Integrating Genetic Algorithms and Fuzzy c-Means for Anomaly Detection

Witcha Chimphlee\textsuperscript{1}, Abdul Hanan Abdullah\textsuperscript{2}, Mohd Noor Md Sap\textsuperscript{2} and Surat Srinoy\textsuperscript{1}

\textbf{Abstract} — The goal of intrusion detection is to discover unauthorized use of computer systems. New intrusion types, of which detection systems are unaware, are the most difficult to detect. The amount of available network audit data instances is usually large; human labeling is tedious, time-consuming, and expensive. Traditional anomaly detection algorithms require a set of purely normal data from which they train their model. In this paper we propose an intrusion detection method that combines Fuzzy Clustering and Genetic Algorithms. Clustering-based intrusion detection algorithm which trains on unlabeled data in order to detect new intrusions. Fuzzy c-Means allow objects to belong to several clusters simultaneously, with different degrees of membership. Genetic Algorithms (GA) to the problem of selection of optimized feature subsets to reduce the error caused by using land-selected features. Our method is able to detect many different types of intrusions, while maintaining a low false positive rate. We used data set from 1999 KDD intrusion detection contest.

\textbf{Keywords} — Anomaly detection, Unsupervised clustering, Genetic Algorithms, Fuzzy c-means

I. INTRODUCTION

As defined in [1], intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. It is also defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. The goal for handle intrusion detection problem is to classify patterns of the system behavior in two categories (normal and abnormal). The amount of available network audit data instances is usually large; human labeling is tedious, time-consuming, and expensive. However, not all the data collected are useful or informative. There are two main intrusion detection systems. Anomaly intrusion detection system is based on the profiles of normal behaviors of users or applications and checks whether the system is being used in a different manner [2]. The second one is called misuse intrusion detection system which collects attack signatures, compares a behavior with these attack signatures, and signals intrusion when there is a match. Generally, there are four categories of attacks [3]. They are: DoS (denial-of-service) R2L (Remote to Local) U2R (User to Root), and PROBING.

IDS can be classified based on the functional characteristics of detection methods as knowledge based intrusion detection and behavior based intrusion detection [4]. The task of an intrusion detection system is to protect a computer system by detecting and diagnosing attempted breaches of the integrity of the system. Anomaly detection still faces many challenges, where one of the most important is the relatively high rate of false alarms (false positives). The problem of capturing a complex normality makes the high rate of false positives intrinsic to anomaly detection except for simple problems [5].

The paper is structured as follows. In section II presents the related works. Section III describes a brief introduction of genetic algorithms. Explains fuzzy clustering in section IV. Section V explains about experimental design. Section VI evaluates our intrusion detection model through experiments. Finally, section VII presents our conclusion and some discussion.

II. RELATED WORKS

Feature selection has been traditionally used in data mining applications as part of the data cleaning and/or pre-processing step where the actual extraction and learning of knowledge or patterns is done after a suitable set of features is extracted. Many approaches have been proposed which include statistical [6], machine learning [7], data mining [8] and immunological inspired techniques [9]. Alves et al [10] presents a classification-rule discovery algorithm integrating artificial immune systems (AIS) and fuzzy systems.

III. THE GENETIC ALGORITHMS

Genetic algorithms (GAs) were formally introduced in the 1970s by John Holland [11]. In particular, genetic algorithms are problem-solving techniques based on the principles of biological evolution, natural selection and genetic recombina-
tion. Potential solutions to the problem to be solved are encoded as sequences of bits, characters or numbers. The unit of encoding is called a gene, and the encoded sequence is called a chromosome. Each chromosome represents one possible solution to the problem or a rule in a classification. GAs is able to select subsets of various sizes in order to determine the optimum combination and number of inputs to network. This allows reducing the computational expense on the training system with near optimal results still reachable. Research [12] has show that GA is one of the most efficient of all feature selection methods for dealing with feature sets containing large (>100) numbers of features.

![Flowchart of GA](image)

Fig.1. A typical GA program flow.

The algorithm begins with a population of chromosomes generated either randomly or from some set of known specimens, and cycles through the three steps, namely evaluation, selection, and reproduction. A chromosome contains the information about the solution to a problem, which it represents. Typically, it can be encoded using a binary string as follows [13]:

- **Chromosome 1**: 11011100100110110
- **Chromosome 2**: 11011100011110110

Fig.1 shows a general outline of the algorithm. In the first step, that is, the evaluation step, each string is evaluated according to a given performance criterion known as fitness function, and assigned a fitness score. In the next step, the selection step, a decision is made according to the fitness score assigned to each individual to decide which individuals are permitted to produce offspring and with what probability. Finally, the reproduction step involves the creation of offspring chromosomes by two generators, namely cross-over and mutation. This is the most important part of the genetic algorithms as these genetic operators have an impact on the performance of GAs. Fig. 2 illustrated using a pair of chromosomes encoded as two binary strings, where the cross-site is denoted by “|” in each chromosome.

![Chromosomes and Offspring](image)

Fig.2. Pair of chromosomes encoded

For a chromosome encoded as a binary string, genes are randomly selected to undergo mutation operation, where ‘1’ is changed into ‘0’ or vice versa, in Fig.3.

- **Original offspring 1**: 11011110000111110
- **Mutated offspring 1**: 1101110000111110
- **Original offspring 2**: 1101100100110110
- **Mutated offspring 2**: 1101101100110110

Fig.3. Mutation operation

In which a bit value of 1 in the chromosome representation means that the corresponding feature is included in the specified subset, and a value of 0 indicates that the corresponding feature is not included in the subset.

The fitness of a feature subset is measured by the test accuracy (or cross-validation accuracy of the classifier learned using the feature subset) and any other criteria of interest [14].

**IV. FUZZY C-MEANS CLUSTERING**

Fuzzy c-means (FCM) algorithm, also known as fuzzy ISODATA, was introduced by Bezdek [16] as extension to Dunn’s [17] algorithm to generate fuzzy sets for every observed feature. FCM is an iterative algorithm to find cluster centers (centroids) that minimize a dissimilarity function. Rather that partitioning the data into a collection of distinct sets by fuzzy partitioning, the membership matrix (U) is randomly initialized according to Equation 1.

\[
\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j = 1, ..., n.
\]

The dissimilarity function which is used in FCM in given Equation

\[
J(U, c_1, c_2, ..., c_n) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^2
\]

where:
- \(u_{ij}\) is between 0 and 1;
- \(c_i\) is the centroid of cluster \(i\);
- \(d_{ij}\) is the Euclidian distance between \(i_{th}\) centroid \((c_i)\) and \(j_{th}\) data point;
- \(m \in [1, \infty)\) is a weighting exponent.

To reach a minimum of dissimilarity function there are two conditions. These are given in (3) and (4).
Algorithm 1. Fuzzy c-means

Step 1: Randomly initialize the membership matrix (U) that has constraints in Equation 1.

Step 2: Calculate centroids (c) by using Equation 3.

Step 3: Compute dissimilarity between centroids and data points using Equation 2. Stop if its improvement over previous iteration is below a threshold.

Step 4: Compute a new U using Equation 4 go to step 2.

Fig. 4. Fuzzy c-means clustering.

By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the “right” location within a data set. FCM does not ensure that it converges to an optimal solution. Because of cluster centers are initializing using U that randomly initialized. (Equation 3)

Performance depends on initial centroids. For a robust approach there are two ways which is described below [13].

1.) Using an algorithm to determine all of the centroids. (for example: arithmetic means of all data points)
2.) Run FCM several times each starting with different initial centroids.

A collection of fuzzy sets, called fuzzy space, defines the fuzzy linguistic values or fuzzy classes. A sample fuzzy space of five membership function is shown in Figure 5.

\[
\begin{align*}
 c_i &= \frac{\sum_{j=1}^{n} u_{ij} m^x_j}{\sum_{j=1}^{n} u_{ij} m} \\
 u_{ij} &= \frac{1}{\sum_{k=1}^{c} \left( d_{ij} / d_{ij} \right)^{m-1}}
\end{align*}
\]

(3)

(4)

Detailed algorithm of fuzzy c-means proposed by Bezdek in 1973 [18] (Fig.4).

V. EXPERIMENTAL DESIGN

In our method have the three steps (Figure 6). First step is cleaning for handle missing and incomplete data. Second step using genetic algorithms for select the best attribute and feature selection and the last step for clustering group of data using fuzzy c-means.

The pre-processor module performs the following tasks:
1. Identifies the attributes and their value.
2. Convert categorical to numerical data.
3. Data Normalization
4. Performs redundancy check and handle about null value.
5. Initializes all the necessary parameters such as the length of a chromosome, population size

We ran our experiments on a system with a 1.5 GHz Pentium IV processor and 512 MB DDR RAM running Windows XP. All the preprocessing was done using MATLAB®. In practice, the number of classes is not always known beforehand. There is no general theoretical solution to finding the optimal number of clusters for any given data set. We choose k = 5 for the study. We will compare five classifiers which have been also used in detecting these four types of attacks.

VII. EXPERIMENTAL SET UP AND RESULTS

In this experiment, we use the KDD Cup 1999 intrusion detection contest [19]. This database includes a wide variety of intrusions simulated in a military network environment that is a common benchmark for evaluation of intrusion detection techniques. The data set has 41 attributes for each connection record plus one class label. There are 24 attack types, but we treat all of them as an attack group. The nominal attributes are converted into linear discrete values (integers). After eliminating labels, the data set is described as a matrix X, which has N rows and m = 41 columns (attributes). There are md = 8 discrete-value attributes and m_c = 33 continuous-value attributes.

A. Data preprocessing

It was necessary to ensure though, that the reduced dataset was as representative of the original set as possible. Table 1
shows the dataset after balanced among category for attack distribution over modified the normal and other attack categories appear in [20]. Preprocessing consisted of two steps. The first step involved mapping symbolic-valued attributes to numeric-valued attributes and the second step implemented non-zero numerical features.

Table 1: Dataset for attack distribution

<table>
<thead>
<tr>
<th>Attack Category</th>
<th>% Occurrence</th>
<th>Number of records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>31.64</td>
<td>5,763</td>
</tr>
<tr>
<td>PROBE</td>
<td>11.88</td>
<td>2,164</td>
</tr>
<tr>
<td>DoS</td>
<td>19.38</td>
<td>3,530</td>
</tr>
<tr>
<td>U2R</td>
<td>0.38</td>
<td>70</td>
</tr>
<tr>
<td>R2L</td>
<td>36.72</td>
<td>6,689</td>
</tr>
<tr>
<td>100%</td>
<td></td>
<td>18,216</td>
</tr>
</tbody>
</table>

B. Feature extraction system

A feature selection method is proposed to select a subset of variables in data that preserves as much information present in the complete data as possible. The feature subset selection problem refers the task of identifying and selecting a useful subset of attributes to be used to represent patterns from a larger set of often mutually redundant, possibly irrelevant, attributes with different associated measurement costs and/or risks [21]. The feature selection process is very important which selects the informative features for used classification process. The benefits and affects of feature subset selection include.

C. Intrusion Detection with Clustering

Based on the practical assumption that normal instances dominate attack instances, our simple self-labeling heuristic for unsupervised intrusion detection appear in [22]. Anomaly detection amounts to training models for normal traffic behavior and then classifying as intrusions any network behavior that significantly deviates from the known normal patterns and to construct a set of clusters based on training data to classify test data instances.

The data set before using feature selection has 41 attributes and after use genetic algorithms for identifies subset of features; we got 8 attributes as follows duration, service, flag, srv count, error rate, dst host srv count, dst_host_diff_svr_rate and dst_host_srv_reerror_rate.

VII. CONCLUSIONS

In this paper we apply genetic algorithm methods with data reduction and to identify subset of features for network security and using fuzzy c-means to intrusion detection to avoid a hard definition between normal class and certain intrusion class. Features selection methods aim at selecting a small or prespecified number of features leading to the best possible performance of the entire classifier. The task of identifying and selecting a useful subset of features to be used to represent patterns from a larger set of often mutually redundant or even irrelevant features. Therefore, the main goal of feature subset selection is to reduce the number of features used in classification while maintaining acceptable classification accuracy.

Intrusion detection model is a composite model that needs various theories and techniques. One or two models can hardly offer satisfying results. We plan to evaluation results and to apply other theories and techniques in intrusion detection in our future work.

REFERENCES


