental support vector learning: Analysis, implementation and applications," Journal of Machine Learning Research, vol. 7, pp. 1909–1936, 2006.
9. S. Jiang, X. Song, H. Wang, J.-J. Han, and Q.-H. Li, "A clustering-based method for unsupervised intrusion detections," Pattern Recognition Letters, vol. 27, pp. 802–810, 2006.
10. H. Cheng, P.-N. Tan, C. Potter, and S. A. Klooster, "Detection and characterization of anomalies in multivariate time series," in Proceedings of the SIAM (SDM), pp. 413– 424, 2009
11. M. S. Hoque, M. A. Mukit, M. A. N. Bikas; "An Implementation of Intrusion Detection System Using Genetic Algorithm"; IJNSA; vol. 4; 2012.
12. Christopher M. K., Curtis E. Dalton, T. E. Osmanoglu, "Security Architecture: Design, Deployment and operations," RSA PRESS, Tata McGraw-Hill Edition: 2003.