Sonika Tiwari, Prof. Roopali Soni / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1495-1500

Performance Comparison Of Different Clustering Algorithms With ID3 Decision Tree Learning Method For Network Anomaly Detection

Sonika Tiwari*, Prof. Roopali Soni**
*(Department of Computer Science, OIST Bhopal)
** (Department of Computer Science, OIST Bhopal)

ABSTRACT

This paper proposes a combinatorial method based on different clustering algorithms with ID3 decision tree classification for the classification of network anomaly detection. The idea is to detect the network anomalies by first applying any clustering algorithm to partition it into a number of clusters and then applying ID3 algorithm for the decision that whether an anomaly has been detected or not. An ID3 decision tree is constructed on each cluster. A special algorithm is used to combine results of the two algorithms and obtain final anomaly score values. The threshold rule is applied for making decision on the test instance normality or abnormality. Here we are comparing the result performance of the best clustering algorithm for the detection of the network anomalies. The algorithms that we shall apply here are k-mean algorithm, hierarchical clustering, expected maximization clustering. All these algorithms are first applied on the data sets consisting of a captured network ARP traffic to group them into a number of clusters and then by applying ID3 decision tree classification on each of the clustering algorithm for the detection of the network anomalies and compare the performance of each clustering algorithm.

I. INTRODUCTION

It is important for companies to keep their computer systems secure because their economical activities rely on it. Despite the existence of attack prevention mechanisms such as firewalls, most company computer networks are still the victim of attacks. According to the statistics of CERT [1], the number of reported incidents against computer networks has increased from 252 in 1990 to 21756 in 2000 and to 137529 in 2003. This happened because of misconfiguration of firewalls or because malicious activities are generally cleverly designed to circumvent the firewall policies. It is therefore crucial to have another line of defence in order to detect and stop malicious activities. This line of defence is intrusion detection systems (IDS).

During the last decades, different approaches to intrusion detection have been explored. The two most common approaches are misuse detection and anomaly detection. In misuse detection, attacks are detected by matching the current traffic pattern with the signature of known attacks. Anomaly detection keeps a profile of normal system behavior and interprets any significant deviation from this normal profile as malicious activity. One of the strengths of anomaly detection is the ability to detect new attacks. Anomaly detection's most serious weakness is that it generates too many false alarms. Anomaly detection falls into two categories: supervised anomaly detection and unsupervised anomaly detection. In supervised anomaly detection, the instances of the data set used for training the system are labelled either as normal or as specific attack type. The problem with this approach is that labeling the data is time consuming. Unsupervised anomaly detection, on the other hand, operates on unlabeled data. The advantage of using unlabeled data is that the unlabeled data is easy and inexpensive to obtain. The main challenge in performing unsupervised anomaly detection is distinguishing the normal data patterns from attack data patterns.

Recently, clustering has been investigated as one approach to solving this problem. As attack data patterns are assumed to differ from normal data patterns, clustering can be used to distinguish attack data patterns from normal data patterns. Clustering network traffic data is difficult because:

1. of high data volume
2. of high data dimension
3. the distribution of attack and normal classes is skewed
4. the data is a mixture of categorical and continuous data
5. of the pre-processing of the data required.

Network anomaly detection

As we explained earlier, detectors need models or rules for detecting intrusions. These models can be built off-line on the basis of earlier network traffic data gathered by agents. Once the model has been built, the task of detecting and stopping intrusions can be performed online. One of the weaknesses of this approach is that it is not adaptive. This is because small changes in traffic affect the model globally. Some approaches to anomaly detection perform the model construction and anomaly detection simultaneously on-line. In some of these approaches clustering has been used. One of the advantages of online modelling is that it is less time consuming because it does not require a

separate training phase. Furthermore, the model reflects the current nature of network traffic. The problem with this approach is that it can lead to inaccurate models. This happens because this approach fails to detect attacks performed systematically over a long period of time. These types of attacks can only be detected by analysing network traffic gathered over a long period of time. The clusters obtained by clustering network traffic data off-line can be used for either anomaly detection or misuse detection. For anomaly detection, it is the clusters formed by the normal data that are relevant for model construction. For misuse detection, it is the different attack clusters that are used for model construction.

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar amongst them and dissimilar compared to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence, it models data by its clusters.

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. It is important to understand the difference between clustering (unsupervised classification) and discriminate analysis (supervised classification). In supervised classification, we are provided with a collection of labelled (preclassified) patterns; the problem is to label a newly encountered, yet unlabeled, pattern. Typically, the given labeled (training) patterns are used to learn the descriptions of classes which in turn are used to label a new pattern. In the case of clustering, the problem is to group a given collection of unlabeled patterns into meaningful clusters. In a sense, labels are associated with clusters also, but these category labels are data driven; that is, they are obtained solely from the data [2,3,4].

ID3 Algorithm

The ID3 algorithm (Inducing Decision Trees) was originally introduced by Quinlan in [11] and is described below in Algorithm 1. Here we briefly recall the steps involved in the algorithm. For a thorough discussion of the algorithm we refer the interested reader to [10].

Require: R, a set of attributes.
Require: C, the class attribute.
Require: S, data set of tuples.
1: if R is empty then
2: Return the leaf having the most frequent value in data set S.
3: else if all tuples in S have the same class value then
4: Return a leaf with that specific class value.
5: else
6: Determine attribute A with the highest information gain in S.
7: Partition S in m parts S(a1), ..., S(am) such that a1, ..., am are the different values of A.
8: Return a tree with root A and m branches labeled a1...am, such that branch i contains ID3(R − {A} ,C, S(ai)).
9: end if

II. RELATED WORK

Paper: A Novel Unsupervised Classification Approach for Network Anomaly Detection by K Means Clustering and ID3 Decision Tree Learning Methods[8].

Author: Yasser Yasami, Saadat Pour Mozaffari, Computer Engineering Department Amirkabir University of Technology (AUT) Tehran, Iran.

Abstract: This paper presents a novel host-based combinatorial method based on k-Means clustering and ID3 decision tree learning algorithms for unsupervised classification of anomalous and normal activities in computer network ARP traffic. The k-Means clustering method is first applied to the normal training instances to partition it into k clusters using Euclidean distance similarity. An ID3 decision tree is constructed on each cluster. Anomaly scores from the k-Means clustering algorithm and decisions of the ID3 decision trees are extracted. A special algorithm is used to combine results of the two algorithms and obtain final anomaly score values. The threshold rule is applied for making decision on the test instance normality or abnormality.

Conclusion: The proposed method is compared with the individual k-Means and ID3 methods and the other proposed approaches based on markovian chains and stochastic learning automata in terms of the overall classification performance defined over five different performance measures. Results on real evaluation test bed network data sets show that: the proposed method outperforms the individual k-Means and the ID3 compared to the other approaches.

Paper: Privacy Preserving ID3 over Horizontally, Vertically and Grid Partitioned Data [7].

Author: Bart Kuijpers, Vanessa Lemmens, Bart Moelans Theoretical Computer Science, Hasselt University & Transnational University Limburg, Belgium.

Abstract: This consider privacy preserving decision tree induction via ID3 in the case where the training data is horizontally or vertically distributed. Furthermore, we consider the same problem in the

case where the data is both horizontally and vertically distributed, a situation we refer to as grid partitioned data. We give an algorithm for privacy preserving ID3 over horizontally partitioned data involving more than two parties. For grid partitioned data, we discuss two different evaluation methods for preserving privacy ID3, namely, first merging horizontally and developing vertically or first merging vertically and next developing horizontally. Next to introducing privacy preserving data mining over grid-partitioned data, the main contribution of this paper is that we show, by means of a complexity analysis that the former evaluation method is the more efficient.
Conclusion: Here the datasets when partitioned horizontally, vertically and after that the clustering algorithm is applied performs better performance than on the whole datasets.

Paper: A comparison of clustering method for unsupervised anomaly detection in network traffic[5].

Author: Koffi Bruno Yao.
Abstract: Network anomaly detection aims at detecting malicious activities in computer network traffic data. In this approach, the normal profile of the network traffic is modelled and any significant deviation from this normal profile is interpreted as malicious. While supervised anomaly detection models the normal traffic behaviour on the basis of an attack free data set, unsupervised anomaly detection works on a data set which contains both normal and attack data. Clustering has recently been investigated as one way of approaching the issues of unsupervised anomaly detection.
Conclusion: The main goal of the paper has been to investigate the efficiency of different classical clustering algorithms in clustering network traffic data for unsupervised anomaly detection. The clusters obtained by clustering the network traffic data set are intended to be used by a security expert for manual labelling. A second goal has been to study some possible ways of combining these algorithms in order to improve their performance.

Paper: Comparisons between Data Clustering Algorithms [6].

Author: Osama Abu Abbas Computer Science Department, Yarmouk University, Jordan
Abstract: Clustering is a division of data into group s of similar objects. Each group, called a cluster, consists of objects that are similar between themselves and dissimilar compared to objects of other groups. This paper is intended to study and compare different data clustering algorithms. The algorithms under investigation are: k-means algorithm, hierarchical clustering algorithm, self-organizing maps algorithm, and expectation maximization clustering algorithm. All these algorithms are compared according to the following factors: size of dataset, number of clusters, type of dataset and type of software used.
Conclusion: The main conclusion that can be concluded is the performance comparison of different clustering algorithm.

Paper: Dynamic Network Evolution: Models, Clustering, Anomaly Detection[9].

Author: Cemal Cagatay Bilgin and B¨ulent Yener Rensselaer Polytechnic Institute, Troy NY, 12180.
Abstract: Traditionally, research on graph theory focused on studying graphs that are static. However, almost all real networks are dynamic in nature and large in size. Quite recently, research areas for studying the topology, evolution, applications of complex evolving networks and processes occurring in them and governing them attracted attention from researchers. In this work, we review the significant contributions in the literature on complex evolving networks; metrics used from degree distribution to spectral graph analysis, real world applications from biology to social sciences, problem domains from anomaly detection, dynamic graph clustering to community detection.
Conclusion: Many real world complex systems can be represented as graphs. The entities in these system represent the nodes or vertices and links or edges connect a pair or more of the nodes. We encounter such networks in almost any application domain i.e. computer science, sociology, chemistry, biology, anthropology, psychology, geography, history, engineering.

III. Proposed SCHEME

Algorithm :1 K-mean Clustering Algorithm

1) Pick a number (K) of cluster centers (at random)
2) Assign every item to its nearest cluster center (e.g. using Euclidean distance)
3) Move each cluster center to the mean of its assigned items
4) Repeat steps 2, 3 until convergence (change in cluster assignments less than a threshold).

Algorithm : 2 Hierarchical Clustering

Bottom up

1) Start with single-instance clusters
2) At each step, join the two closest clusters
3) Design decision: distance between clusters

Top down

1) Start with one universal cluster
2) Find two clusters
3) Proceed recursively on each subset
4) Can be very fast

Algorithm 3: Density Based Clustering

1) select a point ***p***
2) Retrieve all points density-reachable from ***p*** wrt ε and
3) If ***p*** is a core point, a cluster is formed.
4) If ***p*** is a border point, no points are density-reachable visits the next point of the database.
5) Continue the prsocess until all of the points have been

Algorithm 4: Proposed ID3 Algorithm

Input Layer

- Define P_1, P_2, …., P_n Parties.(Horizontally partitioned).
- Each Party contains R set of attributes A_1, A_2, …., A_R.
- C the class attributes contains c class values C_1, C_2, …., C_c.
- For party P_i where i = 1 to n do
- If R is Empty Then
- Return a leaf node with class value
- Else If all transaction in $T(P_i)$ have the same class Then
- Return a leaf node with the class value
- Else
- Calculate Expected Information classify the given sample for each party P_i individually.
- Calculate Entropy for each attribute (A_1, A_2, …., A_R) of each party P_i.
- Calculate Information Gain for each attribute (A_1, A_2,…., A_R) of each party P_i
- Calculate Total Information Gain for each attribute of all parties (TotalInformationGain( )).
- $A_{BestAttribute}$ ← MaxInformationGain( )
- Let V_1, V_2, …., V_m be the value of attributes. $A_{BestAttribute}$ partitioned P_1, P_2,…., P_n parties into m parties
- $P_1(V_1)$, $P_1(V_2)$, …., $P_1(V_m)$
- $P_2(V_1)$, $P_2(V_2)$, …., $P_2(V_m)$
- . .
- . .
- $P_n(V_1)$, $P_n(V_2)$, …., $P_n(V_m)$
- Return the Tree whose Root is labelled $A_{BestAttribute}$ and has m edges labelled V_1, V_2, …., V_m. Such that for every i the edge Vi goes to the Tree
- NPPID3(R – $A_{BestAttribute}$, C, ($P_1(V_i)$, $P_2(V_i)$, …., $P_n(V_i)$))
- End.

IV. DATASET USED

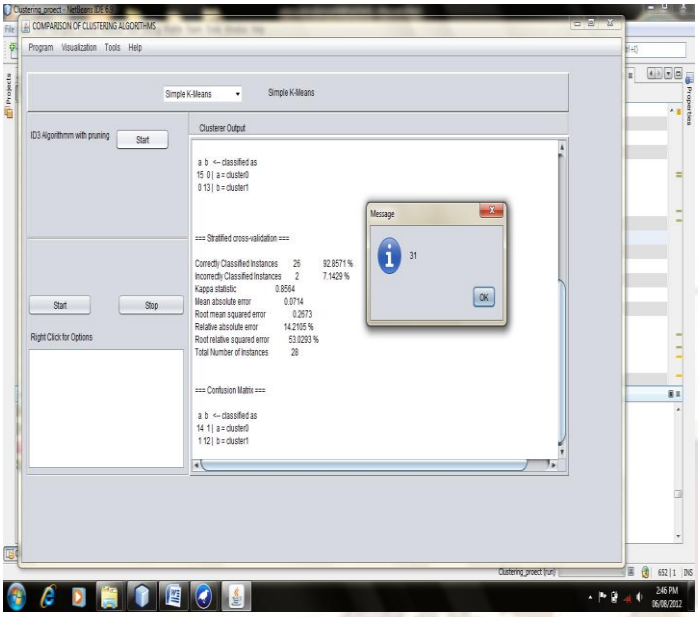
Here we are using a number of attributes and contains a class attribute to find whether an anamoly has been or not.

Node {A, B, C}
Load {Hign,Low}
Transmission {TCP, UDP}
Mac Address {192.168.1.1, 192.168.1.2}
Class_anomaly {yes, no}

A,low,192.168.1.1,udp,no	A,low,192.168.1.1,udp,no
C,low,192.168.1.1,udp,yes	C,low,192.168.1.1,udp,yes
A,high,192.168.1.2,tcp,yes	A,high,192.168.1.2,tcp,yes
B,high,192.168.1.2,tcp,yes	B,high,192.168.1.2,tcp,yes
A,low,192.168.1.1,udp,no	B,low,192.168.1.1,tcp,yes
C,low,192.168.1.1,udp,yes	A,low,192.168.1.1,udp,no
A,high,192.168.1.2,tcp,yes	C,low,192.168.1.1,udp,yes
B,high,192.168.1.2,tcp,yes	A,high,192.168.1.2,tcp,yes
B,low,192.168.1.1,udp,yes	B,high,192.168.1.2,tcp,yes
A,low,192.168.1.1,udp,no	B,low,192.168.1.1,udp,yes
A,high,192.168.1.2,tcp,yes	A,low,192.168.1.1,udp,no
B,low,192.168.1.1,udp,yes	A,high,192.168.1.2,tcp,yes
A,low,192.168.1.1,udp,no	C,high,192.168.1.1,tcp,no
A,high,192.168.1.2,tcp,yes	A,high,192.168.1.2,tcp,yes

A,low,192.168.1.1,udp,no,cluster0	A,low,192.168.1.1,udp,no,cluster0
C,low,192.168.1.1,udp,yes,cluster0	C,low,192.168.1.1,udp,yes,cluster0
A,high,192.168.1.2,tcp,yes,cluster1	A,high,192.168.1.2,tcp,yes,cluster0
B,high,192.168.1.2,tcp,yes,custer1	B,high,192.168.1.2,tcp,yes,cluster0
A,low,192.168.1.1,udp,no,cluster0	B,low,192.168.1.1,tcp,yes,cluster0
C,low,192.168.1.1,udp,yes,cluster0	A,low,192.168.1.1,udp,no,cluster0
A,high,192.168.1.2,tcp,yes,cluster1	C,low,192.168.1.1,udp,yes,cluster0
B,high,192.168.1.2,tcp,yes,cluster1	A,high,192.168.1.2,tcp,yes,cluster0
B,low,192.168.1.1,udp,yes,cluster0	B,high,192.168.1.2,tcp,yes,cluster0
A,low,192.168.1.1,udp,no,cluster0	B,low,192.168.1.1,udp,yes,cluster0
A,high,192.168.1.2,tcp,yes,cluster1	A,low,192.168.1.1,udp,no,cluster0
B,low,192.168.1.1,udp,yes,cluster0	A,high,192.168.1.2,tcp,yes,cluster0
A,low,192.168.1.1,udp,no,cluster0	C,high,192.168.1.1,tcp,no,cluster1
A,high,192.168.1.2,tcp,yes,cluster1	A,high,192.168.1.2,tcp,yes,cluster0

V. RESULT ANALYSIS



number_of_instances	id3_time(ms)	HP_time(ms)
14	78	15
25	93	15
50	110	16
100	125	31
200	150	32

Table 1.

As shown in Table 1. is the time needed for the decision of any dataset. It was observerd that the existing id3takes more time as compared our proposed work.

Where,

HP is the proposed horizontal partioned based ID3.

Relative absolute error can be calculated as: $$\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|\bar{a} - a_1| + \ldots + |\bar{a} - a_n|}$$

Mean squared error can be calculated as: $$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(\bar{a} - a_1)^2 + \ldots + (\bar{a} - a_n)^2}$$

with

Actual target values: a1 a2 … an

Predicted target values: p1 p2 … pn

number_of_instan	ID3 Mean absolute error	HP Mean absolute error
14	0.2857	NP
25	0.24	0.237
50	0.24	0.24
100	0.23	0.22
200	0.235	0.23

Table 2.

In Table 2. Our proposed work performs less means square error as compared to the existing algorithm.

number_of_instances	ID3_Relative absolute error	HP_Relative absolute error
14	60%	NP
25	63.22%	60.13%
50	64.53%	63.24%
100	64.28%	63.63%
200	65.06%	64.28%

Table 3.

In Table 3. It was observed that the proposed algorithm has less absolute error than the existing algorithm.

Clustering with proposed id3	Time (ms)	Mean absolute error	Mean absolute error
K-mean with proposed id3	47	0.0714	14.2105 %
Hierarchical with proposed id3	31	0.0357	36.5854 %
EM with proposed id3	43	0.0238	5.4119 %

Table 4.

As shown in the table 4 is the comparative study of different clustering algorithm with our proposed algorithm.

Clustering with existing id3	Time (ms)	Mean absolute error	Mean absolute error
K-mean with existing id3	65	0.0914	20.2105 %
Hierarchical with existing id3	50	0.0557	45.5854 %
EM with existing id3	60	0.0438	7.4119 %

Table 5

VI. CONCLUSION

The clustering algorithms are used to divide any datasets into a number of clusters, this time clustering algorithms are combined with ID3 algorithm to detect the network anomaly detection and the performance is compared with the other clustering algorithms. The proposed algorithm implemented here provides a way of classifying and provides better leaning of the network anomalies and normal activities in computer network ARP traffic.

REFERENCES

[1] A comparative Study of Anomaly Detection Schemes in Network Intrusion Detection, A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava.

[2] Keogh E., Chakrabarti K., Pazzani M., and Mehrotra S., "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," Knowledge and Information Systems, vol. 3, pp. 263-286, 2001.

[3] Lepere R. and Trystram D., "A New Clustering Algorithm for Large Communication Delays," in Proceedings of 16th IEEE-ACM Annual International Parallel and Distributed Processing Symposium (IPDPS'02), Fort Lauderdale, USA, 2002.

[4] Li C. and Biswas G., "Unsupervised Learning with Mixed Numeric and Nominal Data," IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 4, pp. 673-690, 2002.

[5] A comparison of clustering method for unsupervised anomaly detection in network traffic, Koffi Bruno Yao.

[6] Comparisons between Data Clustering Algorithms,Osama Abu Abbas Computer Science Department, Yarmouk University, Jordan

[7] Privacy Preserving ID3 over Horizontally, Vertically and Grid Partitioned Data,Bart Kuijpers, Vanessa Lemmens, Bart Moelans Theoretical Computer Science, Hasselt University & Transnational University Limburg, Belgium.

[8] A Novel Unsupervised Classification Approach for Network Anomaly Detection by K Means Clustering and ID3 Decision Tree Learning Methods,Yasser Yasami, Saadat Pour Mozaffari, Computer Engineering Department Amirkabir University of Technology (AUT) Tehran, Iran.

[9] Dynamic Network Evolution: Models, Clustering, Anomaly Detection,Cemal Cagatay Bilgin and B¨ulent Yener Rensselaer Polytechnic Institute, Troy NY, 12180.,

[10] Wenke Lee and S. J. Stolfo. Data Mining Approaches for Intrusion Detection, 1998.

[11] Stefano Zanero and Sergio M. Savaresi. Unsupervised learning techniques for an intrusion detection system, ACM March 2004.

[12] Martin Ester, Hans-Peter Kriegel,Jorg Sander,Xiaowei Xu. A density-based clustering algorithm for discovering Clusters in Large Spatial databases with noise. Proceedings of 2nd international Conference on Knowledge Discovery and Data Mining, 1996.

[13] A. Wespi, G. Vigna and L.Deri. Recent Advances in Intrusion Detection. 5th International Symposium, Raid 2002 Zurich, Switzerland, October 2002 Proceedings. Springer.

[14] G. Qu, S. Hariri, and M. Yousif, "A New Dependency and Correlation Analysis for Features," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 9, pp. 1199-1207, Sept. 2005.

[15] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On Combining Classifiers," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226-239, Mar. 1998.