World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:8, 2007

Unsupervised Clustering Methods for Identifying Rare Events in Anomaly Detection

Witcha Chimphlee, Abdul Hanan Abdullah, Mohd Noor Md Sap, Siriporn Chimphlee, and Surat Srinoy

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf

***Abstract*—**It is important problems to increase the detection rates and reduce false positive rates in Intrusion Detection System (IDS). Although preventative techniques such as access control and authentication attempt to prevent intruders, these can fail, and as a second line of defence, intrusion detection has been introduced. Rare events are events that occur very infrequently, detection of rare events is a common problem in many domains. In this paper we propose an intrusion detection method that combines Rough set and Fuzzy Clustering. Rough set has to decrease the amount of data and get rid of redundancy. Fuzzy *c*-means clustering allow objects to belong to several clusters simultaneously, with different degrees of membership. Our approach allows us to recognize not only known attacks but also to detect suspicious activity that may be the result of a new, unknown attack. The experimental results on Knowledge Discovery and Data Mining-(KDDCup 1999) Dataset show that the method is efficient and practical for intrusion detection systems.

***Keywords*—**Network and security, intrusion detection, fuzzy *c*-means, rough set.

I. Introduction

SECURITY techniques such as authentication and access control have been developed to achieve the objective of computer security – namely to prevent unauthorized intruders from accessing and manipulating information. The security administrator is now faced with the problem of selecting suitable IDS for his/her particular computer system.

Rapid expansion of computer network throughout the world has made security a crucial issue in a computing environment. Anomalies pattern sometimes exist within tiny or rare classes of similar anomalies. Anomaly-based network intrusion detection is a complex process. The challenge is thus important to identify "rare events" records in data set.

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". Anomaly Intrusion Detection Systems (IDSs) aim at distinguishing an abnormal activity from an ordinary one.

Intrusion detection is a critical component of secure information systems. Many approaches have been proposed which include statistical [2], machine learning [3], data mining [4] and immunological inspired techniques [5]. Identification of suspicious activities before they have an impact; to perform situational assessment and to respond in a more timely and effective manner. Events that may not be actual security violations but those that do not fit in the normal usage profile of a user may be termed as suspicious events. Monitoring suspicious activities may help in finding a possible intrusion.

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 [6]. 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.

It is often impossible to analyze the vast amount of whole data, but one has to focus the analysis on an important portion of the data such as using some criteria, only the classes of interest can be selected for analysis or processing while the rest is rejected. This paper suggests the use rough set as a dimensionality reduction technique to avoid this information loss.

The theory of rough sets has been specially designed to handle data imperfections same as in fuzzy logic. Rough sets remove superfluous information by examining attribute dependencies. It deals with inconsistencies, uncertainty and incompleteness by imposing an upper and a lower approximation to set membership. Rough sets estimates the relevance of an attribute by using attribute dependencies regarding a given decision class. It achieves attribute set covering by imposing a discernibility relation.

In this paper we apply the combination of rough sets and fuzzy *c*-means to intrusion detection to avoid a hard definition between normal class and certain intrusion class. Clustering-based intrusion detection algorithm**,** unsupervised anomaly

W. Chimphlee is with PhD Student, Faculty of Computer Science and Information Systems, University Technology Malaysia, 81310 UTM Skudai, Johor, Malaysia (phone: +60197931495; fax: +607-5565044; e-mail: witcha_chi@dusit.ac.th).

Prof. Dr. Abdul Hanan Abdullah is Dean of Faculty of Computer Science and Information Systems, University Technology Malaysia (e-mail: hanan@fsksm.utm.my).

Asst. Dr. Mohd Noor Md Sap is head of Information Department, Faculty of Computer Science and Information Systems, University Technology Malaysia (e-mail: mohdnoor@fsksm.utm.my).

S. Chimphlee is with PhD Student, Faculty of Computer Science and Information Systems, University Technology Malaysia (e-mail: siriporn_chi@ dusit.ac.th).

S. Srinoy is with lecturer, Faculty of Science and Technology, Suan Dusit Rajabhat University, Thailand (e-mail: surat_sri@ dusit.ac.th).

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf

detection, which trains on unlabeled data in order to detect new intrusions.

The rest of this paper is organized as follows. In section II, we discuss the related works; introduce rough set in section III; explains clustering in section IV; in section V, to illustrate experimental design; and evaluate our intrusion detection model through experiments in section VI; and in section VII ends the paper with a conclusion and some discussion.

II. Related Works

Most intrusion occurs via network using the network protocols to attack their targets. Twycross [7] proposed a new paradigm in immunology, Danger Theory, to be applied in developing an intrusion detection system. Alves et al [8] presents a classification-rule discovery algorithm integrating artificial immune systems (AIS) and fuzzy systems. For example, during a certain intrusion, a hacker follows fixed steps to achieve his intention, first sets up a connection between a source IP address to a target IP, and sends data to attack the target [6]. Generally, there are four categories of attacks [9]. They are:

- DoS (denial-of-service), for example ping-of-death, teardrop, smurf, SYN flood, and the like.
- R2L : unauthorized access from a remote machine, for example guessing password,
- U2R : unauthorized access to local super user (root) privileges, for example, various "buffer overflow" attacks,
- PROBING: surveillance and other probing, for example, port-scan, ping-sweep, etc.

Some of the attacks (such as DoS, and PROBING) may use hundreds of network packets or connections, while on the other hand attacks like U2R and R2L typically use only one or a few connections.[10]

Attack connection and normal connections have their special feature values and flags in the connection head, and package contents can be used as signatures for normal determination and intrusion detection. Intrusions belonging to the same intrusion category have identical or similar attack principles and intrusion techniques. Therefore they have identical or similar attack connections and are significantly different from normal connections [6].

III. Rough Sets

The rough sets theory has been developed for knowledge discovery in databases and experimental data sets. An attribute-oriented rough sets technique reduces the computational complexity of learning processes and eliminates the unimportant or irrelevant attributes so that the knowledge discovery in database or in experimental data sets can be efficiently learned.

A rough set is an approximation of a vague concept by a pair of precise concepts, called lower and upper approximations (which are a classification of the domain of interest into disjoint categories) in Fig 1.

The classification formally represents knowledge about the problem domain. Objects belonging to the same category characterized by the same attributes (or features) are not distinguishable [11]. Let $I = (U,A)$ be an information system, where U is a non-empty set of finite of objects (the *universe*). A is a non-empty finite set of *attributes* such that $a : U \to V_a$

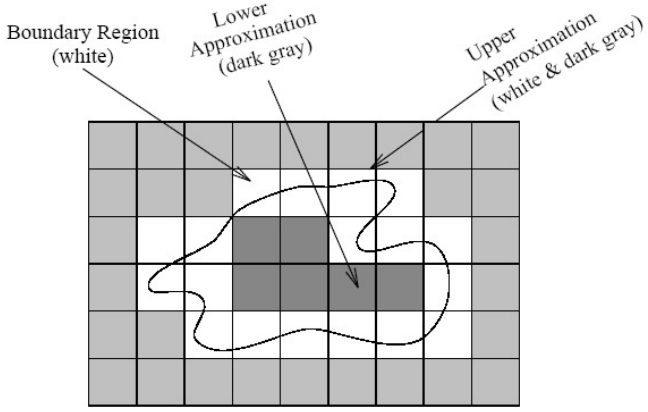


Fig. 1 Depiction of a rough set

For every $a \in A$; V_a is the value set for attribute a. In a decision system, $A = \{C \cup D\}$ where C is the set of conditional attributes and D is the set of decision attributes.

With any $P \subseteq A$ there is an associated equivalence relation *IND (P)*:

$$IND(P) = \{(x, y) \in U^2 \mid \forall_a \in P_a(x) = a(y)\} \quad (1)$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The partition of U, generated by *IND(P)* is denoted U/P and can be calculated as follows:

$$U / P = \otimes\{a \in P : U / IND(\{a\})\}, \text{ where} \quad (2)$$

$$A \otimes B = \{X \cap Y : \forall X \in A, \forall Y \in B, X \cap Y \neq \varnothing\} \quad (3)$$

To illustrate the operation of Rough Set Attribute Reduction (RSAR), an example dataset is presented as in Table I.

TABLE I
Example Dataset

Object	Duration	Src_bytes	Dst_bytes	Attacks
1	0	508	124	No
2	0	508	250	Yes
3	0	509	13400	No
4	0	510	460	Yes
5	0	511	750	Yes
6	0	509	50	No

Information can be incomplete, inconsistent, uncertain, or all three. We adopted the rough set algorithm for data cleaning as proposed by Sarjon and Mohd Noor [12]. To use rough sets by the equivalence up to discernibility, this attribute reduction will have to be minimal with respect to content of information.

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf

A. Reduct Algorithms

The algorithms supplied by the Rosetta library support two types of discernibility: i) Full: In this case the reducts are extracted relative to the system as a whole. ii) Object: This kind of discernibility extract reducts relative to a single object. We are interested in two reduct extraction algorithms supplied by Rosetta library, Johnson's Algorithm and Genetic Algorithm [23].

Johnson's algorithm implements a variation of a simple greedy search algorithm. This algorithm extracts a single reduct.

The **Genetic Algorithm** described by ∅ hrm and Viterbo in [19] is used to find minimal hitting sets.

IV. Clustering

Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition),numerical taxonomy (in biology ecology), typology (in social sciences) and partition (in graph theory) [13].

The aim of cluster analysis is the classification of network connections, or objects, according to similarities among them, and organizing objects into groups. A cluster is a group of objects that are more similar to each other than to objects in other clusters. Similarity is often defined by means of distance based upon the length from a data vector to some prototypical object of the cluster.

The data are typically observations of some phenomenon. Each object consists of *m* measured variables, grouped into an *m*-dimensional column vector $x_i = \{x_{i1}, x_{i2,\ldots}, x_{in}\}$. A set of *n* objects is denoted by $U = x_i = \{x_{i1}, x_{i2,\ldots}, x_p\}$

$$X = [x_{ij}] = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{p,1} & x_{p,2} & \cdots & x_{p,n} \end{pmatrix}$$

A. k-Means Clustering

The *k*-Means clustering is a classical clustering algorithm. After an initial random assignment of example to *k* clusters, the centers of clusters are computed and the examples are assigned to the clusters with the closest centers. The process is repeated until the cluster centers do not significantly change. Once the cluster assignment is fixed, the mean distance of an example to cluster centers is used as the score. Using the *k*-means clustering algorithm, different clusters were specified and generated for each output class.

There are two problems that are inherent to *k*-Means clustering algorithms. The first is determining the initial partition and the second is determining the optimal number of clusters. In Fig. 2 depicted *k*-means algorithms.

Algorithm 1. *k*-means

Step 1: Choose *k* cluster centers to coincide with *k* randomly-chosen patterns or *k* randomly defined points inside the hyper volumn containing the pattern set.

Step 2: Assign each pattern to the closest cluster center.

Step 3: Recomputed the cluster centers using the current luster memberships.

Step 4: If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new luster centers, or minimal decrease in squared error

Fig. 2 *k*-Means Clustering [22]

B. Fuzzy c-Means (FCM) Clustering

Fuzzy *c*-means (FCM) algorithm, also known as fuzzy ISODATA, was introduced by Bezdek [15] as extension to Dunn's [14] algorithm to generate fuzzy sets for every observed feature. The Fuzzy *c*-means clustering algorithm is based on the minimization of an objective function called *c-means functional.*

Fuzzy clustering methods allow for uncertainty in the cluster assignments. 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 4.

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall_j = 1, \ldots, n. \tag{4}$$

The dissimilarity function which is used in FCM in given Equation:

$$J(U, c_1, c_2, \ldots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m {d_{ij}}^2 \tag{5}$$

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 (6) and (7).

$$c_i = \frac{\sum_{j=1}^{n} {u_{ij}}^m x_j}{\sum_{j=1}^{n} {u_{ij}}^m} \tag{6}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \tag{7}$$

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf

Detailed algorithm of fuzzy *c*-means proposed by Bezdek in 1973 [23]. This algorithm determines the following steps in Fig. 3.

Algorithm 2. Fuzzy *c*-means
Step 1: Randomly initialize the membership matrix (U) that has constraints in Equation 4.
Step 2: Calculate centroids (c_i) by using Equation 6.
Step 3: Compute dissimilarity between centroids and data points using Equation 5. Stop if its improvement over previous iteration is below a threshold.
Step 4: Compute a new U using Equation 7 go to step 2.

Fig. 3 Fuzzy *c*-Means Clustering [23]

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 (centroids) are initialize using U that randomly initialized. (Equation 6).

Performance depends on initial centroids. For a robust approach there are two ways which is described below [14].

1.) Using an algorithm to determine all of the centroids. (for example: arithmetic means of all data points)

2.) Run FCM several time each starting with different initial centroids.

V. Experimental Design

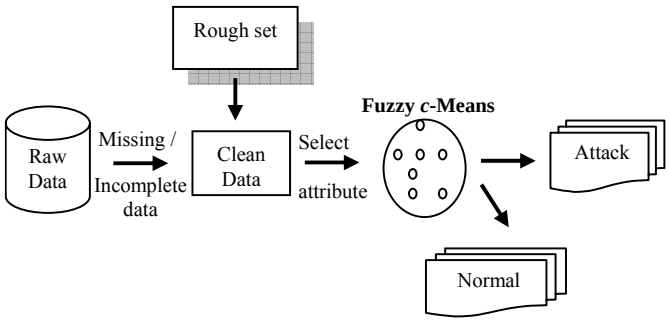


Fig. 4 Step for cluster

In our method have three steps (Fig. 4). First step for cleaning (handle missing and incomplete data) using rough set. Second step for select the best attribute or feature selection and the last step for clustering group of data using fuzzy *c*-means.

VI. Experimental Setup and Results

In this experiment, we use a standard dataset the raw data used by the KDD Cup 1999 intrusion detection contest [16]. 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. Test data use filename "corrected.gz" contains a total of 38 training attack types. It consists of approximately 300,000 data instances, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusion and is labeled normal or a certain attack type. The 41 features can be divided into three groups; the first group is the basic feature of individual TCP connections, the second group is the content feature within a connection suggested by domain knowledge, and the third group is the traffic feature computed using a two-second time window. The distribution of attacks in the KDD Cup dataset is extremely unbalanced. Some attacks are represented with only a few examples, e.g. the *phf* and *ftp_write* attacks, whereas the *smurf* and *neptune* attacks cover millions of records. In general, the distribution of attacks is dominated by probes and denial-of-service attacks; the most interesting and dangerous attacks, such as compromises, are grossly under represented [17].

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. A data set of size N is processed. 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 m_d=8 discrete-value attributes and m_c = 33 continuous value attributes.

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®. MATLAB's Fuzzy Logic Toolbox [18] was used for Fuzzy *c*-means clustering, whereas rough set operations were done in ROSETTA [19]. ROSETTA is a software toolkit capable of performing all the operations for data processing and classification.

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.

A. Data Preprocessing

A considerable amount of data preprocessing had to be undertaken before we could do any of our modeling experiments. It was necessary to ensure though, that the reduced dataset was as representative of the original set as possible. The test dataset that previously began with more than 300,000 records was reduced to approximately 18,000 records. Table II shows the dataset after balanced among category for attack distribution over modified the normal and other attack categories. 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. We reduce the dimensionality of this data set (by using rough set and use Jonhson's algorithms) from 42 to 10 attributes are duration, service, src_bytes, dst_byte, count, srv_count, serror_rate, dst_host_srv_count, dst_host_diff_srv_rate, and dst_host_same_src_port_rate.

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf

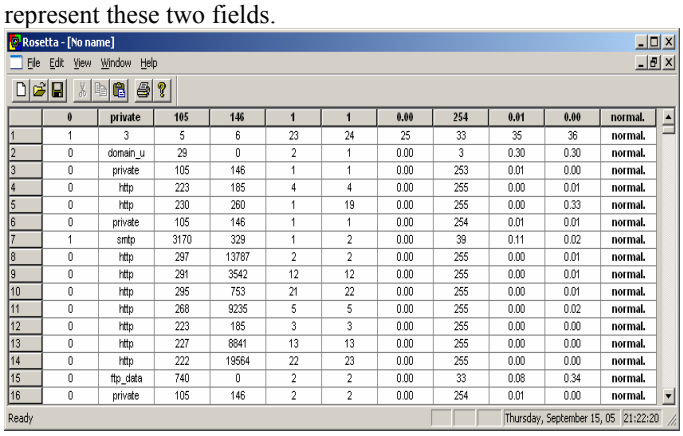
TABLE II
DATASET FOR ATTACK DISTRIBUTION

Attack Class	% Occurrence	Number of records
normal	31.64	5,763
probe	11.88	2,164
dos	19.38	3,531
u2r	0.38	70
r2l	36.72	6,689
	100.00	**18,217**

B. Features Selection

Feature extraction creates new features by irreversibly transforming the original features such that the created features contain most useful information for the target concept. Feature selection only removes the features that are unnecessary or unimportant to the target concept and the remaining features are kept intact. Feature selection is a process to find the optimal subset of features that satisfy certain criteria. The aim of feature selection is to remove unnecessary features to the target concept. If two features are functional dependent, one of them could be removed without the loss of predication accuracy.

When we process volumes of data, it is necessary to reduce the large number of features to a smaller set of features. There are 42 fields in each data record and it is hard to determine which fields are useful or which fields are trivial. Jin et al [6] suggest correlation coefficients between fields by using SPSS. They propose that if the correlation coefficients of fields *i* and *j*, *R(i,j)*, is larger than 0.8, then there is a strong correlation between fields *i* and *j*, and will select either one of them to represent these two fields.



	0	private	105	146	1	1	0.00	254	0.01	0.00	normal.
1	1	3	5	6	23	24	25	33	35	36	normal.
2	0	domain_u	29	0	2	1	0.00	3	0.30	0.30	normal.
3	0	private	105	146	1	1	0.00	253	0.01	0.00	normal.
4	0	http	223	185	4	4	0.00	255	0.00	0.01	normal.
5	0	http	230	260	1	19	0.00	255	0.00	0.33	normal.
6	0	private	105	146	1	1	0.00	254	0.01	0.01	normal.
7	1	smtp	3170	329	1	2	0.00	39	0.11	0.02	normal.
8	0	http	297	13787	2	2	0.00	255	0.00	0.01	normal.
9	0	http	291	3542	12	12	0.00	255	0.00	0.01	normal.
10	0	http	295	753	21	22	0.00	255	0.00	0.01	normal.
11	0	http	268	9235	5	5	0.00	255	0.00	0.02	normal.
12	0	http	223	185	3	3	0.00	255	0.00	0.00	normal.
13	0	http	227	8841	13	13	0.00	255	0.00	0.00	normal.
14	0	http	222	19564	22	23	0.00	255	0.00	0.00	normal.
15	0	ftp_data	740	0	2	2	0.00	33	0.08	0.34	normal.
16	0	private	105	146	2	2	0.00	254	0.01	0.00	normal.

Ready Thursday, September 15, 05 21:22:20

Fig. 5 Dataset after use Rosetta

C. Performance Measure

Standard measures for evaluating IDSs include *detection rate*, *false alarm rate*, trade-off between *detection rate* and *false alarm rate* [20], *performance* (Processing speed + propagation + reaction), and *Fault Tolerance* (resistance to attacks, recovery, and subversion). Detection rate is computed as the ratio between the *number of correctly detected attacks* and the total *number of attacks*, while false alarm (false positive) rate is computed as the ratio between the numbers of normal connections that are incorrectly misclassified as attacks [21]. These are good indicators of performance, since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications are made in the process.

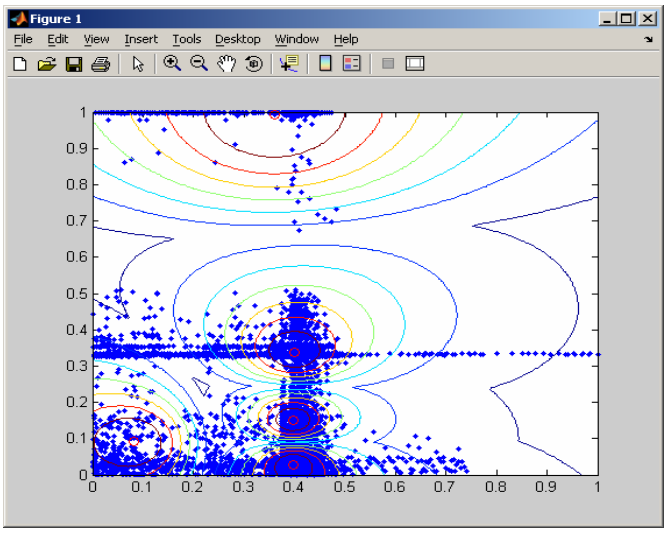


Fig. 6 Five Clusters of data after used FCM

Standard metrics that were developed for evaluating network intrusions usually correspond to detection rate as well as false alarm rate.

1. True Positives (TP), the number of malicious executables correctly classified as malicious;
2. True Negatives (TN), the number of benign programs correctly classified as benign;
3. False Positives (FP), the number of benign programs falsely classified as malicious,
4. False Negative (FN), the number of malicious executables falsely classified as benign.

May be defines as follows:

$$\text{Detection Rate (DTR)} = \frac{TP}{(TP + FN)} \qquad (8)$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{(TN + FP)} \qquad (9)$$

$$\textit{Overall Accuracy } (OA) = \frac{TP + TN}{TP + TN + FP + FN} \qquad (10)$$

Another valuable tool for evaluating an anomaly detection schemeis the receiver operating characteristic (ROC) curve, which is the plot of the detection rate against the false alarm rate. The nearer the ROC curve of a scheme is to the upper-left corner, the better the performance of the scheme is.

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.

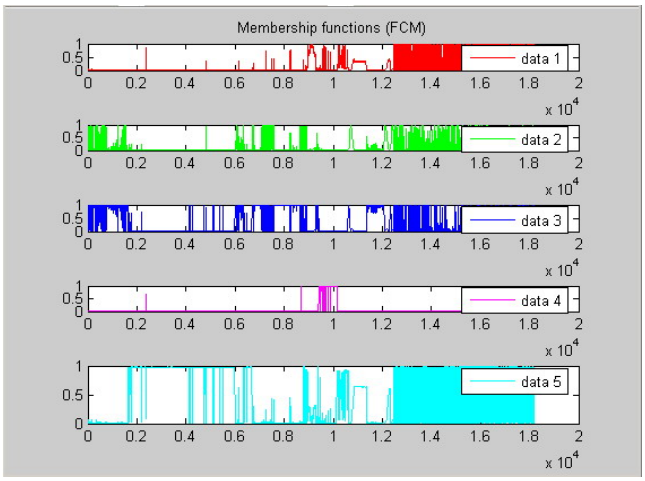


Fig. 7 Membership functions of each cluster

After experiment we got group of data by using membership functions that illustrate in Fig. 6 and Fig. 7 respectively.

VII. Conclusion

Anomaly-based network intrusion detection is a complex process. In this paper we apply fuzzy *c*-means clustering methods to intrusion detection to avoid a hard definition between normal class and certain intrusion class. We introduce the current status of intrusion detection systems (IDS) and data mining research, and present some possible data mining based ways for solving problems. Rough set based methods with data reduction for network security are discussed.

Experiment results show that rough set method is suitable and promising for network security. The advantage of using fuzzy logic is that it allows one to represent concepts that could be considered to be in more than one category (or from another point of view – it allows representation of overlapping categories). The problem of detecting rare events has been variously called deviation detection, outlier analysis, anomaly detection, exception mining, and mining rare classes.

These results are very promising since detection accuracy at low false-positive rates is extremely important in IDS. Intrusion detection model is a compositive model that needs various theories and techniques. One or two models can hardly offer satisfying results. We plan to apply other theories and techniques in intrusion detection in our future work.

References

[1] R. Bace and P. Mell, "*Intrusion Detection Systems*", NIST Special Publications SP 800. 31 November 2001.

[2] D. Denning, "*An intrusion-detection model*," In IEEE computer society symposium on research in security and privacy, 1986, pp. 118-131.

[3] T. Lane, "*Machine Learning techniques for the computer security*", PhD thesis, Purdue University, 2000.

[4] W. Lee and S. Stolfo, "Data mining approaches for intrusion detection," *Proceedings of the 7th USENIX security symposium,* , 1998.

[5] D. Dagupta and F. Gonzalez, "*An immunity-based technique to characterize intrusions in computer networks*", *IEEE Transactions on Evolutionary Computation*, Vol. 6, June 2002, pp.28- 291.

[6] H. Jin, J. Sun, H. Chen, and Z. Han, "*A Fuzzy Data Mining based Intrusion Detection System*", Proceedings of 10thInternational Workshop on future Trends in Distributed Computing Systems (FTDCS04) IEEE Computer Society, Suzhou, China, May 26-28, 2004, pp. 191-197.

[7] J. Twycross, "*Immune Systems, Danger Theory and Intrusion Detection*", presented at the AISB 2004. Symposium on Immune System and Cognition, Leeds, U.K., March 2004.

[8] R.T. Alves, M.R.B.S. Delgado, H.S. Lopes, A.A. Freitas, "*An artificial immune system for fuzzy-rule induction in data mining*", Lecture Notes in Computer Science, Berlin: Springer Verlag, v. 3242, 2004, pp. 1011-1020.

[9] W. Lee, S. Stolfo, and K. Mok, "*A data mining framework for building intrusion detection models*", Proceedings of the 1999 IEEE Symposium on Security and Privacy, May 1999, pp.120-132.

[10] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar, "*A comparative study of anomaly detection schemes in network intrusion detection*", *In* SIAM International Conference on Data Mining, 2003.

[11] R. Jensen and Q. Shen, "*Rough and fuzzy sets for dimensionality reduction*", Proceedings of the 2001 UK Workshop on Computational Intelligence, 2001, pp. 69-74.

[12] D. Sarjon and Mohd Noor Md Sap, "*Association Rules using Rough Set and Association Rule Methods*", Proceedings of 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI-02), Tokyo, Japan, August 18-22, 2002, pp. 238-243.

[13] S. Theodoridis, K. Koutroubas, "*Pattern recognition",* Academic Press, 1999.

[14] S. Albayrak, Fatih Amasyali, *Fuzzy c-means clustering on Medical Diagnostic Systems*, International XII. Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN 2003.

[15] J. Bezkek, "*Pattern Recognition with Fuzzy Objective Function Algorithms*", *Plenum Press*, USA, 1981.

[16] KDD data set, 1999; http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[17] P. Laskov, K. Rieck, C. Schäfer, K.R. Müller, "*Visualization of anomaly detection using prediction sensitivity*", Proceeding of Sicherheit, April 2005, pp. 197-208.

[18] MathWorks, Statistical Toolbox for User's Guide, MathWorks, 2001.

[19] A. ∅ hrm, "*ROSETTA Technical Reference Manual*", Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2000.

[20] W. Chimphlee, Abdul Hanan Abdullah, Mohd Noor Md Sap and S. Chimphlee, "*Unsupervised Anomaly Detection with Unlabeled Data using Clustering*", Proc. Int. Conf. on ICT- Mercu Buana ICT2005. pp. 42-49.

[21] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar, "A comparative study of anomaly detection schemes in network intrusion detection". In SIAM; International Conference on Data Mining, 2003.

[22] Zhengxin Chen, Data Mining and Uncertain Reasoning - An Integrated Approach, Wiley, 2001.

[23] Fernando Godínez, Dieter Hutter, Raul Monroy "Attribute Reduction for Effective Intrusion Detection". AWIC 2004: 74-83.

[24] W. Chimphlee, Abdul Hanan Abdullah, Mohd Noor Md Sap, S.Chimphlee, and S. Srinoy, Unsupervised Anomaly Detection without Prior Knowledge Using Clustering, International workshop on information Technology 2005 (IAIT2005), 25-26 November 2005., Thailand.

[25] W. Chimphlee, Mohd Noor Md Sap, Abdul Hanan Abdullah, and S. Chimphlee, Semi-Supervised Learning to Identify Suspicious Activity for Anomaly Detection, 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS2005), 13-16 December 2005, Singapore.

Open Science Index, Computer and Information Engineering Vol:1, No:8, 2007 publications.waset.org/2474/pdf