Data Clustering for Anomaly Detection in Network Intrusion Detection

Jose F. Nieves

August 14, 2009

Research Alliance in Math and Science

Contents

List of Figures

Abstract

Intrusions pose a serious security risk in a network environment. Network intrusion detection systems aim to identify attacks or malicious activity in a network with a high detection rate while maintaining a low false alarm rate. New emerging threats or attacks are the most difficult to detect. Signature based methods and misuse detection methods, which rely on labeled patterns, can detect previously known attacks with good accuracy but are unable to detect new types of attacks. In addition, maintaining the signature data base and labeling the patterns is time consuming and expensive. Anomaly detection techniques can make use of unsupervised learning methods to identify new emerging threats with no need of labeled patterns, but, with a potential false alarm rate. We reviewed the different network intrusion detection methods and present here a comparative study with more emphasis on the unsupervised learning methods for anomaly detection. The Kmeans algorithm was chosen to evaluate the performance of an unsupervised learning method for anomaly detection using the Kdd Cup 1999 network data set. The results of the evaluation confirm that a high detection rate can be achieve while maintaining a low false alarm rate.

1 Introduction

A network intrusion is any type of attack or malicious activity that can compromise the stability or security of a network environment. Intruders can be classified in two groups. External intruders don't have authorized access to the system or network they attack, while internal intruders have some authority access to the system. Network intrusions keeps increasing over the years with new emerging and complex threats. This new emerging threats are the most difficult to identify. A network intrusion detection system (NIDS) is one that scan the network activities in a computer environment and attempt to detect the intrusions or attacks. Then, the system administrator may be alerted to the take the corrective actions.

There are generally three types of approaches taken toward network intrusion detection: signature-based, misuse detection and anomaly detection. The signature-based method is the oldest method in practice and depends on a signature database of previously known attacks. Misuse detection is a model-based supervised method which train a classifier with labeled patterns to classify new unlabeled patterns. Anomaly detection approaches can make use of supervised or unsupervised methods to detect abnormal behaviors in patterns.

A main objective of this study is to confirm the advantage of anomaly detection for intrusion detection using a simple clustering algorithm over the Kdd Cup 1999 network data set. Cluto [11], a data clustering software, was used with the Kmeans algorithm to cluster the data. Then, the labeling procedure of the clusters was done to evaluate the performance of the data clustering. The criteria for evaluation was the detection rate, false alarm rate and average cluster purity.

This report is organized as follows. A machine learning background review is presented in the next section. We then present a comparative study of the different network intrusion detections methods with more emphasis on the unsupervised learning methods for anomaly detection. Subsequently we discuss the method, data and software used, and finally we present the results of the evaluation.

2 Machine Learning Background

2.1 Supervised Methods

The main goal of the supervised methods is to build a predictive model (classifier) to classify or label incoming patterns. The classifier has to be trained with labeled patterns to be able to classify new unlabeled patterns. The given labeled training patterns are use to learn the description of classes. Some supervised methods include support vector machines, neural network and genetic algorithms among others.

2.2 Unsupervised Methods

Unsupervised methods, also called data clustering, take a different approach by grouping unlabeled patterns into clusters based on similarities. Patterns within the same clusters are more similar to each other than they are to patterns belonging to different clusters. Data clustering is very useful when little priori information about the data is available. Clustering methods can be classified into two categories: hierarchical clustering algorithms (figure 1) and partitional clustering algorithms (figure 2).

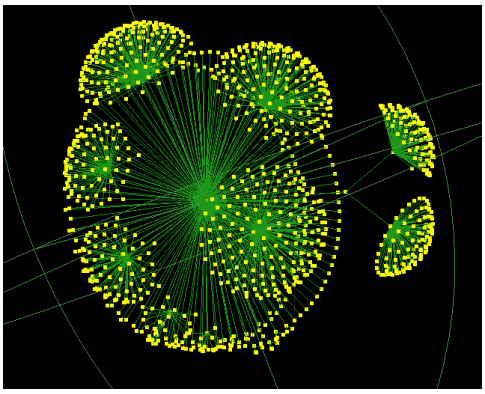

Figure 1: Hierarchical clustering output

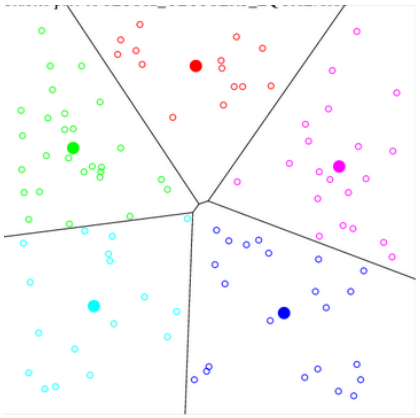

Figure 2: Partitional clustering output

Hierarchical clustering methods can be further divided into agglomerate and divisive. Agglomerative hierarchical clustering start with each pattern in a cluster, then start merging clusters until a stopping criteria is met. Divisive hierarchical clustering start with all the patterns in one cluster and start splitting until a stopping criteria is met. Partitional clustering methods can be further divided into hard and fuzzy. Hard clustering assign each pattern to a cluster while fuzzy clustering assign a membership degree to several clusters for each pattern. Several applications where clustering algorithms have been employed are: image segmentation, document retrieval, data mining, character and object recognition among others.

There are three general basic steps for data clustering: Pattern representation, similarity measure to use and the clustering or grouping. Pattern representation refers to the number of classes, patterns and features of the data. In addition, the feature selection, feature extraction and normalization are apply in this step if necessary. Feature extraction and selection are apply to the data to reduce dimensionality and speed up the process. The feature selection process identify the most effective subset of features from the original data while feature extraction apply a transformation to the original features to produce new features. Normalization is apply to prevent large scale features from dominating the others. In the next step, a similarity measure has to be chosen. Usually a distance based metric is used, but also non-metric similarities can be applied. Distance based metrics are use to quantify the similarity of patterns. The Euclidean metric,

$$d_2\left(\vec{a}-\vec{b}\right)=\left(\sum_{k=1}^{d}\mid a_k-b_k\mid^2\right)^{1/2}$$

a special case of the Minkowski metric, is often used to measure the similarity of patterns with continuous values. The last step is to group the patterns into clusters based on similarities.

3 Network Intrusion Detection Methods

3.1 Signature-based

The signature-based method depend on a signature database of previously known attacks. It is able to detect previously known attacks with good accuracy. The disadvantages of this method are that is unable to detect new emerging threats and the signature database of attacks has to be manually updated, which is expensive and time consuming.

3.2 Misuse detection

Misuse detection methods, a model based supervised method, make use of a classifier that has to be trained with labeled patterns. The training patterns are labeled as 'normal' or 'attacks'. After the classifier is trained, it can classify or label new unlabeled patterns. These methods are also able to detect previously known attacks with good accuracy but also have some disadvantages. They are unable to detect new emerging threats and the labeling procedure of the training data is expensive and time consuming.

3.3 Anomaly detection

Anomaly detection approaches attempt to identify abnormal behavior in patterns and can make use of supervised or unsupervised methods to detect the anomalies or attacks. Unlike the other two methods, these approaches can detect new emerging threats.

The supervised anomaly detection approach train a classifier with just 'normal' labeled patterns. Deviations from 'normal behavior', everything that is not 'normal', are consider attacks. The disadvantage of the supervised methods for anomaly detection is that the labeling procedure of the training data is expensive and time consuming.

The unsupervised anomaly detection approach overcome this problem by making use of data clustering algorithms, which makes no assumption about the labels or classes of the patterns. The patterns are grouped together based on a similarity measure and the anomalies or attacks are the patterns in the smaller clusters. Two assumptions need to be made for this to be true: the normal patterns or connections are many more than the attacks and that the attacks are different than the normal patterns.

The drawback of data clustering for anomaly detection is a potential false alarm rate. Figure 3 shows the output of a clustering procedure with the anomalies or attacks are in the red circles.

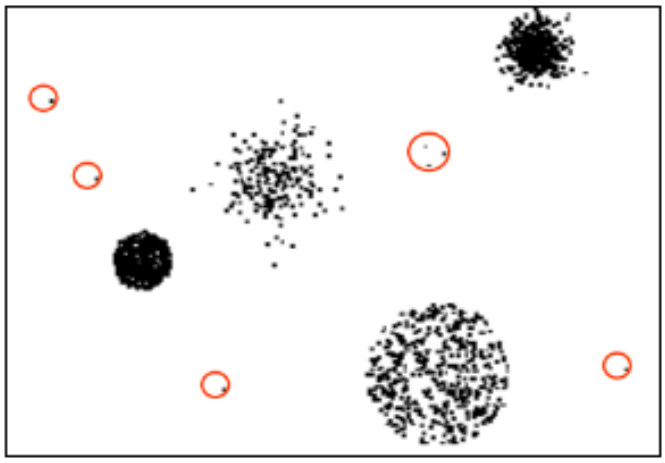

Figure 3: Data clustering for anomaly detection

4 Performance Evaluation

4.1 Method

There are several algorithms that may be used for the clustering procedure. The Kmeans algorithm, a hard partitional clustering algorithm, was chosen for its simplicity and speed with the Euclidean metric as the similarity measure. The general steps for the Kmeans algorithm were the following:

1. Chose number of clusters (K)
2. Initialize centroids (K patterns randomly chosen from data set)
3. Assign each pattern to the cluster with closest centroid
4. Calculate means of each cluster to be its new centroid
5. Repeat step 3 until a stopping criteria is met (no pattern move to another cluster)
6. This procedure was repeat 10 times and the best clustering solution was chosen

The following figure, is an example that shows how the centroids changed position and how the samples are assigned to different clusters for several iterations of the Kmeans algorithm.

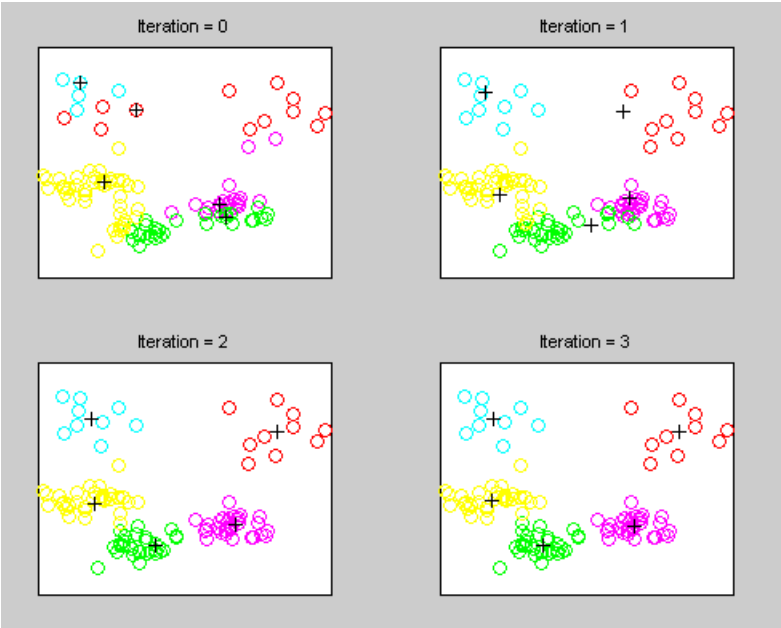

Figure 4: Kmeans output for different iterations

To evaluate our system, the following criteria was used: detection rate and false alarm rate. The detection rate is defined as the number of attacks detected divided by the total number of attacks. The false alarm rate is defined as the number of 'normal' patterns classify as attacks divided by the total number of 'normal' patterns. The labels of the patterns were used for this evaluation, but never used for the clustering procedure.

4.2 Data

The data set used in this study is base on the Kdd Cup 1999 network intrusion data. This data set was created by processing the tcdump portions of the 1998 Darpa evaluation data set, created by MIT Lincoln Laboratories and intended to simulate the traffic seen in a medium size US Air Force base. The Kdd Cup 1999 data set consist of approximately 400,000 data instances or connections that contains 41 features (continuous and categorical) and a total of 22 attacks which fell into the following 4 categories: DoS, Probe, R2L and U2R.

The subset we used consist of 9,200 samples with a total of 10 attacks which fell into the 4 categories mention before. Since a network have many more number of normal connections than intrusions, the attacks were reduced to 2 % of the total samples to be able to simulated a real network. The 3 categorical were encoded to binary values, which expanded the total features to 80. The continuous features were normalized, so their maximum value is one.

4.3 Software

To cluster the data with the Kmeans algorithm the Cluto [11] toolkit was used. Cluto is a software package for clustering low and high dimensional data sets. It provides three different classes of clustering algorithms: partitional, agglomerative and graph-partitioning. It has the option to use a simple interface called gCluto and include a 3-D output visualization of the clusters.

4.4 Results

Based on our assumption that a real network contains many more normal connections than attacks, the smaller clusters are consider to contains attacks and the bigger clusters are consider to contains normal or good connections. But, we may have some normal connections in the smaller clusters and some attacks in the bigger clusters. Using the labels of the patterns we verified how many attacks (detection rate) and how many normal connections (false alarm rate) were in the smaller clusters that we considered to contain attacks.

The clustering procedure was done for 10, 20 and 30 clusters (K). Figure 5 show the output visualization of the clustering procedure for K=10.

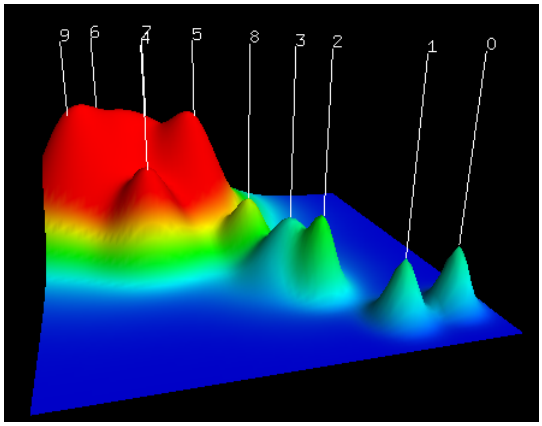

Figure 5: Clustering output visualization for K=10 using Cluto

By observing the data in figure 5, the three smaller clusters (0,1,2) were assume to be the groups containing attacks. As shown in figure 6, the detection rate was not good enough. To increase the detection rate, the five smaller clusters (0,1,2,3,8) were chosen to contain attacks. The detection rate was increased but also the false alarm rate increased to a value that is not acceptable. We then try to increase the purity of the clusters by increasing the total number of clusters (K) to 20 and 30. By increasing the number of clusters to contain attacks, we should be able to detect more attacks but we should also find more normal connections insides those clusters. Figure 6 show the detection rate and false alarm rate for each K.

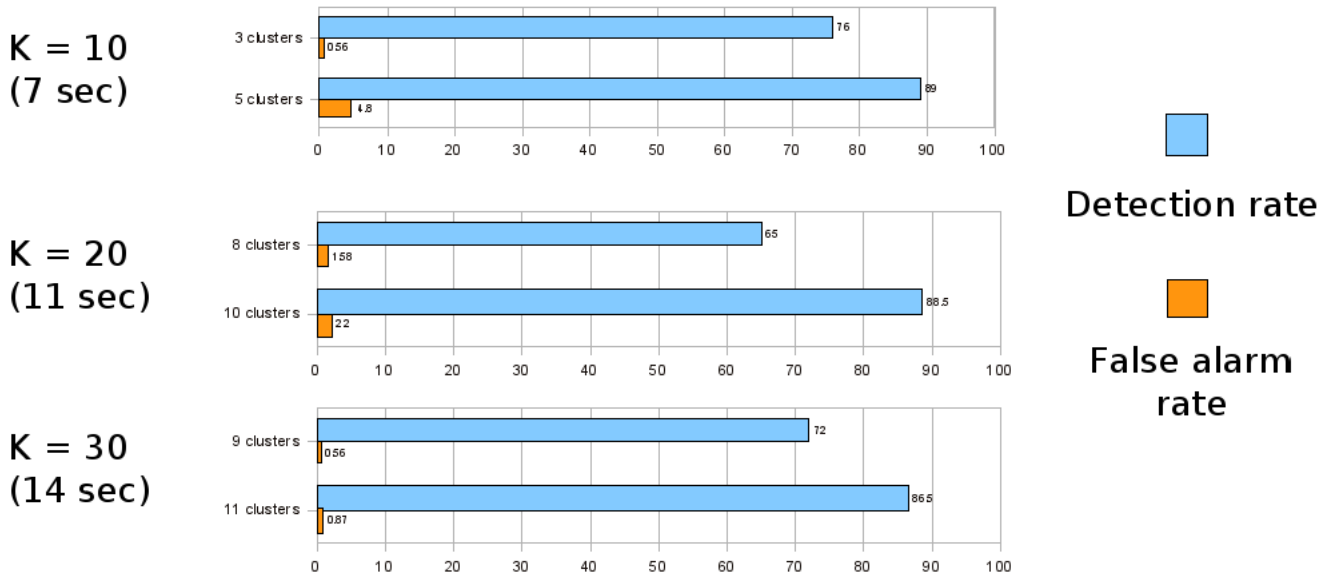

Figure 6: Bar plot of detection and false alarm rate

Clusters	Purity ($0 \rightarrow 1$)
K=10	0.989
K=20	0.995
K=30	0.997

Figure 7: Average purity of clusters

The bar plot show how the number of clusters labeled as attacks affect the detection rate and false alarm rate. Increasing the number of clusters labeled as attacks increase the detection rate, but also increase the false alarm rate. If we try to decrease to false alarm rate by decreasing the number of clusters labeled as attacks, we will also decrease the detection rate. Since we want the detection rate to be as higher as possible while keeping the false alarm rate at a minimum, it is important to choose the right number of clusters to be labeled as attacks.

Increasing the total number of clusters (K) help to achieve a higher detection rate while maintaining a low false alarm rate because the purity of clusters increase. More pure clusters mean that we have more attacks and less amount of normal connection in the smaller clusters and more normal connections and less amount of attacks in the bigger clusters. There is a limit of how many clusters (K) can be used, if K is increased too much we could end with each pattern in a separate cluster, which is useless.

Using data clustering methods for anomaly detection in network intrusion detection, we may achieve a high detection rate of attacks (including previously unseen attacks) while maintaining a low false alarm rate without the need of going through the labeling procedure.

References

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A survey. 2007

[2] Leonid Portnoy, Eleazar Eskin, Sal Stolfo. Intrusion detection with unlabeled data using clustering. 2001

[3] S Zhong, TM Khoshgoftaar, N Seliya. Clustering-based network intrusion detection. 2007

[4] Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification. 2001

[5] Stefano Zanero, Sergio M. Savaresi. Unsupervised learning techniques for a intrusion detection system. 2004

[6] Alexandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, Jaideep Srivastava. A Comparative study of anomaly detection schemes in network intrusion detection. 2003

[7] Varun Chandola, Eric Eilertson, Leven Ertoz, Gyorgy Simon, Vipin Kumar. Data mining for cyber security. 2006

[8] Anita K. Jones, Robert S. Sielken. Computer system intrusion detection: A survey. 2000

[9] Shyam Boriah, Varun Chandola, Vipin Kumar. Similarity measures for categorical data. 2008

[10] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. 2000

[11] George Karypis. University of Minnesota, Computer Science. Cluto: A clustering toolkit. 2003