VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

INTRUSION DETECTION MODEL USING INTEGRATED CLUSTERING AND DECISION TREES

Ranita Pal and Sumaiya Thaseen
School of Computing Science and Engineering, VIT University, Chennai, India

ABSTRACT

This paper proposes a hybrid technique for intrusion detection model using K-means clustering, attribute selection and decision tree. K-means clustering is a very simple and convenient clustering method when it comes to grouping anomalies and the different attack types in network traffic. An enhanced mechanism is developed using the Cluster center initialization algorithm for k-means clustering and decision trees using the entropy method. After the clustering is done, attribute subset selection is done using entropy method and final classification of attack categories is done using decision trees. It works in two modes: online and offline. Offline mode works on the sample data which is processed to obtain the rule set of the decision tree. The data from the online mode is then compared against those rules to determine their category and identify the intrusions in the packet.

Keywords: anomalies, cluster center initialization, decision trees, information gain, K-means clustering.

1. INTRODUCTION

Intrusions [7] are attacks or malicious activities that challenge the security of a system. An intrusion detection system tries to analyze data fed to it in order to determine whether it is a normal data or an anomalous data. Anomaly detection is to find patterns in the data that differs from normal behavior.

Nomenclature

IDS Intrusion detection system
SD Standard deviation

Anomaly detection [4] is often performed both with labeled data and unlabeled data. The three types of anomaly detection are:

1. **Supervised anomaly detection:** The labeled training data set is available and is checked with the real time data to find out whether it is normal or not.
2. **Semi-supervised anomaly detection:** Only for the normal class labeled data is obtained. Anything other than this pattern is as anomalous.
3. **Unsupervised anomaly detection:** Training dataset is not used here. It is assumed that normal data is far more frequent than anomalous data.

Supervised anomaly detection technique is used here where the dataset identifies normal class of data and the attack types. Clustering is used to group these classes. Clustering is used for grouping similar kinds of data types into one class. Normal data instances usually belong to large and dense clusters and the various attack types group together to form smaller clusters as compared to the normal cluster. The data that lie far away from the normal region is taken to be anomalous data.

Clustering techniques [5] are divided into several categories- grid based clustering, partitioning clustering, hierarchical clustering, and density based clustering. K-means clustering is the most popular clustering technique.

K-means clustering technique has been used classification of data. The traditional k-means clustering has certain limitations [2]:

The number of clusters to be formed is predetermined.

The initial seeds effect the clusters formed and the initial seeds are usually taken randomly.

Therefore, an enhanced clustering technique for the offline mode is performed and the initial seeds so obtained are used in the online phase to categorize the real time data.

The traditional k-means clustering calculates the Euclidean distances of the individual data sets from the initial seeds. The data set is then said to belong to the cluster whose seed is nearest to it. Data which lies within a certain diameter around cluster centers belong to this class and the ones outside it either fall into some other class or are treated as outliers. K-means technique is used twice: once for online phase and a modified one for the offline phase. In the offline phase the cluster seeds are initialized randomly causing the results to vary according to it.

After the data is clustered, we determine the attack category the data packet falls into. The entropy method is used to determine in a stepwise manner which sub-category of attack it is by identifying the relevant attribute for that category. After clustering the online data it is compared to the decision tree formed in the entropy method to determine its category. In this way a method is devised for classifying network data and assigning them their proper class.

2. RELATED WORK

K-means clustering forms the very integral part of our study. Clustering of real time data is done with the k-means clustering algorithm. It takes n objects and a clustering number K as input and gives K clusters including those n objects in total as output [1]. Random initializations of the cluster seeds are done. The Euclidean distance is calculated for each object from each of the cluster seeds and then assigned to the nearest cluster seed.

The cluster seeds are computed again and several iterations of this process are carried on the basis of the error rule function E.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - m_i)^2 \quad (1)$$

The improved K-means algorithm performs better in global searching and is not much dependent on the initial centroid taken. If m_i is the cluster centroid of the i^{th} cluster P is represented as a spatial vector. P(t) denotes the clustering centroid in the t^{th} iteration. The error function of this cluster centroid is calculated to find best solution. Then the cluster centroid is adjusted.

The features of a data packet are a major factor in the efficiency of the intrusion detection system. Li *et al*. [13] proposed gradual feature removal along with clustering, ant colony algorithm and support vector machines (SVM) to produce an efficient classifier of data. Many clustering based intrusion detection techniques have been developed with unsupervised anomaly detection [7] to detect many different types of attack.

Another variety of k-means clustering was introduced where the number of clusters was not predetermined [9]. The minimum numbers of clusters are obtained by minimizing a cost function. In the first stage clustering is done and one seed point is assigned to each cluster. In the second stage, the seed points are used to minimize the cost function. Anomaly detection is preferred in many ways over misuse detection [11]. Many types of clustering algorithms were studied and the accuracy for anomaly detection was found to be much greater than misuse detection. It is seen that on modifying k-means by using normalization and preprocessing steps [14] then effectiveness is improved. This method works for network intrusion data.

In a study of intrusion detection system, a new metric was introduced which stated the representative power of an instance in that class [12]. The most representative element is chosen from the subset and then this is utilized to train the data set and to build various intrusion detection models. Another study has achieved better accuracy and detection rate when it combined clustering with classification techniques [10].

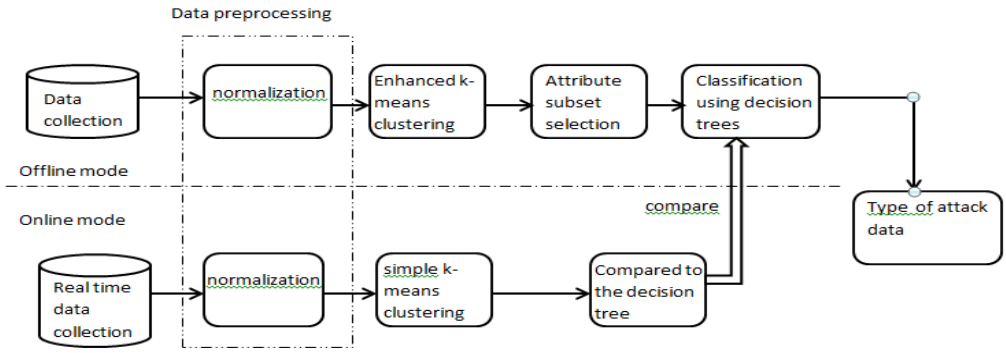

Figure-1. Proposed intrusion detection model.

3. BACKGROUND

3.1 Clustering

Clustering techniques [5] are being used here to form clusters of different classes of data. Since the separate classes need to be formed we use the k-means clustering. The simple k-means clustering is formed using the Euclidean data metric:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where d(x, y) is the distance between two points x and y each having n attributes. The algorithm works in this way:

1) Initially cluster seeds are taken from a few of the data tuples randomly which are considered to be cluster centers.
2) The distance from each instance of data to each of these cluster centers are calculated.
3) The point is assigned to the cluster whose center is the nearest.
4) After all the points are assigned the cluster centers are calculated again from the points assigned to that clusters in that iteration
5) Several iterations are made till no change in assignment of clusters is noticed.

Cluster center initialization algorithm [2] is used in intrusion detection to determine the cluster centers.

It is assumed that each of the attribute is normally distributed in pattern space.

The cluster center initialization algorithm works as:

1. A parameter K is fixed and the normal curve is divided into K regions.

2. The percentile is calculated by the area under the normal curve from -∞ to $\frac{x_s-\mu}{s.d.}$, where s=1,2…K.
3. The mean and s.d. of each attribute is calculated separately. Then $x_s=z_s*s.d.+\mu$ is calculated. These serve as the initial cluster centers for the k-means clustering. This is repeated for all the attributes. Now it is determined for each data point which attributes falls under which cluster.
4. A pattern is generated by collecting the classes of each attribute. After scanning the patterns generated by each of the data point, the unique patterns are retrieved. These serve as the new clusters.
5. The centers are determined and then the cluster merging is required to be done. Merging is done by deciding upon a constant, say q, and the center which has the nearest q^{th} neighbor is selected.
6. All the centers around a specific range of this center form a cluster. This process is repeated till all the centers belong to some cluster.

Hence the cluster center is determined.

3.2 Subset selection method

The attributes of the different classes are dissimilar. But within a class attributes are similar. So in order to categorize a data within a class a subset of attributes should be taken [5]. It is determined in the offline mode. This is done by the entropy method. The information gain

$$I = \sum_i (-p_i \log p_i)$$

where p_i is the probability of each class.

Using the same formula the information in each of the attributes is calculated separately. In each attribute information for values less than and more than the average value is calculated taking the attack labels into consideration and then summed up proportionately. The attribute with the highest information gain is to be eliminated first. Then the procedure continues with the rest of the attributes.

3.3 Decision trees

The attribute which gets eliminated first in the previous step, forms the root of the tree [5]. The tree is constructed with sample data from the offline mode. Depending on the values of the data set the different branches are formed from. Then the second attribute which gets eliminated forms the next level of the tree. In this way a decision is taken that a data lies in which category. Moving from the root to leaf, following the correct branching allows the online data to recognize its category.

4. PROPOSED MODEL

In the proposed approach, cluster center initialization and decision trees are used in intrusion detection model. There are two modes of operation: offline mode and online mode.

A. In the offline mode, labeled data is collected from the network traffic and samples are taken from it in order to form a model. To do so, first the data is normalized and then cluster center initialization algorithm is applied in order to get the initial clusters and its centers. This is obtained without fixing the number of clusters as cluster center initialization algorithm is followed. Then attribute subset selection method is used to select the most determining attribute of that cluster. The method used here is entropy method. This operation is performed separately in each cluster. A decision tree is formed after previous step. The first eliminated attribute in subset selection forms the root of the tree. Following this manner we form the next levels of the tree.
B. In the online mode of operation, real time network packets are captured. The entire data set is normalized prior to any operation. With this cluster centers formed in the offline mode, the real time data is clustered by simple k-means method. The data is now grouped into various classes. It is now compared with the decision tree of that class and after comparing the features of its attributes with the tree its attack type is determined.

5. RESULTS

For the offline mode, samples were taken from the KDD dataset. Different numbers of packets were taken. Among them in medium sampling the best type of clusters were obtained. Around seven thousand packets were collected which included all the attack types. The number of samples is 7099 out of which 4000 are normal data and the rest attack data. After the first stage of cluster center initialization algorithm, 258 unique patterns of data were obtained, on merging which a minimum of 14 clusters are obtained. Hence 14 cluster centers were also obtained each having 38 numeric attributes. These centers can be used to determine in which cluster each sample from the network traffic falls under in the online mode. The 14 clusters contain the following attacks:

Table-1. Clusters of different attack categories.

cluster	Attack types
Cluster 1	Teardrop
Cluster 2	Teardrop, neptune, pod,satan, imap, nmap, portsweep, warezclient, multihop, perl, roootkit
Cluster 3	Normal, back, nmap, warezclient
Cluster 4	Normal, back, nmap, warezclient
Cluster 5	Normal, back, phf, smurf
Cluster 6	Normal
Cluster 7	Warezclient, nmap, imap, spy, portsweep, Neptune
Cluster 8	Normal, ftp_write, land, multihop, perl, teardrop, back, imap, loadmodule, nmap, portsweep, warezclient, buffer_overflow, ipsweep, neptune, pod, rootkit
Cluster 9	Pod, nmap, ipsweep, normal
Cluster 10	Teardrop
Cluster 11	Normal
Cluster 12	Teardrop, pod, normal, satan, nmap, portsweep, neptune
Cluster 13	Smurf, warezclient, normal
Cluster 14	Buffer_overflow, loadmodule, multihop

Using the information gain at each step we arrive at the decision trees. The trees are separate for each cluster. After determining the particular cluster of the network traffic data using simple k-means algorithm, the attack can be determined by comparing its attribute values with the decision tree for that cluster. The attribute values have to be matched with the rule set of the decision tree.

Table-2. Rule set for cluster 2.

Dst_host_error_rate=0, dst_host_diff_srv_count=0, same_srv_rate=1, wrong_fragment=333	teardrop
Dst_host_error_rate=1, same_srv_rate=0 or Dst_host_error_rate=0, dst_host_diff_srv_count=0, & same_srv_rate=0, diff_srv_rate=0 or same_srv_rate=1, wrong_fragment=0, logged_in=1, hot=0, number_shells=0, num_compromised=421	imap
Dst_host_error_rate=0, dst_host_diff_srv_count=0, same_srv_rate=1, wrong_fragment=0, logged_in=1, & hot=33, duration=12 or hot=6,7	multihop
Dst_host_error_rate=1, same_srv_rate=0	neptune
Dst_host_error_rate=1, same_srv_rate=1, wrong_fragment=0	nmap
Dst_host_error_rate=0, dst_host_diff_srv_count=0, same_srv_rate=1, wrong_fragment=0, logged_in=1, hot=0, number_shells=500	perl
Dst_host_error_rate=1, same_srv_rate=1, wrong_fragment=333	pod
Dst_host_error_rate=0, dst_host_diff_srv_count=0, & same_srv_rate=0, diff_srv_rate=1, dst_host_diff_srv_count=0, dst_host_error_rate=1 or same_srv_rate=1, wrong_fragment=0, logged_in=0, rerror_rate=1 or Dst_host_error_rate=0, dst_host_diff_srv_count=1	portsweep
Dst_host_error_rate=0, dst_host_diff_srv_count=0, same_srv_rate=1, wrong_fragment=0, & logged_in=0, rerror_rate=0, duration=0 or 1 or logged_in=1, hot=33, duration=1,6 or logged_in=1, hot=0, number_shells=0, num_compromised=0,158	rootkit
Dst_host_error_rate=0, dst_host_diff_srv_count=0, , same_srv_rate=0, diff_srv_rate=1, dst_host_diff_srv_count=1 or dst_host_diff_srv_count=0, dst_host_error_rate=0 or same_srv_rate=1, wrong_fragment=0, logged_in=0, rerror_rate=0, duration=0	satan
Dst_host_error_rate=0, dst_host_diff_srv_count=0, same_srv_rate=1, wrong_fragment=0, logged_in=1, hot>=100	warezclient

Table-3. Rule set for cluster 3.

Flag=7,8	back
Flag=2	warezclient
Flag=0, logged_in=0	nmap

Table-4. Rule set for cluster 4.

Num_compromised=26 or Num_compromised=0, hot=33	back
Num_compromised=0, hot=933	warezclient
Num_compromised=0, hot=0, logged_in=0	nmap

Table-5. Rule set for cluster 5.

Hot=33 or Hot=67, num_compromised=26	back
Hot=67, num_compromised=0	phf
Hot=0, dst_host_same_src_port_rate=1	smurf

Table-6. Rule set for cluster 7.

Count=0 or >=16 & <=160 or Count=2, logged_in=0, serror_rate=0, duration=0	portsweep
Count=4,6 or Count=2, logged_in=1	warezclient
Count>=120 & <=210	neptune
Count>=400 & <=600	imap
Count>800	satan
Count=2, logged_in=0, serror_rate=1	nmap
Count=2, logged_in=0, serror_rate=0, duration=20	spy

Table-7. Rule set for cluster 8.

Logged_in=1, & dst_host_srv_diff_host_rate=1, root_shell=0 or dst_host_srv_diff_host_rate=0, dst_host_same_srv_rate=1, root_shell=0, num_compromised=26 or num_compromised=0, num_file_creations=0, dst_host_diff_srv_count=0	back
Logged_in=1, dst_host_srv_diff_host_rate=0, dst_host_same_srv_rate=1, root_shell=1 or root_shell=0, num_compromised=0, num_file_creations=250, flag=2,3 or Logged_in=1, dst_host_srv_diff_host_rate=1, root_shell=0 or root_shell=1, diff_srv_rate=0	Buffer_overflow
Logged_in=1, dst_host_srv_diff_host_rate=1, root_shell=0 or Logged_in=0, dst_host_srv_diff_host_rate=1, wrong_fragment=0, land=0	ftp_write
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=0, srv_error_rate=1, count=2, land=0 or srv_error_rate=0, dst_host_diff_srv_count=0, count=2, duration=3	imap
Logged_in=1, dst_host_srv_diff_host_rate=0, dst_host_same_srv_rate=0, dst_host_diff_srv_count=1 or Logged_in=0, & dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=1 Or dst_host_same_src_port_rate=0, srv_error_rate=0, dst_host_diff_srv_count=1, or dst_host_srv_diff_host_rate=1, wrong_fragment=0, land=0	ipsweep
Logged_in=0, dst_host_srv_diff_host_rate=0, & dst_host_same_src_port_rate=0, srv_error_rate=1, count=6 or count=2, land=1,or dst_host_same_src_port_rate=1 or Logged_in=0, dst_host_srv_diff_host_rate=1, wrong_fragment=0, land=1	land
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=1 or, Logged_in=1, dst_host_srv_diff_host_rate=1, root_shell=0 or root_shell=1, diff_srv_rate=1 or, Logged_in=1, dst_host_srv_diff_host_rate=0, dst_host_same_srv_rate=1, root_shell=0, num_compromised=0,& num_file_creations=0, dst_host_diff_srv_count=1 or num_file_creations=250, flag=1	loadmodule
Logged_in=0, dst_host_srv_diff_host_rate=0,& dst_host_same_src_port_rate=0, srv_error_rate=1, count>30 or dst_host_same_src_port_rate=1	neptune
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=1	multihop
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=0, srv_error_rate=0, dst_host_diff_srv_count=0, count=4,6	nmap
Logged_in=0, & dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=1 Or dst_host_srv_diff_host_rate=1, wrong_fragment=333	pod
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=0, srv_error_rate=0, dst_host_diff_srv_count=0, count=2, duration=0, wrong_fragment=0, rerror_rate=1	portsweep
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=1 or Logged_in=1, dst_host_srv_diff_host_rate=1, root_shell=0	rootkit
Logged_in=0, dst_host_srv_diff_host_rate=0, dst_host_same_src_port_rate=0, srv_error_rate=0, dst_host_diff_srv_count=0, count=2, duration=0, wrong_fragment=333	teardrop

Table-8. Rule set for cluster 9.

Logged_in=0, dst_host_srv_diff_host_rate=1	ipsweep
Logged_in=0, dst_host_srv_diff_host_rate=0, wrong_fragment=0	nmap
Logged_in=0, dst_host_srv_diff_host_rate=0, wrong_fragment=333	pod

Table-9. Rule set for cluster 12.

Wrong_fragment=0, same_srv_rate=0, count=12, 14	satan
Wrong_fragment=0, same_srv_rate=0, count>140	neptune
Wrong_fragment=0, same_srv_rate=1, rerror_rate=1	portsweep
Wrong_fragment=0, same_srv_rate=1, rerror_rate=0, logged_in=0, srv_diff_host_rate=0	nmap
Wrong_fragment=333, dst_host_count>=330 & <=410	pod
Wrong_fragment=333, dst_host_count=799	teardrop

Table-10. Rule set for cluster 13.

Logged_in=0	smurf
Logged_in=1, dst_host_count=1000	warezclient

Table-11. Rule set for cluster 14.

Num_compromised=26 or Num_compromised=0, dst_host_count=12 or Num_compromised=105, duration=7	loadmodule
Num_compromised=100,579	multihop
Num_compromised=0, dst_host_count=0 or Num_compromised=105, duration=19	Buffer_overflow

6. CONCLUSION AND FUTURE SCOPE

In this approach, clustering algorithm is used to group similar data and then classify them using entropy method and decision trees. The algorithm used in this approach does not take random initialization which would have brought variation to the result. It provides a minimized number of clusters by itself which are further used for classification. The different attributes which determine the attack are obtained through the entropy method. The rule set of the decision trees can be used to identify the attack type. Hence any type of network traffic can be given as input and it can be determined whether it is normal or an attack. As an extension to this model, various feature selection techniques can be analyzed along with the proposed method. This can determine which technique is the most efficient and suitable for the purpose of detecting attack traffic.

REFERENCES

[1] Zhe Zhang, Junxi Zhang, Huifeng Xue, Improved K-means Clustering Algorithm. 2008 Congress on Image and Signal Processing.

[2] Khan, S., Ahmad, A., 2004. Cluster centre initialization algorithm for k-means clustering. Pattern Recognition Lett. 25, Issue 11, August.

[3] Jiawei, Han. M. Kamber. 2001. Data Mining: Concepts and Techniques. Los Altos, CA: Morgan Kaufmann Publishers.

[4] Chandola, V., Banerjee, A., and Kumar, V. 2009. Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15.

[5] G.K. Gupta. 2009. Introduction to Data Mining with Case Studies, PHI Learning Private Limited.

[6] Ertoz, l., Eilertson, e., Lazarevic, a., Tan, p.-N., Kumar, v., Srivastava, j., and dokas, P. 2004. MINDS-Minnesota Intrusion Detection System. In Data Mining-Next Generation Challenges and Future Directions. MIT Press.

[7] Eleazar Eskin, Andrew Arnold, Michael Prerar,Leonid Portnoy, Sal Stolfo. 2002. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data.

[8] Li Tian. 2009. "Research on Network Intrusion Detection System Based on Improved K-means Clustering Algorithm", Computer Science-Technology and Applications. IFCSTA '09. International Forum.

[9] Rizman Žalik, Krista. 2008. An efficient k'-means clustering algorithm. Pattern recogn. lett. (Print). [Print ed.], July, vol. 29.

[10] Muda, Z., Yassin, W., Sulaiman, M.N., Udzir, N.I. 2011. Intrusion detection based on k-means clustering and OneR classification, Information Assurance and Security (IAS), 7th International Conference.

[11] Iwan Syarif, Adam Prugel-Bennett, Gary Wills. 2012. Unsupervised clustering approach for network anomaly detection. 4th International Conference, NDT, Dubai, UAE, April 24-26. Proceedings.

[12] Chun Guo, Ya-Jian Zhou, Yuan Ping, Shou-Shan Luo, Yu-Ping Lai, Zhong-Kun Zhang, Efficient intrusion detection using representative instances, In Press.

[13] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai. 2012. "An efficient intrusion detection system based on support vector machines and gradually feature removal method"; presented at Expert Syst. Appl., pp. 424-430.

[14] M. Varaprasad Rao, A. Damodaram, N. Ch. Bhatra Charyulu. 2012. Algorithm for Clustering with Intrusion Detection Using Modified and Hashed K - Means Algorithms; Proceedings of the Second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25-27, New Delhi, India. Vol. 2.