

Detection and Prevention of intrusion attacks using Fuzzy Clustering

M. Masthan¹, R. Ravi²

Research Scholar, Dept. of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India¹

Professor & Head, Dept. of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India²

Abstract— Detection of intrusion attacks is a major problem in the privacy and security of any networks, and the fuzzy clustering approach has been carried out in many fields today. Therefore researches on fuzzy clustering technique is very much essential to real time applications as well. This paper detailed about the intrusion detection using C-Means fuzzy likelihood approach. The proposed approach attains over 80 percentage of average detection rate, which has been revealed from the experiments carried out with the KDD Cup 1999 set. Other Parametric analysis are detailed in the results section. An outclass performance is achieved greater than RIPPER method.

Index Terms— Intrusion Detection, Fuzzy Clustering, Fuzzy Possibility C-Means Algorithm, RIPPER.

I. INTRODUCTION

Intrusions pose a critical protection chance for the stability and the safety of data in a network environment. An intrusion is defined as any set of moves that try to compromise the integrity, confidentiality or availability of a resource, it consists of trying to destabilize the community, gaining unauthorized get entry to documents with privileges, or mishandling and misusing of software. The intrusion detection is to automatically scan community hobby and discover intrusion assaults. A fundamental assumptions of intrusion detection are customers and software activities are observable, and more importantly, regular activities and intrusion sports have awesome behaviors. The aim of intrusion detection is to display network sports automatically, hit upon malicious assaults and to establish a proper architecture of the computer community safety with cooperation of different intrusion prevention tactics consisting of the firewall. An intrusion detection device (IDS) monitors atomizing answer in a variety of areas. D restricts person get entry to (behavior) to the computer system by applying positive guidelines. Misuse detection and anomaly detection are procedures of the intrusion detection device. Fuzzy set theory was proposed via Zaiden, In his conventional paper, Prof. Zaiden said that, "A fuzzy set A in X is characterized with the aid of a membership characteristic $f_A(x)$ which buddies with every factor in X an actual quantity in the interval $[0, 1]$, with the cost of $f_A(x)$ at x representing the 'grade of membership' of x in A". The bushy set concept intends to seize the vagueness of meaning in words utilized in maximum of our everyday communications to explain ideas, gadgets, events, phenomena, or statements themselves. A

linguistic variable is a variable that takes phrases utilized in our natural language as its values, with every phrase interpreted as a fuzzy set. As we know, many traditional clustering algorithms, consisting of the distinguished k-approach set of rules, produce a clustering structure wherein every item is assigned to one cluster in an unequivocal way. Therefore, the individual clusters are separated by using sharp obstacles. In exercise, such limitations are often now not very herbal or maybe counterintuitive. Alternatively, the boundary of unmarried clusters and the transition between exceptional clusters are generally "clean". This is the main motivation underlying fuzzy extensions to clustering algorithms. In fuzzy clustering, an item can also belong to one-of-a-kind clusters at the equal time, at the least to some extent, and the degree to which it belongs to a selected cluster is expressed in phrases of a fuzzy club. The membership capabilities of the distinct clusters is commonly assumed to form a partition of harmony. Fuzzy clustering has proved to be extremely beneficial in practice and is now automatically implemented also outside the bushy community. In this paper, a fuzzy clustering technique is proposed to intrusion detection, our technique employs the fuzzy opportunity C-approach algorithm and RIPPER (Repeated Incremental Pruning to provide error reduction, like different rule gaining knowledge of systems, is usually used for classification troubles.)To categorize every new program conduct into both normal and intrusive class. The relaxation of this paper is organized as follows: segment 2 gives some method have been used to clear up this issues; in section three, we introduces fuzzy clustering principle and algorithm; the experimental settings, results, and some conclusions are offered in segment four.

II. RELATED WORK

There are several approaches for solving intrusion detection problems. Various data mining techniques have been applied to intrusion detection because it has the advantage of discovering useful knowledge that describes user's or program's behavior from large audit data sets. Statistics [07, 08, 13], artificial neural network [10], HMM (Hidden Markov Model) [03, 04, 12], and rule learning [01, 09] detection scheme are some of the data mining techniques widely used for anomaly and misuse detections. Statistics is the most widely used technique in intrusion detection, neural networks are trained to detect intrusion systems, and Natural immune system is another proposed method to deal with the



intrusion detection problem in distributed manner. Distributed positive and negative detectors are used to distinguish self and non-self-behaviors. Agents exploiting the learning power of genetic programming are evaluated with their performance and agents having highest performance are chosen to detect intrusions. Rough set theory already be used in the data selection and preparation phase, e.g., for modeling vague data in terms of fuzzy sets, to “condense” several crisp observations into a single fuzzy one, the data to be analyzed thus becomes fuzzy, the problem of analyzing fuzzy data can be approached in at least two principally different ways. First, standard methods of data analysis can be extended in a rather generic way by means of an extension principle, the second, often more sophisticated approach is based on embedding the data into more complex mathematical spaces, such as fuzzy metric spaces, and to carry out data analysis in these spaces. If fuzzy methods are not used in the data preparation phase, they can still be employed in a later stage in order to analyze the original data. Thus, it is not the data to be analyzed that is fuzzy, but rather the methods used for analyzing the data. A rough set theory method is applied to learn a rule set modeling the normal behaviors with small size of training data sets and improved detection accuracy [02]. It is extended by another method based on plan recognition for predicting intrusion intentions [05]. Evolving fuzzy classifiers have been studied for possible application to the intrusion detection problem [01, 06].

III. UNITS

The fuzzy c-means (FCM) algorithm was introduced by J. C. Bedeck [2]. The idea of FCM is using the weights that minimize the total weighted mean-square error:

$$J(w_{qk}, \mathbf{z}^{(k)}) = \sum_{(k=1,K)} \sum_{(k=1,K)} (w_{qk}) \|\mathbf{x}^{(q)} - \mathbf{z}^{(k)}\|^2$$

$$\sum_{(k=1,K)} (w_{qk}) = 1 \text{ for each } q$$

$$w_{qk} = (1/(D_{qk})^2)^{1/(p-1)} / \sum_{(k=1,K)} (1/(D_{qk})^2)^{1/(p-1)}, p > 1$$

The FCM allows each feature vector to belong to every cluster with a fuzzy truth value (between 0 and 1), which is computed using Equation (4). The algorithm assigns a feature vector to a cluster according to the maximum weight of the feature vector over all clusters.

IV. THE FCM ALGORITHM

Based on the original FCM algorithm [9], we added Steps 7, 8, 9 and modified Step 2. The pseudocode of our FCM algorithm follows.

// K is initial number of clusters, Imax is the iteration of fuzzy

// c-means, p is for the weight

Input initial number of clusters K, Imax, p

-----step 1: -----

//initialize weights of prototype

for k = 0 to K-1

for q = 0 to Q-1

w[q,k] = random();

-----step 2: -----

//standardize the initial weight over K

for q = 0 to Q-1

sum = 0.0;

for k = 0 to K-1

sum = sum + w[q,k];

for k = 0 to K-1

w[q,k] = w[q,k] /sum;

// starting fuzzy c-means loop

I = 0

-----step 3: -----

// standardize cluster weights over Q

for k = 0 to K-1

min = 99999.0; max = 0.0;

for q = 0 to Q-1

if (w[q,k] > max)

max = w[q,k];

if (w[q,k] < min)

min = w[q,k];

sum = 0.0

for q = 0 to Q-1

sum = sum + (w[q,k] - min) / (max - min);

for q = 0 to Q-1

w[q,k] = w[q,k]/sum;

-----step 4: -----

// compute new prototype center

for k = 0 to K-1

for n = 0 to N-1

sum = 0.0;

for q = 0 to Q-1

sum = sum + w[q,k] x[n,q];

z[n,k] = sum;

-----step 5: -----

// compute new weight

for q = 0 to Q-1

sum = 0.0

for k = 0 to K-1

D[q,k] = 0.0;

for n = 0 to N-1

D[q,k] = D[q,k] + (x[n,q] - z[n,k])²

sum = sum + (1/(1 + D[q,k]))^{1/(p-1)};

for k = 0 to K-1



$$W[q,k] = (1/(1 + D[q,k]))1/(p-1) / \text{sum};$$

-----step 6: -----

I = I + 1

If I < I_{max}

Goto step 3;

// end of fuzzy c-means loop

-----step 7: -----

// assign feature vector according the max weight

for q = 0 to Q-1

maxWeight = 0.0;

for k = 0 to K-1

if maxWeight < weight[q,k];

maxWeight = weight[q,k];

kmax = k;

cluster[q] = k;

-----step 8: -----

// eliminate clusters with no feature vectors

eliminate(0); /* call the process of eliminating clusters contains

less than or equal to the number

passed to it.

Here we only pass 0 for this

algorithm. */

-----step 9: -----

// compute arithmetic center of clusters

// calculate sigma and Xie_Beni value

for k = 1 to K do

fuzzyweights(); /* Calculate fuzzy weight (Eqn. 4)

variance(); /* Get variance (mean-square error) of each cluster (Eqn. 9) */

Xie-Beni(); /* Compute modified XB (Eqn. 8) */

V. THE FUZZY C-MEANS RESULTS

We used the following data sets to run both algorithms:

The iris data set [7] is Anderson's [1] 150 feature vectors of iris species. The data is labeled as K = 3 that represents 3 subspecies (Sestosa, Versicolor and Virginica). Each feature vector has 4 features. The given sample contains 50 labeled feature vectors from each class.

The test13 data set consists of 2 natural classes. There are 13 two-dimensional feature vectors. Figure 1 shows the feature vectors.

The Wisconsin breast cancer data set (WCBDB) [7] consists of 200 randomly selected from more than 500 breast cancer vectors of the University of Wisconsin Medical School. Each feature vector has 30 features in [0, 1]. The vectors are labeled for two classes. One label is attached to 121 vectors while the other is attached to 79 vectors.

The geological data set [7] is labeled for K = 2 classes. Each of the Q = 70 feature vectors has N = 4 features. The data labels are estimates by humans that give 35 to each class. These were assigned by humans providing their best guesses.

VI. DIFFERENT COMBINATION OF PARAMETERS

We tried different combinations of parameters in Table 1 and Table 2. The different combinations of Kinit, iterations and p (Equation (4)) result in different outputs.

In Table 1, 2, 3 and 4, we run the FCM without eliminating empty clusters. That is, if iteration I = 300, we did not interrupt the process until it reaches to the 300 iterations, then we deleted the empty clusters and then compute the modified XB.

TABLE 1: ITERATIONS AND PROCESS OF CLUSTERS

Iterations	P = 2		P = 3		P = 4	
	Clusters	XB	Clusters	XB	Clusters	XB
100	56, 57, 37	2.19E-30	48, 93, 9	1.8E-31	53, 68, 25, 4	0.39E-32
200	56, 94	2.365	57, 93	2.225	57, 93	2.190
300	56, 94	2.365	57, 93	2.225	57, 93	2.190

Table 1 presents the results on iris data set. We standardized the components of feature vectors between 0 and 1, set Kinit = 150, assigned the feature vectors to the prototypes, and varied the value p and iteration number, p = 2, 3, 4, 5 and iteration = 100, 200, 300. The best result we get in Table 1 is when p = 2 and iteration >= 200, the modified XB = 2.365, there are two clusters, one has 56 feature vectors, the other has 94.

TABLE 2: ITERATIONS OF MODIFIED CLUSTERS

Table 2 shows the results on iris data set under the same condition as in Table 1 except Kinit = 50. The best result so far was when p = 2 and iteration I >= 100, the modified XB = 2.385, there are two clusters, one has 56 feature vectors, the other has 94.

Iterations	P = 2		P = 3		P = 4	
	Clusters	XB	Clusters	XB	Clusters	XB
100	56, 94	2.385	51, 93, 6	9.46E-32	53, 93, 4	1.36E-31
200	56, 94	2.385	57, 93	2.225	57, 93	2.190
300	56, 94	2.385	57, 93	2.225	57, 93	2.190

VII. INITIALIZING THE PROTOTYPES

To study the difference between initializing prototypes randomly and using the feature vectors, we run the program at same conditions but initialized the prototypes using feature vectors in Table 3 and initialized randomly in Table 4. The

resulting clusters were affected by the initial prototype centers.

Table 3 shows the results on iris data set. We standardized the feature vectors into [0, 1], initialized prototypes using the first Kinit feature vectors, and run the program by fixing $p = 2$ and varying the iteration number of fuzzy clustering and Kinit, where iteration number $I = 100$ and 200 , Kinit = 150, 120, 90, 60, 30, 10 and 5. The results were very similar, after 200 iteration, the first cluster contains 56 feature vectors and the second contains 94 feature vectors.

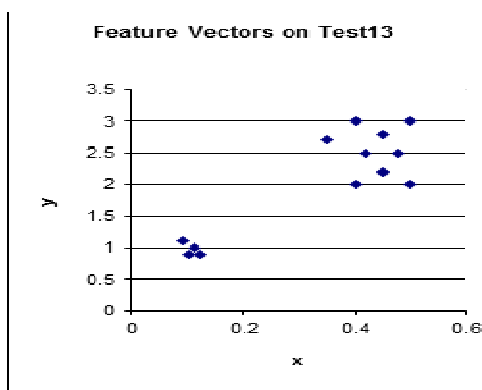


Fig 1. Feature Vector on Test 13

TABLE III: INITIALIZATION OF K W.R.T I

K_{init}	$I = 100$	
	Clusters	XB
150	56, 57, 37	2.19E-30
120	56, 94	2.355
90	56, 94	2.355
60	56, 94	2.355
30	56, 94	2.405
10	56, 94	2.355
5	50, 94, 6	1.15E-31

TABLE IV: APPROXIMATION OF K W.R.T I

K_{init}	$I = 100$		$I = 200$	
	Clusters	XB	Clusters	XB
150	59, 91	2.175	59, 91	2.175
120	54, 90, 3, 3	3.05E-32	59, 90, 1	3.63E-31
90	60, 90	2.12	60, 90	2.17
60	59, 91	2.17	59, 91	2.17
30	20, 90, 2, 7, 31	2.08E-32	52, 90, 8	5.07E-31
10	58, 73, 19	2.87E-31	58, 92	2.195
5	56, 94	2.355	56, 94	2.355

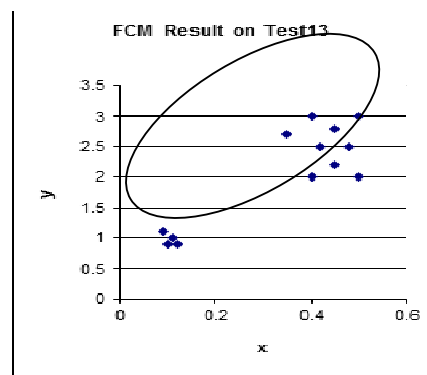


Fig 2. Clustering Results on Test 13

VIII. CONCLUSION

A digitized result is through Table 1 and Table 2. The results show that the average detection rate is to 91.0%, and the false positive rate ranges from 0.50% to 1.80%, the total performance evaluation is outperforms the RIPPER method. Besides the results show from the digitized, we should tackle additional two issues in practice. The first is choosing the number of clusters, the second is defining the categorical attributes. In this paper we assumed that the number of clusters is fixed and the categorical attributes are pre-defined by experts. As we know, defining categorical attribute is a difficult and important task in cluster analysis, different definitions of the attributes might lead to undesirable clustering results. Intrusion detection is an important and active area of research, although different algorithms such as statistics, machine learning, neural network, decision trees and so on have been explored in intrusion detection, various research groups have suggested methods that look promising on at least one set of data. In order to choose from among these different methods, we need good comparisons between them on a variety of data, such comparisons are not easy. In order to evaluate the framework, more datasets are required, the future work is to do experiments on other datasets.

REFERENCES

- [1]E. Anderson, "The iris of the Gaspé peninsula" Bulletin American Iris Society, Vol. 59, 2-5, 1935.
- [2]J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York 1981.
- [3]J. C. Bezdek, etc. Convergence Theory for Fuzzy c-Means: Counterexamples and Repairs, IEEE Trans. Syst., September/October 1987.
- [4]Maria Colmenares & Olaf Wolkenhauer, "An Introduction into Fuzzy Clustering",



- <http://www.csc.umist.ac.uk/computing/clustering.htm>, July 1998, last update 03 July, 2000
- [5] Marti Hearst, K-Means Clustering, UCB SIMS, Fall 1998.
 - [6] Uri Kroszynski and Jianjun Zhou, Fuzzy Clustering Principles, Methods and Examples, IKS, December 1998.
 - [7] Carl G. Looney A Fuzzy Clustering and Fuzzy Merging Algorithm, CS791q Class Notes, <http://www.cs.unr.edu/~looney/>.
 - [8] Carl G. Looney Pattern Recognition Using Neural Networks, Oxford University Press, N.Y., 1997.
 - [9] Carl G. Looney "Chapter 5. Fuzzy Clustering and Merging", CS791q Class Notes, <http://www.cs.unr.edu/~looney/>.
 - [10] Ramze Rezaee M, Lelieveldt B P F and Reiber J H C, A New Cluster Validity Index for the Fuzzy c-mean, Pattern Recognition Letters, (Netherlands) Mar 1998.
 - [11] M.J.Sabin, Convergence and Consistency of Fuzzy c-means /ISODATA Algorithms, IEEE Trans. Pattern Anal. Machine Intel. September 1987.