Network Intrusion Detection using Clustering: A Data Mining Approach

S.Sathya Bama
VLB Janakiammal College of Engineering and Technlogy, Coimbatore.

M.S.Irfan Ahmed
Hindusthan College of Engineering and Technology, Coimbatore.

A.Saravanan
VLB Janakiammal College of Engineering and Technlogy, Coimbatore.

ABSTRACT
Network intrusion detection system includes identifying a set of spiteful actions that compromises the basic security requirements such as integrity, confidentiality, and availability of information resources. The enormous increase in network attacks has made the data mining based intrusion detection techniques extremely useful in detecting the attacks. This paper describes a system that is able to detect the network intrusion using clustering concept. This unsupervised clustering technique for intrusion detection is used to group behaviors together depending on their similarity and to detect the different behaviors which are then grouped as outliers. Obviously, these outliers are attacks or intrusion attempts. This proposed method which uses data mining technique will reduce the false alarm rate and improves the security.

Key Words
Intrusion Detection, Security, Clustering, Classification, Common Outliers.

1. INTRODUCTION
In this computer era, almost all the companies and organizations have their computers connected to the network. Network-based attacks on business computers have been increasing in frequency and severity over the past several years. As a result, many research and other organization's efforts have concentrated on network intrusion detection techniques whose aim is to identify such attacks. Intrusion Detection Systems (IDS) are proposed to protect information systems against intrusions and attacks and to close security gaps of operating systems and network access controls. However, many intrusion detection systems are based on traditional methods which need a prior knowledge about the security flaws. The main drawback in this traditional approach is that only the known attacks can be detected by using audit records [15]. Therefore, new kinds of attacks have to be updated to the audit record frequently. Unfortunately, the recent discovery of a new security flaws for instance, the IDS will ignore it since this new attack has not yet been updated in the audit record. In severe cases of security breach, companies may lose business, and eventually become bankrupt, as a result of one attack [2].

Security attacks come from different sources. Natural disasters such as earth-quakes, floods, etc can damage essential information. But, completely different threats come from people known as intruders, e.g. unauthorized users of computers. There are external intruders or masqueraders, who are unauthorized users of the machines they attack, and internal intruders or misfeasors, who have permission to access the system with a number of restrictions and external/internal intruders or clandestine user, who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection. Several techniques have been used to prevent unauthorized access to the data; some suitable to prevent the access by external and internal intruders, while others only prevent the access by external intruders [15]. In this paper we mainly concentrate on preventing the access from external intruders.

2. RELATED WORK
Identifying new attacks and protecting a system, by using suitable approach is an important topic in this security domain. One such approach relies on data mining concepts. Data mining is an important tool, which provides the Intrusion detection system with more automatic detection of network attacks [8, 14]. Among those data mining approaches, anomaly detection tries to deduce intrusions [3, 9]. The overall method used in this paper is to build clusters or groups of usage data and find outliers (i.e. the set of events that are considerably dissimilar from the remainder of the normal usage data) [5]. However, the drawback of detecting intrusions by means of anomaly (outliers) detection is the high rate of false alarms since an alarm can be triggered because of a new kind of usages that has never been seen before though it is not an attack (and is thus considered as abnormal). Considering the large amount of new usage patterns emerging in the Information Systems, even a weak percent of false positive will give a very large amount of spurious alarms that would be overwhelming for the analyst [1, 10].

The main objective of this paper is to propose an intrusion detection algorithm based on data mining technique that is based on the analysis of usage data coming from multiple partners in order to reduce the number of false alarms. On the other hand, when a new security flaws has been found on a system, the hackers will want to use it in as many information systems as possible. Thus a new anomaly that occurs on two or more information systems is probably not a new kind of usage, but rather an intrusion attempt [4]. Based on the analysis of the usage data coming from the different partners, our algorithm will detect the common outliers they share. Such common outliers are likely to be true attacks and will trigger an alarm. Thus this method will reduce the false alarm rate.

The paper is organized as follows. In section 3 we present the novel method for intrusion detection using clustering and classification technique. Section 4 presents the performance analysis with the existing system. And finally section 5 gives the conclusion and future work.

3. PROPOSED INTRUSION DETECTION SYSTEM

An intrusion can be defined as an action aimed at compromising the security requirements such as confidentiality, integrity or availability of data. This includes unauthorized attempts to access data, manipulate data or make the system not viable [7]. An algorithm aimed to detect the outliers shared by a networked organization has been proposed to detect the intrusion. Outliers are usually small clusters and the goal is to use outlier lists from different systems (based on a similar clustering, involving the same similarity measure). If an outlier occurs for at least two systems, then it is considered as an attack based on the assumption that an intrusion attempt that tries to find a weakness of a script will look similar for all the victims of this attack. Once the intrusion has been detected successfully then the administrator can properly set up a network to be more secure.

An algorithm is applied, to perform a clustering on the usage patterns of each site and to find the common outliers. The first step for clustering the patterns of each site is to find the similarity between the patterns. The similarity measure (presented in section 3.2) will allow normal usage patterns to be grouped together and distinguishes an intrusion pattern from normal usage patterns and from other intrusion patterns (since different intrusion patterns will be based on a different security hole and will have very different characteristics). The algorithm performs successive clustering for each site. At each step we check the potentially matching outliers between both sites. The clustering algorithm is agglomerative and depends on the similarity measure respected between two objects. Then, the alarms will be triggered at each step of the monitoring (for instance for every one hour). The assumption is that common outliers, sorted by similarity from one site to another, will be added to the intrusions list.

3.1 IDS Algorithm

As explained above, an algorithm will process the usage patterns of both sites step by step. For each step, clustering has been done with the usage pattern and analyzed for intrusion detection. The overall algorithm is shown below:

Algorithm:
Input : *P_1* and *P_2* the usage patterns of sites *S_1* and *S_2*
Output : *Op* the set of common outliers corresponding to malicious patterns.

```
Begin
  For all objects or usage patterns P1 and P2
    C1 = Clustering (P1); //C1 is the set of clusters in site S1
    C2 = Clustering (P2); //C2 is the set of clusters in site S2
    O1 = Outliers (C1); //O1 is the outliers in site S1
    O2 = Outliers (C2); //O2 is the outliers in site S2
    If Common Outliers (O1,O2) ≠ NULL then
      Trigger the alarm; Op = Op ∪ Common Outliers (O1,O2);
    End if
  Next for
End algorithm
```

3.2 Similarity between objects

A cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. So it is needed to compute the similarity between the objects before clustering them. Here each object is a sequence of characters. In sequenced data, the larger the number of common characters and the more identical the order of the characters shared between sequences, the greater the degree of similarity between the sequences. Therefore, we must seek for common subsets of characters with the same order that exist among the sequences. A pair of characters having the same order is found between the sequences; the more times a pair of identical characters are found in two sequences, the greater the similarity of the sequences.

Definition 1: Sequence $Sq = \langle x_1 x_2 \ldots x_i \ldots x_j \ldots x_n \rangle$ is an ordered list of characters, where x_i is a character. The number of characters in S is referred to as the size of Sq and denoted by $|Sq|$. A sequence element e_k is a pair of characters, $x_i x_j$ $(i<j)$, in sequence Sq. E = (e_1, e_2,….. e_k,….) is the collection of sequence elements e_k. The number of elements in E is referred to as the size of E and is denoted by $|E|$.Both the characters in the sequences and the order of the characters in sequences are used to measure the similarity between those sequences.

An efficient method to measure the Maximum dissimilarity [13] is defined as

$$\mathrm{sim}(Sq_1, Sq_2) = \frac{|E_3 \cap E_4|}{\frac{|E_1| + |E_2|}{2}}$$

Here, E_3 be the collections of elements of Sq_3 and E_4 be the collections of elements of Sq_4, where Sq_3 and Sq_4 are generated sequences from Sq_1 and Sq_2. Sq_3 is generated as follows. All of the characters in Sq_1 are compared with those of Sq_2 sequentially. If the identical characters exists in Sq_2, then the characters is inserted into Sq_3. By the same method, Sq_4 is generated by comparing all characters with Sq_1. Therefore, computation of similarity between Sq_3 and Sq_4 is more efficient than computation of similarity between Sq_1 and Sq_2.

Example: Consider the sequences Sq_1 = prevail and Sq_2 = prevent. $|E_1| = 6$ and $|E_2| = 6$. Sq_3 = <prev> and Sq_4 = <prev>. $E_3 \cap E_4$=<pr,re,ev> i.e.,$|E_3 \cap E_4| = 3$. sim(Sq_1,Sq_2) = 3/6. The computed similarity measure between sequences Sq_1 and Sq_2 is 1/2. This means a similarity between the two objects is of 50%.

3.3 The Clustering Method

Clustering algorithms are either of type partitioned or hierarchical methods. The algorithms studied on clustering of categorical sequences [6, 11] use an edit distance or sequence alignment method for finding the similarity between sequences. An agglomerative hierarchical clustering algorithm [13] is used here for clustering sequences. Consider the problem of clustering n sequences of characters. First, each of the (n) x (n-1)/2 pairs of possible merges is evaluated, and the two clusters that have maximum value of the criterion function are merged. After performing m merging steps, each of the (n-m) x (n-m-1)/2 pairs possible merges is evaluated. This process continues until there are only k clusters left. The criterion function [13] used is

$$\text{Maximize } C_f = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{i,j=c} \mathrm{sim}(i, j)$$

where n_r is the number of sequences in C_r and k is the number of clusters. The method used for clustering the remaining sequence data is the k-nearest-neighbor (k-nn) method. This method merges a new sequence with one of the generated clusters by computing the similarity between the new sequence and sequences of the clusters and finds a cluster having the most k-nearest neighbors out of it. If an equal number of k-nearest-neighbors exists for more than one cluster, choose one cluster randomly.

3.4 The Proposed Clustering and Classification Method

In the proposed method, if the data or object to be clustered is larger, then the clustering is initially done on random sample data rather than on the entire dataset for easy processing. To cluster the remaining sequence, a new approach is applied for finding a cluster with more neighbors. The existing method suffers from more computations between cluster sequences and a new sequence to be clustered and also no criterion is applied on clusters to check for new sequence characters in the clusters before the computation takes place. Hence, there is m x n computations for m clusters with n sequences. To overcome the shortcomings of the existing method, a new approach has been conceived with an idea of clustering a new sequence with one of the existing clusters. It is finding the frequency of common pairs in each cluster for the new sequence using cluster index and choosing a cluster having maximum number of common pairs with most k-nearest neighbors for merging. This reduces the number of computations considerably and performs better than the existing method.

Once a user specified number of clusters have been generated using random sample data, remaining sequences are clustered using dynamic cluster indexes. Initially, a dynamic cluster index table is created for each cluster with random sample, using an idea derived from a study on association rule mining using index table [12]. Each cluster index table as illustrated in Table 2 consists of two fields namely 'Character Key' and 'List of Pairs with Count' which are sequence characters in the cluster and the corresponding sequence element pairs with their count. Here, the index table preserves memory by having the list of characters field as a variable length field. It allows us to store many characters with their pair count in a row separated by delimiter. A character having no sequence element pair is not added to the cluster index. The cluster index is said to be dynamic because when a new sequence arrives it could dynamically be extended by introducing new characters. For example, if a new sequence Sq_i is merged with a cluster C_i, check for characters in Sq_i but not in cluster index of C_i. Then, those characters not in C_i are added to the cluster index of C_i. And, also for each common pair found between Sq_i and C_i, their counts are updated by incrementing them in the cluster index. The steps of the new algorithm are illustrated in Figure 1.

Proposed Classification Algorithm:

Input : A new sequence to be clustered
Output : A cluster with K-nearest neighbors

Step 1: Let Sq_i be the sequence to be clustered.

Step 2: Check for common characters between Sq_i and cluster C_i using index key of cluster index. If there is no common character found, go to step 2 for next cluster else go to step 3.

Step 3: Generate sequence element pairs of the common character set and find their counts using index key and list of pair characters in the cluster index.

Step 4: Do scaling for each common pairs count by dividing it with number of sequences in the cluster. Then, choose a cluster C_i having most common pairs with maximum count. If there is more than one cluster having equal number of common pairs with same count, choose one cluster randomly.

Step 5: Assign S_i to the cluster selected in Step 4.

Step 6: Go to Step 1 if sequences remain to be clustered, otherwise exit the program.

Example: Three clusters consisting of sampled sequences are shown in Table 1. Each cluster contains three sequences. Table 2 gives the structure of cluster index for cluster 1.

Table 1. Clusters with Sampled Sequences

C_1	C_2	C_3
Sq_1=<abcj>	Sq_3 = <ejfk>	Sq_4= <gjki>
Sq_2=<acjk>	Sq_5 = <jfdk>	Sq_6 =<hijk>
Sq_7=<abj>	Sq_8 = <djkfe>	Sq_9 = <ghi>

Table 2. Structure of C_1's Index Table

Character Key	List of Pairs with Count
a	b:2,c:1,j:3,k:1
b	c:1,j:2
c	j:2,k:1
j	k:1

Here, we show how to cluster the new sequences Sq_{10} = <abk> and Sq_{11} = <cjkf>. First, the sequence characters of Sq_{10} are compared with character key of C_1's cluster index. For each common character, pairs are generated and their count is found from the cluster index. The same process is repeated for the remaining clusters. There is no common character found between sequence Sq_{10} and clusters C_2 and C_3. Hence, Sq_{10} is assigned to C_1. The above said process is done for sequence Sq_{11}. For this sequence, clusters C_1 and C_2 are only having common pairs. However, common pairs count in C_2 is maximum compared to C_1 and Sq_{11} is assigned to C_2.

3.5 Common Outliers Detection

The common outlier detection can be done by comparing the two lists of outliers. For each pair of outliers, it calculates the similarity between them. If this similarity is below the user threshold, then we consider those outliers as similar and we add them to the alarm list.

4. PERFORMANCE ANALYSES

The performance analysis is concentrated on the execution time of the algorithms by reducing the number of computations between sequences using dynamic cluster indexing approach and on the number of false alarm rate. The algorithms are analyzed for k-nearest neighbor approach with CTI dataset for the effect of number of sequences in the clusters. This data set contains the preprocessed and filtered data. The data is based on a random sample of individual users visiting the site for a 2 week period during April of 2002. The original (unfiltered) data contained a total of 20950 sessions from 5446 users. The filtered data contains

13745 sessions and 683 page views. Here each user's sites has been clustered individually and the outliers are listed which in turn finds the common outliers.

Our algorithm outperforms the existing algorithm in finding a cluster having k-nearest neighbors. Our experimental results are shown in Figure 1.

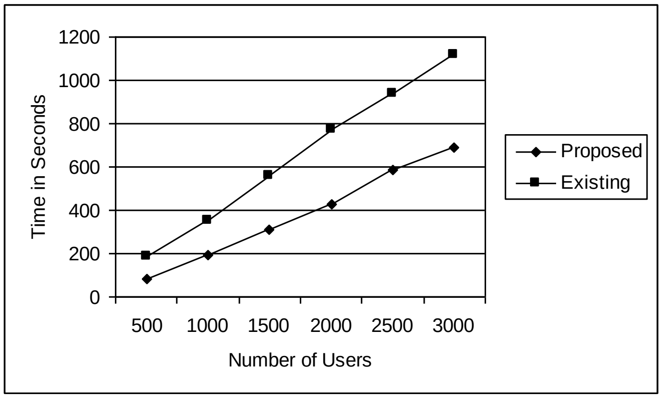

Fig 1: Results of the sample dataset

Table 2 shows the execution time of the proposed and existing algorithms for the sequence dataset with various sample sizes. Since the proposed algorithm uses dynamic cluster indexes, it has significant effect on the execution time compared to existing algorithm. The false alarm rate has also been reduced eventually.

Table 3. Comparison of Algorithms

Number of users	Execution Time (in Seconds)	
	Proposed Algorithm	Existing Algorithm
500	185	285
1000	290	452
1500	412	661
2000	526	871
2500	687	1040
3000	791	1216

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel approach for unsupervised clustering scheme for isolating malicious behaviors. This proposed method is used to find the outliers in various sites and to reduce the false alarms in various sites. For this it checks for the common outliers in all the sites which are then viewed as a network intrusion. Future work aims at experimental evaluation and the comparison with existing method using real network audit data set and to exploit other data mining techniques such as association rule mining for network intrusion detection system and for other network security issues.

6. REFERENCES

[1] E. Bloedorn, A. D. Christiansen, W. Hill, C. Skorupka, and L. M. Talbot. Data mining for network intrusion detection: How to get started. Technical report, MITRE, 2001.

[2] Buyer's guide for intrusion prevention systems (IPS), June 3, 2004 (http://www.juniper.net/solutions/literatur/buryerguide/710005.pdf)

[3] Dunigan and Hinkel, \Intrusion detection and intrusion prevention on a large network, a case study," in Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring, Apr. 1999.

[4] Goverdhan Singh, Florent Masseglia, et al Data Mining for Intrusion Detection: from Outliers to True Intrusions, published in "The 13th Pacific-Asia conference on Knowledge Discovery and Data Mining (PAKDD-09)(2009) 891-898"

[5] J. Han and M.Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, pp-335-393,2001.

[6] B.Hay, G. Wets and K.Vanhoof, "Clustering Navigation Patterns on a Website using a Sequence Alignment Method", 2001 Int.Joint Conf.on Artificial Intelligence,2001.

[7] Kazimierz Kowalski and Mohsen Beheshti Improving Security Through Analysis of Log Files Intersections, International Journal of Network Security, Vol.7, No.1, PP.24{30, July 2008

[8] W. Lee, S. J. Stolfo, and P. K.Chan, \Real Time Data Mining-based Intrusion Detection," in Proceedings Second DARPA Information Survivability Conference and Exposition, 2001. (http://citeseer.ist.psu.edu/452795.html)

[9] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In 7th conference on USENIX Security Symposium, 1998.

[10] A. Marascu and F. Masseglia. A multi-resolution approach for atypical behaviour mining. In The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09), Bangkok, Thailand, 2009.

[11] T. Morzy, M.Wojcieechowski and M.Zakrzewicz, "Scalable Hierarchical Clustering Method for Sequences of Categorical Values", Proc.5th Pacific-Asia Conf. Knowledge Mining, Kowloon,HongKong 2001.

[12] H Ravi Sankar, M M Naidu and K Swapna Devi, "An Algorithm for Finding Frequent Itemsets Using Index Key", Journal of the CSI, Vol.37 No.3,30-34,July-September 2007.

[13] Seung-Joon Oh, Jae-Yearn Kim, "Clustering Categorical Sequences Using a K-Nearest-Neighbor Method" International Journal on Computer Applications, Vol 12 No.3,141-150,Sept 2005.

[14] A. Valdes and K. Skinner. Probabilistic alert correlation In Recent Advances in Intrusion Detection, pages 54–68, 2001.

[15] William Stallings, Cryptography and Network Security, Third Edition.