Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 13, No. 3, September 2013, pp. 178-185
http://dx.doi.org/10.5391/IJFIS.2013.13.3.178

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

A Systematic Approach to Improve Fuzzy C-Mean Method based on Genetic Algorithm

Xiao-Yun Ye and Myung-Mook Han
Department of Computer Science, Gachon University, Seongnam, Korea

Abstract

As computer technology continues to develop, computer networks are now widely used. As a result, there are many new intrusion types appearing and information security is becoming increasingly important. Although there are many kinds of intrusion detection systems deployed to protect our modern networks, we are constantly hearing reports of hackers causing major disruptions. Since existing technologies all have some disadvantages, we utilize algorithms, such as the fuzzy C-means (FCM) and the support vector machine (SVM) algorithms to improve these technologies. Using these two algorithms alone has some disadvantages leading to a low classification accuracy rate. In the case of FCM, self-adaptability is weak, and the algorithm is sensitive to the initial value, vulnerable to the impact of noise and isolated points, and can easily converge to local extrema among other defects. These weaknesses may yield an unsatisfactory detection result with a low detection rate. We use a genetic algorithm (GA) to help resolve these problems. Our experimental results show that the combined GA and FCM algorithm's accuracy rate is approximately 30% higher than that of the standard FCM thereby demonstrating that our approach is substantially more effective.

Keywords: Principal component analysis, Fuzzy C-means, Genetic algorithm

1. Introduction

Intrusion detection is an important technology in computer defense; it can detect anomalous activity through feature matching. Portnoy [1] was the first to propose intrusion detection techniques based on cluster analysis using Euclidean distance; after identification, classification can be used to detect anomalies. However, there are some problems with these methods, such as weak self-adaptability, sensitivity to the initial value, vulnerability to the impact of noise and isolated points, and the easy of converge to local extrema among other defects. This may yield an unsatisfactory detection result with a low detection rate. Genetic algorithms (GAs) are used to simulate the natural mechanisms of a biological evolutionary randomized search algorithm. They are more suitable that processing with traditional search methods to solve complex optimization problems. GAs have strong global search capabilities, but they are weak for local search. Improving upon the traditional fuzzy C-means (FCM) algorithm, we use a GA to optimize the processing results. First, the data is divided into many subsets of data. Second, we use the FCM clustering algorithm to obtain the clustering center of each subset of data. Then we use a GA to optimize these cluster centers. As a result we can get an approximation of the global optimal cluster centers. Finally, we use this result as the

Received: Jun. 21, 2013
Revised : Sep. 10, 2013
Accepted: Sep. 12, 2013

Correspondence to: Myung-Mook Han
(mmhan@gachon.ac.kr)

initial value of the FCM algorithm. In this way, we combine the two algorithms for processing. Not only can we overcome the FCM algorithm's sensitivity to the initial value and the problem of converging to a local optimal solution, we also implement a GA, which can find a better global solution of the problem. Our experimental results show that the combined GA and FCM algorithm's accuracy rate is approximately 30% higher than that of the standard FCM, demonstrating that our approach is substantially more effective.

The remainder of this paper is organized as follows. We begin by introducing some existing technology, such as FCM, GA, and principal component analysis (PCA). Then, in Section 3, we establish a system for diagnosing anomalies. In this system, we combine GA and FCM to process the data in order to obtain better results than those from using the technologies alone. In Section 4, we test our system, and compare the results obtained. Finally, we present the conclusion in Section 5.

2. Related Work

2.1 Data

KDDCUP1999 [2] is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99 the Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

At present, the main attack method is denial of service (DoS) attacks, probe attacks, remote to local (R2L) attacks, and user to root (U2R) attacks.

2.1.1 Denial of Service Attacks

A DoS attack is one in which a single user occupies a large number of shared resources [3], so that the system has few or no remaining resources available for other users. DoS can be used to attack the domain name servers, routers, and other network operation services. It can be used to reduce the availability resources of the CPU, disk space, printers, and modems. Typical attack methods of DoS are via SYN Flooding, Ping Flooding, Echl, Land, Rwhod, Smurf, and Ping of Death.

2.1.2 Probe Attacks

Probe attacks scan the computer network or NDS server to obtain a valid IP address, active ports, and the host operating system's weakness [4]. Hackers can use this information to attack the target host. Probe attacks can be divided into two types: the hidden type and the public type. Common features collected by all probe attacks include the IP address, vulnerable port numbers, and the type of operating system in use. However, the hidden probe is generally lower speed but receives more concentrated information than the public type. Probe attacks typically include the use of SATAN, Saint, NTScan, Nessus, SAFEsuite, and COPS.

2.1.3 Remote to Local Attacks

In R2L attacks, hackers can get local host machine access on target host machines, and can obtain or modify the host machine's data [5]. R2L is also a remote attack method. The remote access process includes: (1) collecting the host machine's information and analyzing the system's possible weaknesses. (2) building a simulation environment and performing a simulated attack to test the target machine's possible response, (3) using suitable software to scan the host machine, and (4) attacking the host machine.

2.1.4 User to Root Attacks

A U2R attack is one in which a local user obtains Unix's advanced user permissions or Windows' administrator permissions [6]. Utilizing buffer overflow is a typical method of U2R attack.

To counter these attacks, we need a method to classify the given network data effectively. The method we propose in this paper achieves a higher accuracy than the original method on which it is based.

Experimental data is obtained by using four major attacks of KDDCUP1999 data distributed uniformly throughout 50,000 data samples

2.2 Principal Component Analysis

When the dimensionality of a data set is high, we use the PCA method to convert the set to one of lower dimensionality. PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables

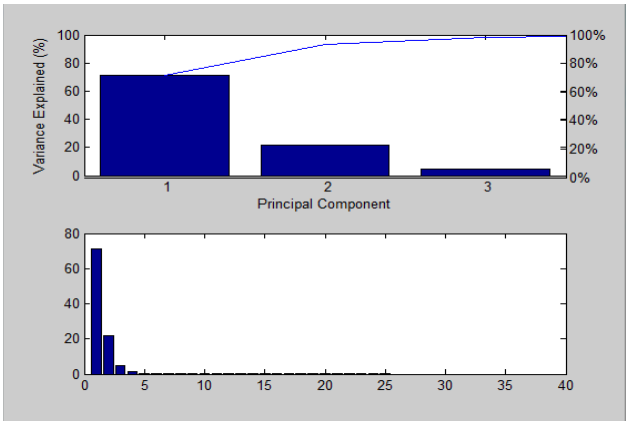

Figure 1. Principal component analysis processing result.

called principal components [7]. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

Figure 1 shows, after PCA analysis, the main characteristic factors of the dataset, as well as their sum. It can be seen that the first three principal components of the sum of the contribution rate has reached 98%, so there are three main factors in the data set. Thus we can restrict ourselves to the analysis of only these three components.

2.3 Fuzzy C-Means

The FCM algorithm [8,9] attempts to partition a finite collection of n elements $X = \{x_1, \ldots x_n\}$ into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centers $C = \{c_1, \ldots c_c\}$ and a partition matrix:

$$W = w_{i,j} \in [0,1], \; i = 1, \ldots, n, \; j = 1, \ldots, c \qquad (1)$$

where each element $u_{i,j}$ indicates the degree to which the element x_i belongs to cluster c_j. As in the k-means algorithm, the FCM aims to minimize an objective function. The standard function [10,11] is:

$$w_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)} \right)^{2/(m-1)}} \qquad (2)$$

which differs from the k-means objective function by the addition of the membership values $u_{i,j}$ and the fuzzifier m. The fuzzifier m determines the level of cluster fuzziness. A large m value results in smaller memberships $u_{i,j}$ and hence, in fuzzier clusters. The limit of m is 1 in formula (2), the memberships $u_{i,j}$ converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, m is commonly set to 2. In the basic FCM algorithm, we are given n data points $\{x_1, \ldots x_n\}$ to be clustered, a number of c clusters with $\{c_1, \ldots c_c\}$ the center of the clusters, and m, the level of cluster fuzziness.

In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster may be in a cluster to a lesser degree than points in the center of the cluster. An overview and comparison of different fuzzy clustering algorithms is available in [12].

Any point x has a set of coefficients giving the degree of being in the kth cluster, $w_k(x)$. With FCMs, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$c_k = \frac{\sum_x w_k(x)\, x}{w_k(x)} \qquad (3)$$

The degree of belonging, $w_k(x)$ is related inversely to the distance from x to the cluster center as calculated on the previous pass. It also depends on a parameter m that controls how much weight is given to the closest center. The FCMs algorithm is very similar to the k-means algorithm [13].

2.4 Genetic Algorithm

In the computer science field of artificial intelligence, a GA is a search heuristic that mimics the process of natural evolution [14-16]. This heuristic is routinely used to generate useful solutions to optimization and search problems. GAs belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The flowchart of a GA is shown in Figure 2.

Simple generational GA procedure:

1. Choose the initial population of individuals.

2. Evaluate the fitness of each individual in that population.

3. Repeat on this generation until termination (time limit, sufficient fitness achieved, etc.):

(1) Select the best-fit individuals for reproduction.

(2) Breed new individuals through crossover and mutation operations to give birth to offspring.

(3) Evaluate the individual fitness of new individuals.

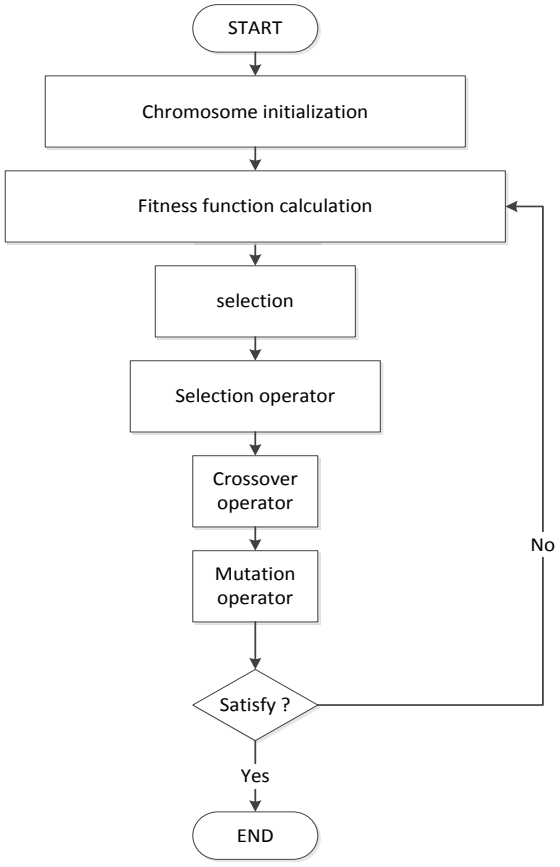

Figure 2. Flowchart of genetic algorithm.

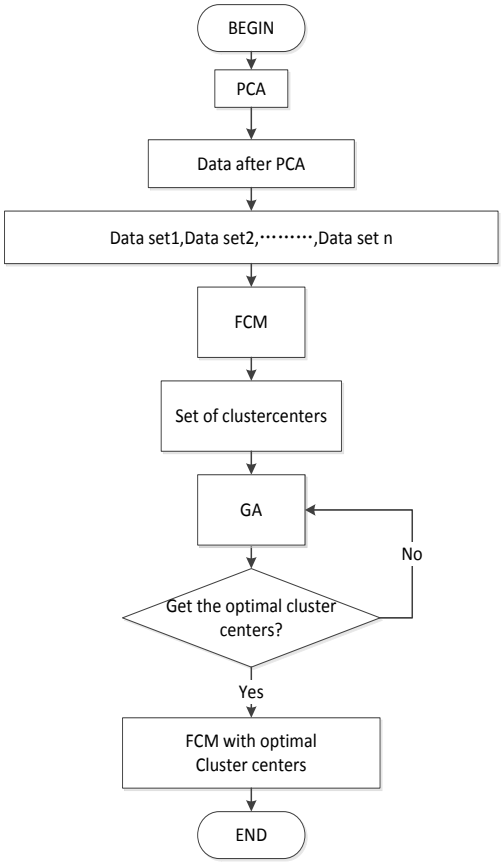

Figure 3. Flowchart of genetic algorithm (GA) and fuzzy C-means (FCM). PCA, principal component analysis.

(4) Replace least-fit population with new individuals.

3. Improve FCM Algorithm based on GA

The flowchart of GA+FCM is shown in Figure 3. First, we input the KDDCUP1999 data. We then use PCA to process the data. The data pre-process method is shown in Section 2.2.

Next, the data is divided into many subsets of data. For example, in the data set used for Figure 4, there are 5000 data points, which are divided into 10 groups of equal size. We use the FCM clustering algorithm to determine the clustering center of each subset of data. We will show the GA processing in Section 2.4.

Figure 4 shows one group of 10 data sets. We use FCM to obtain its cluster centers, and the small red circles in the centers are the positions of cluster centers.

3.1 GA Process

3.1.1 Code

An individual in a population is a cluster center [17]. To avoid the complexity of the encoding and to improve efficiency, we link the center of each group clustering. This helps to shorten chromosome length, and to improve the convergence speed and global optimum searching capability.

For example, the clustering center V= $[v_1,v,\ldots,v_c]^T$ after coding is $\{v_{11}, v_{12},\ldots, v_{1k},\ldots, v_{c1},\ldots, v_{ck}\}$, the v_{ij} is jth component of v_i.

3.1.2 Fitness function

For the FCM algorithm, the optimal clustering results correspond to the minimum value of the objective function. Therefore, the individual fitness function can make use of the objective function of FCM algorithm for its definition.

The fitness function is:

$$f = \frac{1}{1+k} \tag{4}$$

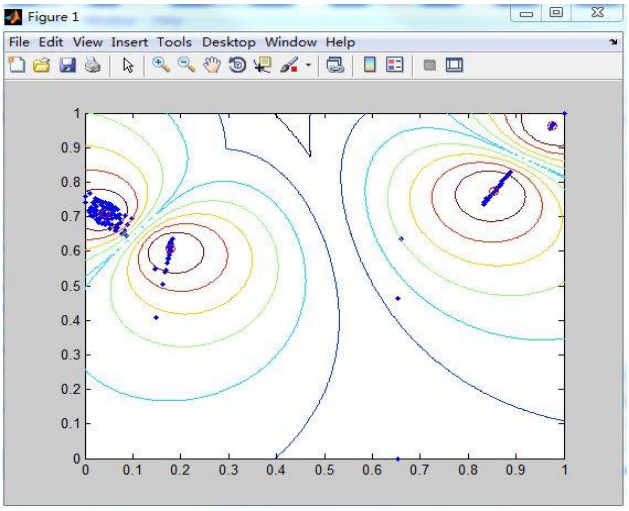

Figure 4. One group of the 10 datasets after fuzzy C-means processing.

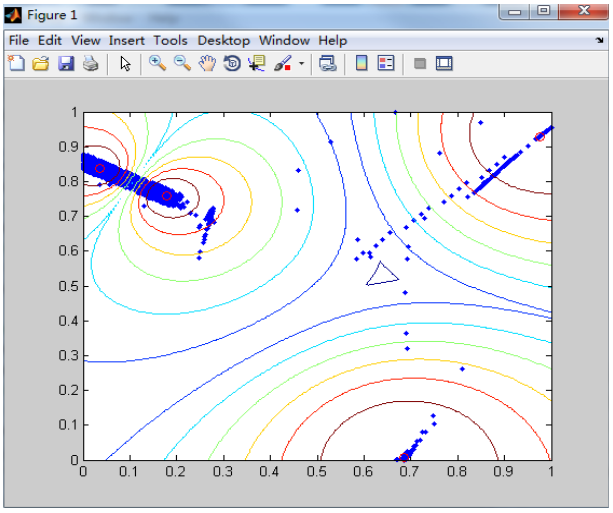

Figure 5. Optimal clustering.

where k is the square of the distance of each data instance to the class.

3.1.3 Selection

Selection is the most important part of GA. In population evolutionary processes, the best individuals of the population are retained to avoid crossover and mutations, so that their desirable characteristics can be passed on to the next generation directly. The worst individuals do not participate in crossover, but they will have mutations with a larger probability than normal individuals. We then use the roulette method to choose the individuals, and we calculate the fitness function's probability distribution for the populations. We choose individuals according to the probability distribution for crossover and mutation processing. In this way, we can improve the population's average fitness performance. The selection probability function is defined via:

$$\text{Rate}\left(V^i\right) = \frac{f\left(i\right)}{\sum_{j=1}^{m} f\left(i\right)} \tag{5}$$

where $f\left(i\right)$ is the individual V_i's fitness value.

3.1.4 Crossover

We set (V^a,V^b) as crossover parents [18]. They crossover between i and $i+1$, and we can use the method to get the next generation $V^{a'}$ and $V^{b'}$. The cross position i is an integer.

$$v^{a'} = v_i^a + v_i^b \tag{6}$$

$$v^{b'} = v_i^b + v_i^a \tag{7}$$

3.1.5 Mutation

We set V^a as a mutation individual, and we use a mutation operation on i, to obtain the next generation $V^{a'}$ on i position

$$v_i^{a'} = \beta v_i^a + \left(1-\beta\right) v_i^b \tag{8}$$

<1> Set the cluster class equal to 4, the population to 40, the crossover rate to 0.3, the mutation rate to 0.5, and maximum number of iterations to 100.
<2> Calculate individual's fitness through the fitness function.
<3> GA operations: selection, crossover, and mutation.
<4> Calculate the children's fitness rate, and put them into their parents. Delete the individuals with low fitness rate.
<5> If the maximum iteration number is reached then return the individual with the largest fitness rate.
<6> End GA

We can get the optimal cluster centers through GA, and we use these cluster centers for the FCM algorithm's initial value. We then use FCM to process the data. The result of using the optimal clustering with FCM as indicated in the above algorithm is shown in Figure 5 for a data set of 5000 points.

4. Experimental Result and Analysis

First, we extract 50000 data points from KDDCUP1999. We divided the data into 10 groups, as in Figure 6. The y-axis in the figure is data set number, and the x-axis in the figure is the size of the data used. The colors indicate attacks as the right side in Figure 6. The number of data points used is 5000 per group.

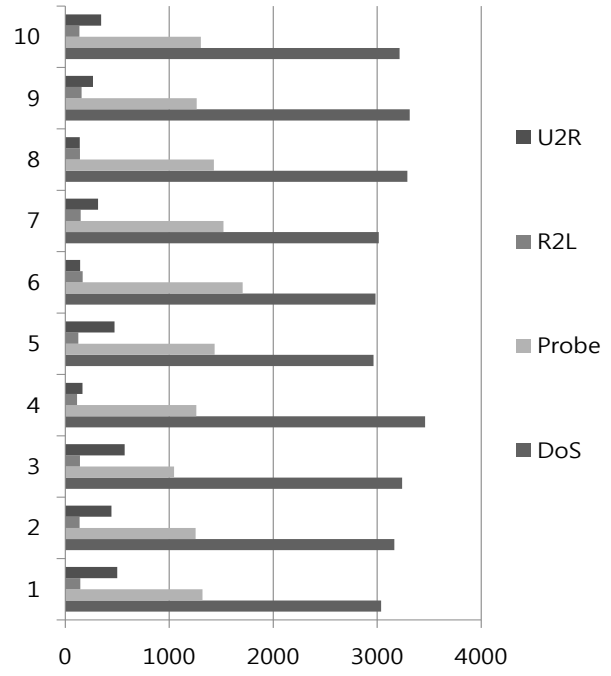

Figure 6. 50000 data points in 10 groups.

Using the algorithm from Section 3, we arrive at the conclusions in Table 1.

DR (detection rate) is the ratio of true intrusion instances detected by the system to the total number of intrusion instances in the data set.

OR (omission rate) is the ratio of the intrusion instances incorrectly identified by the system as non-intrusion instances to the total number of intrusion instances in the data set.

FR (false alarm) rate is ratio of the non-intrusion instances incorrectly identified by the system as intrusions to the total number of non-intrusion instances in the data set.

These rates are all defined with respect to the KDDCUP1999 data.

To facilitate analysis of the experimental results, we defined the function f(x)=DR(x)-OM(x)-FR(x) [19,20]. Our results are shown in Table 1.

The average DR shown in Table 1 is the average over the average detection rates of the 10 sets of data. Upon using the function f(x)=DR(x)-OM(x)-FR(x), we can determine the actual detection rate. The "FCM only" column gives the actual detection rate with only the FCM used to process the same data.

5. Conclusion

This paper uses a combination of a GA and the FCM method in intrusion detection. We solve the problems of the GA's weakness for local search and the FCM's weakness for global search. Consequently, we not only overcome the FCM algorithm's sensitivity to the initial value and its tendency to yield local optimal solutions, but we can also utilize the GA's primary strength of finding good global solutions. Through our experimental results, we find that the "GA+FCM" combined algorithm is better than the "FCM only" algorithm as measured by detection rate. However, the detection rate of our proposed method is not higher than that of other methods. Hence, more work is required to improve on this method. In future research, we will select other useful data mining methods to deal with these data and continue to reduce the redundancy in the data, and we will continue to learn about intrusion detection methods and find a more effective method to get a higher correct rate of intrusion detection.

Table 1. Experimental results

Anomaly	Average DR(%)	OM (%)	FR(%)	GA + FCM(%)	FCM only (%)
DoS	88.45	2.63	6.49	79.33	58.18
U2R	79.31	3.35	4.75	71.21	48.33
Probing	82.37	0.47	9.74	72.16	47.75
R2L	71.59	4.68	10.37	56.64	43.67

DR, detection rate; OM, omission rate; FR, false alarm; GA, genetic algorithm; FCM, fuzzy C-means.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

[1] L. Protnoy, E, Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, pp. 1-14, 2001.

[2] KDD cup 1999 data, Available http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3] Y. Liu, Z. Wang, and Y. Feng, "DoS intrusion detection based on incremental learning with support vector machines," *Computer Engineering*, vol. 32, no. 4, pp. 179-186, 2006.

[4] M. P. O'Mahony, N. J. Hurley, and G. C. Silvestre, "Recommender systems: attack types and strategies," in *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, PA, 2005, pp. 334-339.

[5] M. Sabhnani and G. Serpen, "KDD feature set complaint heuristic rules for R2L attack detection," in *Security and Management*, H. R. Arabnia and Y Mun, Eds. Lasvegas: CSREA Press, 2003, pp. 310-316.

[6] M. Birker-Robaczewska, C. Boukhadra, R. Studer, C. Mueller, C. Binkert, and O. Nayler, "The expression of urotensin II receptor (U2R) is up-regulated by interferon-gamma," *Journal of Receptors and Signal Transduction*, vol. 23, no. 4, pp. 289-305, 2003.

[7] I. T. Jolliffe. *Principal Component Analysis,* 2nd ed., New York: Springer, 2002.

[8] J. H. Min and F. C. H. Rhee, "An interval type-2 fuzzy PCM algorithm for pattern recognition," *Journal of The Korean Institute of Intelligent Systems*, vol. 19, no. 1, pp. 102-107, 2009.

[9] B. Y. Kang and D. W. Kim, "VS-FCM: validity-guided spatial fuzzy C-means clustering for image segmentation," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, no. 1, pp. 89-93, Mar. 2010. http://dx.doi.org/10.5391/IJFIS.2010.10.1.089

[10] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1983.

[11] D. H. Park, S. Ryu, P. H. Jeong, and S. K. Lee, "Application of similarity measure for fuzzy C-means clustering to power system management," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 8, no. 1, pp. 18-23,Mar. 2008.

[12] R. Nock and F. Nielsen, "On weighting clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1223-1235, Aug. 2006. http://dx.doi.org/10.1109/TPAMI.2006.168

[13] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the 11th International Conference on Information and Knowledge Management*, New York, 2002, pp. 600-607. http://dx.doi.org/10.1145/584792.584890

[14] H. T. Kim, J. H. Lee, and C. W. Ahn, "A recommender system based on interactive evolutionary computation with data grouping," *Procedia Computer Science*, vol. 3, pp. 611-616, 2011. http://dx.doi.org/10.1016/j.procs.2010.12.102

[15] W. Li, "Using Genetic Algorithm for network intrusion detection," in *Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*, Kansas, 2004, pp. 24-27.

[16] D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms: part 1. fundamentals," *University Computing*, vol. 15, no. 2, pp. 58-69, 1993.

[17] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656-667, Apr. 1994. http://dx.doi.org/10.1109/21.286385

[18] J. H. Min and F. C. H. Rhee, "An interval type-2 fuzzy PCM algorithm for pattern recognition," *Journal of The Korean Institute of Intelligent Systems*, vol. 19, no. 1, pp. 102-107, Feb. 2009. http://dx.doi.org/10.5391/JKIIS.2009.19.1.102

[19] H. C. Jeong, S. T. Seo, I. K. Lee, and S. H. Kwon, "Clustering method for reduction of cluster center distortion," *Journal of The Korean Institute of Intelligent Systems*, vol. 18, no. 3, pp. 354-359, Jun. 2008. http://dx.doi.org/10.5391/JKIIS.2008.18.3.354

[20] J. W. Han, S. H. Jun, and K. W. Oh, "Cluster merging using enhanced density based fuzzy C-means clustering algorithm," *Journal of The Korean Institute of Intelligent Systems*, vol. 14, no. 5, pp. 517-524, Aug. 2004. http://dx.doi.org/10.5391/JKIIS.2004.14.5.517

Xiao-Yun Ye He received the Bachelor degree in Computer Science from Gachon University, Korea in 2011. He also received a Master degree of Computer Science in Gachon University in 2013, Korea.
Tel: +82-10-9113-9266
E-mail: yxysun@gmail.com

Myung-Mook Han He received MS degree in Computer Science from New York Institute of Technology in 1987 and Ph.D. degree in Information Engineering from Osaka City University in 1997, respectively. From 2004 to 2005, he was a visiting professor at Georgia Tech Information Security Center(GTISC), Georgia Institute of Technology. Currently, he is a professor in the Department of Computer Engineering, Gachon Univ., Korea. His research interests include Information Security, Intelligent System, and Big Data.
Tel: +82-31-750-5522
E-mail: mmhan@gachon.ac.kr