A CLUSTERING ALGORITHM FOR DETECTING DDoS ATTACKS IN NETWORKS

Dr. K. Sarmila, G. Kavin
Assistant Professor, Research scholar, Department of Computer Science and Engineering,
RVS College of Engineering, Coimbatore, Tamilnadu, India

Abstract— As the number of networked computers grows, intrusion detection system is an essential component in keeping networks secure. Recently data mining methods have gained importance in addressing network security issues, including network intrusion detection a challenging task in network security. Intrusion detection systems aim to identify attacks with a high detection rate and a low false alarm rate. The most widely deployed and commercially available methods for intrusion detection employ signature based detection. However, they cannot detect unknown intrusions intrinsically which are not matched to the signatures, and their methods consume huge amounts of cost and time to acquire the signatures. In order to cope with the problems, many researchers have proposed various kinds of algorithms that are based on unsupervised learning techniques. In this paper, we present a novel clustering based intrusion detection algorithm, unsupervised anomaly detection, which trains on unlabeled data in order to detect intrusions and to improve the detection rate while maintaining a low false positive rate. We evaluated our method using 2000 DARPA Intrusion Detection Scenario Specific Data Set.

Index terms: Anomaly detection, heuristic clustering, true positive rate, false positive rate.

1. INTRODUCTION

With rapid development in the computer based technology, new application areas for computer networks have emerged in Local Area Network and Wide Area Network. This became an attractive target for the abuse and a big vulnerability for the community. Securing this infrastructure has become the one research area. Network intrusion detection systems have become a standard component in security infrastructures. Intrusion detection is “the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network”.

There are generally two types of attacks in network intrusion detection. In misuse detection, each instance in a data set is labeled as ‘normal’ or ‘intrusion’ and a learning algorithm are trained over the labeled data. An anomaly detection technique builds models of normal behavior, and automatically detects any deviation from it, flagging the latter as suspect. Attacks fall into four main categories[8] they are Denial of Service (DoS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, A remote to user (R2L) attack is class of attacks where an attacker sends packets to a machine over a network, then exploits machine’s vulnerability to illegally gain local access a user, User to root exploits is a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Probing is a class of
attacks where an attacker scans a network to gather information or find known vulnerabilities.

The main reason for using Data Mining Techniques for intrusion detection system is due to the enormous volume of existing and newly appearing network data that require for processing. The data accumulated each day by a network is huge. Several data mining techniques such as clustering, classification, and association rules are proving to be useful for gathering different knowledge for intrusion detection.

Unsupervised Anomaly Detection (UAD) algorithms have the major advantage of being able to process unlabeled data and detect intrusions that otherwise could not be detected. The goal of data clustering, or unsupervised learning, is to discovery a “natural” grouping in a set of patterns, points, or objects, without knowledge of any class labels.

We need a technique for detecting intrusions when our training data is unlabeled, as well as for detecting new and unknown types of Intrusions. We evaluated our clustering method over a data set of network connections. The network data we examined was from DARPA 2000 [9], which is a very popular and widely used intrusion attack data set. Our experimental results show that the detection rate of the proposed method consistently outperforms other existing algorithms reported in the literature; especially at the false positive rate.

2. RELATED WORKS

Network-based computer systems play increasingly vital roles in modern society, security of network systems has become important than ever before. As an active defense technology, IDS (Intrusion Detection Systems) attempts to identify intrusions in secure architecture. In 1970s, the U.S Department of Defense outlined security goals for audit mechanisms, among which were allowing the discovery of attempts to bypass protection mechanism.

Over the past years there are many different types of intrusions, and different detectors are needed to detect normal and anomalous activities in network. Some Clustering algorithms have recently gained attention in related literature, since they can help current intrusion detection systems in several respects.

Several existing supervised and unsupervised anomaly detection schemes and their variations are evaluated on the DARPA 1998 data set of network connections [3] as well as on real network data using existing standard evaluation techniques.

Recently, anomaly detection has been used for identifying attacks in computer networks [1], malicious activities in computer systems and misuse in web systems [6], [7]. A more recent class of Anomaly Detection Systems developed using machine learning techniques like artificial neural-networks [8], Kohonen’s self organizing maps [9] fuzzy classifiers and others [7] have become popular because of their high detection accuracies at low false positives. Data mining for security knowledge [2] employs a number of search algorithms, such as statistical analysis, deviation analysis, rule induction, neural abduction, making associations, correlations, and clustering.

Another data mining approach [6] for intrusion detection is the application of clustering techniques for effective intrusion detection. However, the existing works that is based on either K-Means or K-Medoids have two shortcomings in clustering large network datasets namely number of clusters dependency and lacking of the ability of dealing with character attributes in the network transactions. Among these the number of clusters dependency suggests that the value of K is very critical to the clustering result and should be fixed before clustering.

This paper presents the idea of applying data mining techniques to intrusion detection systems to maximize the effectiveness in identifying attacks, thereby helping the users to construct more secure information systems.

Contributions of the paper

The rest of the paper is organized as follows. In Sect. 3, we present Heuristic clustering algorithm. In Sect. 4, we describe the details of our experiment and present the results and their analysis. Finally, we present concluding remarks and suggestions for future study.

3. INTRUSION DETECTION SYSTEM

Clustering is one of unsupervised learning techniques to group data instances into meaningful subclasses. In this section, we describe the Heuristic Clustering Algorithm, and illustrate how to apply this algorithm to generate detection models from audit data.

The Overall clustering process of the clustering algorithm and labeling as follows, is composed of two phases: Clustering Process and Labeling (shown in Fig 3.1). In the first phase, partitions the training instances into clusters using heuristic clustering. In the second phase using labeling detects the normal and anomaly instances.

3.1 Data preprocessing

Depending on the monitoring data and the data mining algorithm, it may be necessary or beneficial to perform cleaning and filtering of the data in order to avoid the generation of misleading or inappropriate results. Extract nine attributes values from the dataset and store into the database. Attributes are Source IP address, destination IP address, Protocol, Source Port, destination port, Sequence number, Acknowledgment number, length, window size.

3.2 Clustering Algorithm

Notations of Some Terms

Some notations needed in heuristic clustering algorithm are:
Notation 1: Let $H = \{H_1, H_2, \ldots, H_m\}$ be a set of attribute values, the $m$ is number of attribute values. Some attribute values are duration, src-bytes, dest-bytes, flag, etc.,
Notation 2: Let $H = H_N \cup H_S$ and $H_N \cap H_S = \emptyset$, where $H_N$ is the subset of numerical attribute.
(e.g., no of bytes), and $H_S$ is the subset of character attribute. (e.g., service, Protocol).

Notation 3: Let, $e_i = (h_{i1}, h_{i2},...,h_{im})$, $e_i$ is a record, the $m$ is number of attribute values and $h_{ij}$ is the value of $H_m$.

Notation 4: $E = \{e_1, e_2,...,e_n\}$, $E$ is the set of records; $n$ is the number of packets.

**Heuristic Clustering Algorithm (HCA)**

In the HCA algorithms we don’t need fix the value of $K$ in the beginning of clustering. We use the heuristic clustering technology to determine the number of cluster automatically. We compute the similarity between $e_i$ and every center of cluster: $\text{Sim}(e_i, C_j)$ and the similarity between every center of cluster $\text{Sim}(C)$, if the minimal of $\text{Sim}(e_i, C_j)$ is more than the minimal of $\text{Sim}(C)$ , we create a new cluster, and the center of cluster is $e_i$ , otherwise we insert the $e_i$ into $C_j$.

Step 1. Confirm two initial cluster centers by algorithm search ( ).
Step 2. Import a new record $e_i$. Repeat 3 to 5 until no more records.
Step 3. Compute the similarity by algorithm Similar ( ), and find $\text{Min}(\text{Sim}(e_i, C_j))$, $\text{Min}(\text{Sim}(C))$.
Step 4. If $\text{Min}(\text{Sim}(e_i, C_j)) > \text{Min}(\text{Sim}(C))$ then $C_{k+1} = \{e_i\}$, $C = \{C_1, C_2,...,C_{k+1}\}$, create a new cluster, and the $e_i$ is the center of the new cluster.
Else $C_j=C_j U \{e_i\}$, insert $e_i$ into $C_j$.

**The initial center of clustering**

In the beginning of clustering, we should confirm two initial center of clustering by the algorithm Search ( ).

Algorithm: Search_m ($E,l$).
Import data set $E$, the number of sampling $l$.
Output: initial center $m_1,m_2$.
(1)Sampling $E$, get $S_1,S_2,...,S_l$
(2)For $i=1$ to $l$ do
    mi=Count_m($S_i$)
(3)For $i=1$ to $l$ do
    m=Count_m($m_i$)
    // $m$=center $\{m_1,m_2, m_3,...,m_l\}$
(4) $m_1=m$ , $m_2=\text{max}(\text{Sim}(m, m_i))$

**The method of computing similarity**

The audit data consists of numerical attribute (e.g. the receive Bytes) and character attribute. (E.g. Protocol). But whether k-Means or K-medoids all lack the ability of dealing with character attribute. The HCA algorithm resolves this problem by processing the character attribute using the method of attribute matching.

The similarity of character attributes:

Let $e_i$ and $e_j$ be two records in the $E$. all containing $m$ attributes (including $P$ character attributes), the $n_{h_{ik}}$ and $n_{h_{jk}}$ is the number of $h_{ik}$ and $h_{jk}$ respectively.

$$\text{Sim}^P(e_i, e_j) = \sum_{k=1}^{p} \frac{n_{h_{ik}} \times n_{h_{jk}}}{(n_{h_{ik}} + n_{h_{jk}}) \times A}$$

if ($h_{ik}=h_{jk}$) then $A=0$ else $A=1$.

The similarity of numerical attribute (to the numerical attribute, still use the classical Euclidean distance to compute similarity)

$$\text{Sim}^N(e_i, e_j) = \sqrt{\sum_{k=1}^{q} |h_{ik} - h_{jk}|^2}$$

The similarity of two records (including similarity of numerical attribute and similarity of character attribute)

$$\text{Sim}(e_i, e_j) = \text{Sim}^N(e_i, e_j) + \text{Sim}^P(e_i, e_j)$$

**The center of cluster**

A cluster are represented by its cluster center .In the HCA algorithm, we use the algorithm Count ( ) to compute the cluster center. The center of a cluster is composed of the center of numerical attributes and character attribute.
Let \( P = (P_N + P_S) \), and \( P = (P_1, P_2, \ldots, P_m) \)
where \( P_N \) is the center of numerical attribute, the \( P_S \) is the center of character attribute,
\[
P_N = \frac{1}{n} \sum_{j=1}^{n} h_{ji} \quad i= 1,2,\ldots, p (p<= m)
\]
The \( h_{ji} \) is the numerical attribute. The \( P_S \) is a frequent character attribute set which consists of \( q (q=m-p) \) most frequent character attribute.

**Labeling**

We propose a method to detect anomaly that does not either depend on the population ratio of the clusters. In our labeling method, we assume that center of a normal cluster is highly close to the initial cluster center \( v_h \) which are created from the clustering.

In other words, if a cluster is normal, the distance between the center of the cluster and \( v_h \) will be small, otherwise it will be large. Thus, we first, for each cluster center \( C_j (1 \leq j \leq k) \), calculate the maximum distance to \( v_h \). We then calculate the average distance of the maximum distances. If the maximum distance from a cluster to \( v_h \) is less than the maximum average distance, we label the cluster as normal. Otherwise, label as attack.

**4. EXPERIMENTS AND RESULTS**

In this section, we present the results of the Heuristic clustering method compare it with K-Means over the dataset 2000 DARPA. In heuristic clustering the relations between categorical and continuous features are handled naturally, without any forced conversions (k-means) between these two types of features. Fig 3.2 shows how can we detect the anomaly instances.

We use the measures for evaluating the performance
1. TPR or recall is the percentage of anomaly instances correctly detected,
2. FPR is the percentage of normal instances incorrectly classified as anomaly,
3. “Precision” is the percentage of correctly detected anomaly instances over all the detected anomaly instances,
4. “Total accuracy” or “accuracy” is the percentage of all normal and anomaly instances that are correctly classified.

The identification of normal and abnormal classes has been represented in the form of a matrix called Confusion matrix as given in Table 4.1.

**Table 4.1 Confusion Matrix**

<table>
<thead>
<tr>
<th>Actual \ Predicted Class</th>
<th>Normal</th>
<th>Abnormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>True positive (TP)</td>
<td>False negative (FN)</td>
</tr>
<tr>
<td>DDoS1</td>
<td>False positive (FP)</td>
<td>True negative (TN)</td>
</tr>
</tbody>
</table>

*Fig 4.1 Assigning Scores to IDS*
The detection rate is computed using the equations

Detection Rate = \( \frac{TN}{FN + TN} \times 100 \)

False Positive Rate = \( \frac{FP}{TP + FP} \times 100 \)

Error Rate = \( \frac{(FP + FN)}{(TP + FP + FN + TN)} \times 100 \)

The results obtained from heuristic clustering based intrusion detection accuracy is 93.05 percent at a false-positive-rate of 3.08 percent on 10% subset of the training dataset 2000 DARPA. Table 4.2 depicts the results obtained from the DARPA by using the heuristic Clustering approach. For each iteration the training instances taken for testing and the corresponding Detection accuracy, false positive rate is shown in fig 4.2.

Table 4.2 Results for anomaly detection using heuristic clustering.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Training instances</th>
<th>Detection Rate</th>
<th>False Positive Rate</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>415</td>
<td>90.9%</td>
<td>2.8%</td>
<td>3.4%</td>
</tr>
<tr>
<td>1</td>
<td>820</td>
<td>94.7%</td>
<td>2.5%</td>
<td>2.9%</td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
<td>89.0%</td>
<td>4.4%</td>
<td>5.4%</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>87.0%</td>
<td>6.4%</td>
<td>7.5%</td>
</tr>
<tr>
<td>4</td>
<td>5200</td>
<td>89.3%</td>
<td>2.4%</td>
<td>3.1%</td>
</tr>
<tr>
<td>5</td>
<td>6130</td>
<td>89.7%</td>
<td>1.9%</td>
<td>2.7%</td>
</tr>
<tr>
<td>6</td>
<td>7117</td>
<td>88.3%</td>
<td>3.1%</td>
<td>3.9%</td>
</tr>
<tr>
<td>7</td>
<td>8045</td>
<td>90.7%</td>
<td>6.1%</td>
<td>7.3%</td>
</tr>
<tr>
<td>8</td>
<td>9300</td>
<td>87.5%</td>
<td>2.8%</td>
<td>3.6%</td>
</tr>
<tr>
<td>9</td>
<td>11245</td>
<td>91.7%</td>
<td>5.3%</td>
<td>5.4%</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND FUTURE WORKS

This paper Network intrusion detection with heuristic clustering incorporates the idea of applying data mining techniques to intrusion detection system to maximize the effectiveness in identifying attacks, thereby helping the users to construct more secure information systems. The main advantage of our algorithm is that the relations between categorical and continuous features in 2000 DARPA data set are handled naturally, without any forced conversions (k-means) between these two types of features.

For future work, we need to verify performance of the proposed clustering algorithm over real data and make a new benchmark dataset for intrusion detection.

Future directions in this research area can be extracting more variables and to develop an advanced detection algorithm to suit these multiple variables. The new system may be tested with more than one benchmark data set
REFERENCES


