Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Clustering Based Network Intrusion Detection Using Kdd Train 20 Percent

Poonam Dabas*
Assistant Professor
Department Of Computer Sc &Engg.Uiet
Kurukshetra Universty, India

Rashmi Chaudhary
M tech
Department Of Computer Sc &Engg.Uiet
Kurukshetra Universty , India

Abstract— In this Paper a clustering algorithm is proposed to work on network intrusion data. The algorithm is experimented with KDD Train 20 percent dataset and found satisfactory results. We perform clustering to group training data points into clusters, from which we select some clusters as normal and known-attack profile according to certain criterion. For those training data excluded from the profile, we use them to build a specific classifier. During the testing stage, we utilize influence based classification algorithm to classify network behaviors. In the algorithm, an influence function quantifies the influence of an object. The experiments on the KDD Train 20 percent Intrusion Detection Data Set demonstrate the detection performance and the effectiveness of our ID approach In this Dissertation, Intrusion Detection Systems detect the malicious attack which generally includes theft of information or data. It is found from the studies that clustering based intrusion detection methods may be helpful in detecting unknown attack patterns compared to traditional intrusion detection systems

Keywords—. Intrusion Detection system, Cluster, KDD Train 20 percent, False rate, True Rate, Normal, Anomaly,Condition

I. INTRODUCTION

Instruction Detection System

Informally, an Intrusion Detection system is a system for raising attention towards potential misbehaviors of the system caused by external adversaries. We could think of a 'burglar alarm' in the real world as the physical analogue of an intrusion detection system in the computerized world. (Just as a burglar alarm in the real world, Intrusion Detection only deals with discovering that an intrusion might have happened into a network. A number of additional aspects related to intrusions, such as intrusion avoidance; that is, augmenting systems so to have a lower likelihood of an external attacker that successfully performs an intrusion; or intrusion tolerance; that is, augmenting systems A network intrusion attack can be any use of a network that compromises its stability or the security of information that is stored on computers connected to it. A wide range of activity falls under this definition, including attempt to de-stabilize the network as a whole, gain unauthorized so that the intended system behavior does not change even after an intrusion; are the subject of study of different research areas.)

1.1 Architecture of IDS

Intrusion Detection Systems (IDSs) are proposed to improve computer security because it is not feasible to build completely secure systems [12]. In particular, IDSs are used to identify, assess, and report unauthorized or unapproved network activities so that appropriate actions may be taken to prevent any future damage [9]. Based on the information sources that they use, IDSs can be categorized into two classes: network-based and host-based. Network intrusion detection systems (NIDSs) analyze network packets captured from a network segment, while host- based intrusion detection systems (HIDSs) such as IDES (Intrusion Detection Expert System) [11] examine audit trails or system calls generated by individual hosts. NIDSs use software programs called sensor s to collect network packets. Because raw packets cannot be used directly for detection, many sensors have preprocessing units that transform the packets into a useful format. For example, MADAM ID [10] employs a preprocessor to transform binary tcpdump data into connections that contain context information of network sessions. As the volume of network traffic increases, many NIDSs employ multiple sensors and distributed computing to improve their processing capability. NIDSs can also detect IP-based attacks such as denial- of-service attacks which involve multiple computers. A host-based IDS has difficulty detecting these attacks since it monitors only information gathered from the computer system. NIDS is gaining popularity since more and more systems are connecting over networks.

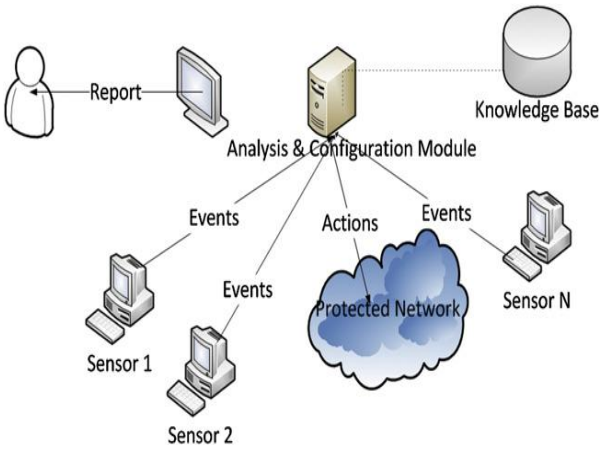

Fig 1.1 Architecture of a typical IDS

2. Related Work

lvarez, et al [1] A new clusters labelling strategy, which combines the computation of the Davies-Bouldin index of the clustering and the centroid diameters of the clusters is proposed for application in anomaly based intrusion detection systems (IDS). The aim of such a strategy is to detect compact clusters containing very similar vectors and these are highly likely to be attack vectors. Experimental results comparing the effectiveness of a multiple classifier IDS with such a labelling strategy and that of the classical cardinality labeling based IDS show that the proposed strategy behaves much better in a heavily attacked environment where massive attacks are present. The parameters of the labeling algorithm can be varied in order to adapt to the conditions in the monitored network. **Kr Singh et al [2]** Intrusion Detection Systems detect the malicious attacks which generally include theft of information or data. It is found from the studies that clustering based intrusion detection methods may be helpful in detecting unknown attack patterns compared to traditional intrusion detection systems. In this paper a new clustering algorithm is proposed to work on network intrusion data. The algorithm is experimented with KDD99 dataset and found satisfactory results. The inter cluster distance of the clusters formed is calculated by finding the Euclidean distance between the cluster means and the clusters members of all the clusters formed. A graph is plotted to study the variations of the clusters formed. A larger inter-cluster distance would mean clusters are more distinct and tight.

Lu et al [3] This paper presents an anomaly detection approach based on clustering and classification for intrusion detection (ID). We use connections obtained from raw packet data of the audit trail as basic elements, then map the network connection records into 8 feature spaces typically of high dimension according to their protocols and services. The approach includes two steps, training stage and testing stage. We perform clustering to group training data points into clusters, from which we select some clusters as normal and known-attack profile according to certain criterion. For those training data excluded from the profile, we use them to build a specific classifier. During the testing stage, we utilize influence based classification algorithm to classify network behaviors. In the algorithm, an influence function quantifies the influence of an object. The experiments on the KDD'99 Intrusion Detection Data Set demonstrate the detection performance and the effectiveness of our ID approach. This paper presents a new ID approach to detecting anomaly attacks. The ID approach maps connection records to different feature spaces in the light of protocol and service of connection and then detects anomalies by clustering and classification. We use the labeled training data to model the network behavior. Differing from ADAM which uses frequent episodes to build the normal profile, we cluster data instances that contain both normal behaviors and attacks and then select some clusters as the normal and known-attack profile according to a criterion. In order to detect novel attacks, we designed a classifier with the "default" label. We present influence-based classification algorithm to classify network behaviors.

Su et al [4] To solve the shortages of traditional k-means algorithm that it needs to input the clustering number and it is sensitive to initial clustering center, the improved k-means algorithm is put forward. In the improved algorithm, each data object will be represented by the number of points around it in a certain region. Data objects will be clustered on the basis of that the distances between data objects belonging to different kinds are farther than the ones between the same. Both k-means and improved k-means are used in intrusion detection, which shows that the improved can overcome inherent disadvantages of k means and has good clustering results. For perfecting k-means algorithm, this paper puts forward improved k-means algorithm based on k-means. It can overcome the disadvantages of k-means. By performing experiment on KDD cup99, it shows that the improved k-means has higher detection performance and is feasible.

Sun et al [5] Through analyzing the advantages and disadvantages between anomaly detection and misuse detection, a mixed intrusion detection system (IDS) model is designed. First, data is examined by the misuse detection module, then abnormal data detection is examined by anomaly detection module. In this model, the anomaly detection module is built using unsupervised clustering method, and the algorithm is an improved algorithm of K-means clustering algorithm and it is proved to have high detection rate in the anomaly detection module. Through analysis the merit and shortcoming between traditional anomaly detection and misuse detection technique, discover that these two kinds of techniques can repair with each other, a intrusion detection model is designed mixing these two techniques. In order to improve the validity of detection model, unsupervised clustering method is used in abnormal detection module, make the whole

detection model can identify intrusion behavior already known more accurately, and can discover unknown intrusion behavior.

3.Purposed Algorthim

3.1 Clustring K-mean Algorithm Intrusion Detection Algorthim

K-means algorithm is a classical clustering algorithm. Its aim is to divide data into k clusters, and ensures that the data within same cluster has high similarity; the data in different cluster has low similarity.

K-means algorithm first select K data at random as initial cluster center, for the rest data, add it to the cluster with highest similarity according to its distance to cluster center; then recalculate the cluster center of each cluster. Repeat this process until each cluster center doesn't change. Thus data is divided into K clusters. K-means algorithm is very simple, it is suitable for large scale data set. But K-means algorithm is sensitive to initial value and sequence of data object, different initial data may lead to different clustering result. The purpose of the proposed approach is to perform a clustering analysis on a set of tested connections through K means, and then compute the distribution of false alerts in these clusters. This operation is repeated several times, with a different number of clusters each time, until obtaining a final configuration of clusters where each cluster is ideally highly representative of false alerts (the percentage of false alerts is high), or it is highly representative of real attacks (the percentage of false alerts is low).

The following algorithm, which is called Clustering K-mean algorithm here, is an improved algorithm to K- means In KD algorithm, suppose that a constant R stands for clustering radius threshold, C stands for cluster, dist(C,D) stands for the distance from the center of clustering C to vector D, KD clustering algorithm is following:

Step 1: initialize clustering set S to null set;
Step 2: fetch a vector d from data set;
Step 3: if S is null, build a new cluster centered on d, and add it to S. Go to Step 7;
Step 4: else find a cluster C j from S, which is the closest to d among all created clusters, that is dist(C j, d) is the smallest;
Step 5: if dist(C j, d) ≤R, add d to C j. Go to Step 7;
Step 6: else, build a new cluster centered on d, add it to S;
Step 7: repeat (2) (3) until all vectors of data set are processed.
Step 8: recalculate the center of each cluster, for each cluster scanning data set from beginning, if the distance from certain vector of data set to cluster center isn't larger than R, add this vector to this cluster. Repeat it until all cluster center not changed.

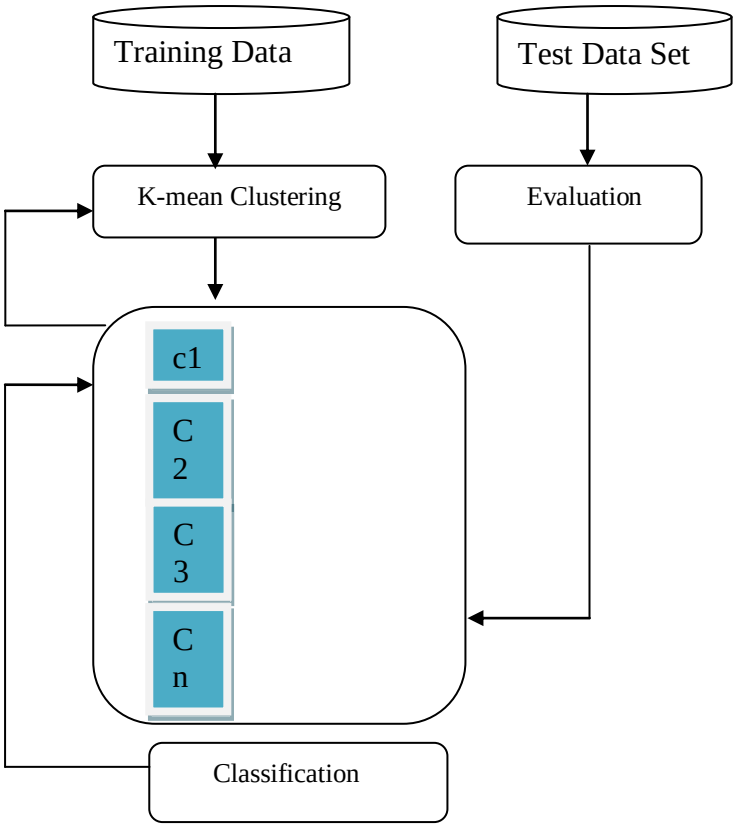

Fig3.1: Main Step of the Proposed Approach

4. Implementation and Result Analysis

We Implemented for k-mean clustering based intrusion detection technique are given. First the dataset used for the evaluation is described and then a large training data set for intrusion detection is presented. Finally the experimental results are presented and discussions about the performance of the detection method are given

4.1 Analysis Data Set

We have performed reduction of dimensionality of the KDD Train 20 percent data set. It is an important step, not only to reduce the complexity of the training process but also to gain an insight as to which network connection features are significant for the process of any network intrusion detection. Having done that, these features are real numbers on different scales collected. TCPDUMP files are converted to arff and csv format file.

4.2 Anomaly Detection

In order to ensure that all possible normal program behaviors are included, a large training data set is preferred for anomaly detection. K-mean algorithm is applied on test data using Weka 3.6.2 and centroid are calculated

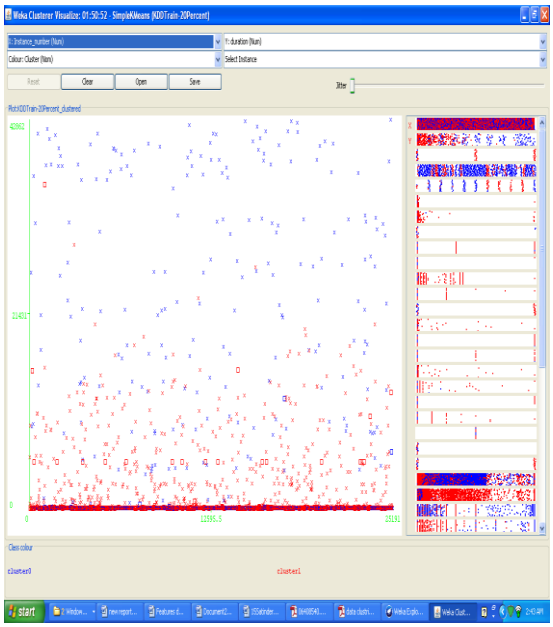

Fig 4.1: Cluster Visualization

Matching Matrix

Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class. One benefit of a matching matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

Table 5.3 Matching Matrix for k=2

O	1
Normal	Anomaly
65	13384
9490	2253

Standard measures which were developed for evaluating IDSs include detection rate (DTR), false positive rate (FPR), and overall accuracy (OA). These three performance metrics may be defined as follows [21]:

$$DTR = \frac{TP}{TP+FN} \times 100\%$$

$$FPR = \frac{FP}{TN+FP} \times 100\%$$

$$OA = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

where TP, TN, FP, and FN are the numbers of malicious executables correctly classified as Malicious, benign programs correctly classified as benign, benign programs falsely classified as malicious, and malicious executables falsely classified as benign, respectively. An IDS requires high DTR, low FPR, and high OA.

Accuracy:

The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation.

acc = (a+d)/(a+b+c+d)

= (65+2253)/(65+13384+9490+2253)=9.2013%

Probability of False Alarm :

It is the proportion of negatives cases that were incorrectly classified as positive.

pf = c/(a+c) = 9490/(65+9490)=0.3%

A set of normal and abnormal processes was taken for experiments. This set was fed into k-mean clustering. Using K-mean clustering algorithm centroid is calculated for anomaly and normal data. The value of k was varied from 2 to 6 The clustering algorithm performed best for k=2. The detection rate reached 96% rapidly for a smaller threshold value and the false positive rate

5 Conclusion

A new clustering algorithm is proposed which is based on K-means. The efficiency of the algorithm with which intrusions are detected is around 90%-95%. The accuracy of this algorithm depends on the training data taken. If the number of instances of a particular type is equal to that of the other then there is an algorithm based on the k-mean clustering for analyzing program behavior in intrusion detection is evaluated by experiments. The preliminary Experiments with the KDD Train 20 percent audit data have shown that this approach is able to effectively detect intrusive program behavior. Compared to other methods using normal and anomaly based tcp attributes, the k-mean clustering centroid calculated. The results also show that a low false positive rate can be achieved. we statistically analyzed the entire KDD data set. The analysis showed that there are two important issues in the data set which highly affects the performance of evaluated systems, no duplicate records in the proposed test sets; therefore, the performance of the learners are not biased by the methods which have better detection rates on the frequent records. The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

6 Future Scope

In the near feature, we will conduct more experiments. We'd like to test the time based features on different sizes as on KDD Train 20 Percent dataset. We will also study the impact of different normalization methods and the impact of weighted features. To improve the usability of the IDS, the future work can be done as follows : more checking rules can be developed and implemented for the KDD Train set to improve its ability to detect compromised components; host-based IDSs can be introduced to the IDS and used to monitor the hosts and the IDS components ; an intelligent system can be employed to analyze the intrusion alerts generated by the k-mean based IDS and aid the intrusion-tolerant mechanism in treating the compromised components; and finally, a response mechanism can be introduced in order to stop intrusions before a failure occurs. We believe it still can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods. For the future, we envision three major thrusts. First, we will continue to expand the archive as rapidly as possible. The enthusiasm for large databases in KDD has spread over to other Fields

REFERENCES

[1]Slobodan Petrovic´, Gonzalo A´ lvarez, Agust´ın Orfila, and Javier Carbo" Labelling Clusters in an Intrusion Detection System Using a Combination of Clustering Evaluation Techniques" Proceedings of the 39th Hawaii International Conference on System Sciences – 2006

[2] Samarjeet Borah, Saugat P.K. Chetry, and Pramod Kr Singh "Hashed-K-Means: A Proposed Intrusion Detection Algorithm" Springer-Verlag Berlin Heidelberg 2011.

[3] Hongyu Yang, Feng Xie, and Yi Lu "Clustering and Classification Based Anomaly Detection" Springer-Verlag Berlin Heidelberg 2006.

[4] Mingjun Wei, Lichun Xia, and Jingjing Su "Research on the Application of Improved K-Means in Intrusion Detection" ICICA 2011, Part I, CCIS 243, pp. 673–678, 2011.

[5] Cuixiao Zhang; Guobing Zhang; Shanshan Sun "A Mixed Unsupervised Clustering-based Intrusion Detection Model" 2009 Third International Conference on Genetic and Evolutionary Computing.

[6] Sanoop Mallissery 1, Jeevan Prabhu 2, Raghavendra Ganiga" SURVEY ON INTRUSION ETECTION METHODS" Proc. of1nt. Con/, on Advances in Recent Technologies in Communication and Computing 2011.

[7] Z. Muda, W. Yassin , M.N. Sulaiman "Intrusion Detection based on K-Means Clustering and OneR Classification 2011 IEEE.

[8] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani "A Detailed Analysis of the KDD CUP 99 Data Set" Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[9] Meng Jianliang Shang Haikun Bian Ling "The Application on Intrusion Detection Based on K-means Cluster Algorithm" 2009 International Forum on Information Technology and Applications.

[10] Tayeb Kenaza, Abdelhalim Zaidi" Clustering approach for false alerts reducing in behavioral based intrusion detection systems 2010 IEEE.

[11] Pavel Laskov, Patrick D¨ussel, Christin Sch¨afer, and Konrad Rieck "Learning Intrusion Detection: Supervised or Unsupervised? Springer-Verlag Berlin Heidelberg 2005.

[12] Giovanni Di Crescenzo, Abhrajit Ghosh, and Rajesh Talpade "Towards a Theory of Intrusion Detection_"Springer 2005

[13] Aurobindo Sundaram. "An introduction to intrusion detection". Crossroads, 2(4):3–7, 1996.

[14] V. Rao Vemuri, "Text Processing Techniques With a Binary-Weighted Cosine Metric, " university of California Davis, CA 95616, USA rvemuri@ucdavis.edu.

[15] S. R. Snapp, J. Brentano, G. V. Dias, T.L. Goan. "DIDS (distributed intrusion detection system) - motivation, architecture,and an early prototype." In Proceedings of the 14th National Computer Security Conference, pages 167–176, Washington, DC, October 1991.

[16] Koral Ilgun, Richard A. K emmerer, and Phillip A. Porras. State transition analysis: A rule- based intrusion detection approach. IEEE Transactions on Software Engineering, 21(3):181–199, 1995.

