Attribute Weighting via Differential Evolution for Attribute Weighted Clonal Selection Algorithm

Jia Wu[1], Zhihua Cai[1]★, Xiaolin Chen[1], Shuang Ao[2] ,Yongshan Zhang[1]
1. School of Computer Science,
China University of Geosciences, Wuhan 430074, P.R. China,
2. School of Computer Science,
University of Western Ontario, London, Ontario N6A 5B7, Canada.
Email: wujiawb@126.com; zhcai@cug.edu.cn; cxlcug@126.com; zhangyongshan789@126.com; sao@uwo.ca

***Abstract*—Classification is an important technology in data mining, while clonal selection algorithm (CSA) is a very effective classification method. Although CSA brings a new effective tool for solving complex problems, we can not completely say that it over-performs to other algorithms especially in the classification field. A main problem of CSA classifier is that it does not carry attribute imbalance. It uses a pure distance criterion to calculate affinity degree of the antibody and antigen. So we utilize weighting attribute scheme to balance the effects of attributes in classification process and attribute weighted CSA (AWCSA) comes into existence. The efficiency of AWCSA lies mainly in the attribute weighting scheme it uses. In this paper we use differential evolution (DE) algorithm to determine the weights of attributes and then use these weights in AWCSA. We evaluate the performance of new algorithm (DE-AWCSA) on six standard datasets. Experimental results show that this attribute weighting process highly benefits the classification accuracy.**

***Index Terms*—Clonal selection algorithm; Attribute weighting; Differential evolution; Classification accuracy**

I. INTRODUCTION

In recent years, data mining has been widely used in the area of science and engineering, such as bioinformatics, genetics, medicine, education and electrical power engineering. Classification algorithm is a data analysis method for data mining. Traditional data mining techniques and classification methods are decision tree, nearest neighbor, naive Bayesian, neural networks and support vector machines algorithm. While new complex and hard problems are coming into scene, current problem solving tools becomes insufficient and new tools are developed for this need. Techniques like artificial neural networks (ANN) and genetic algorithms (GA) are effectively used methods and developed as a result of that need and they bring artificial intelligence (AI) concept to the problem solving field.

Artificial immune system (AIS) is a new AI technique which can be applied to solve various branches of problem space like classification, virus detection, robotics, optimization, etc. The AIS aims at using ideas gleaned from immunology in order to develop systems capable of performing different tasks in various areas of research. Over the last few years, there has been an increasing interest in the area of artificial immune system (AIS) and its applications. Among many works in this new field of research, the most important and famous are those of Carter (2000) [1], De Castro &Von Zuben (2001) [2] and De Castro & Timmis (2002) [3].

Clonal selection algorithm (CSA) is a classic representative of artificial immune system (AIS). Clonal selection algorithm is a algorithm inspired by the clonal selection theory of acquired immunity that explains how B and T lymphocytes improve their response to antigens over time called affinity maturation [4]. The algorithm focuses on the theory of Darwinian where selection is inspired by the affinity of antigen-antibody interactions; reproduction is inspired by cell division. Clonal selection algorithm is most commonly applied to optimization and pattern recognition domains. Despite of its generality in this wide range of application area, successful studies obtaining better results are not so many.

This paper aims to reach higher classification accuracy by assigning weights to important attributes in clonal selection algorithm for classification. In a classification process the contribution of attributes may be different. So, giving weights to attributes can correct this imbalance and improve classification accuracy. This can be done with some modifications to affinity measures of CSA and then a new algorithm named AWCSA (Attribute Weighted Clonal Selection Algorithm) comes into existence. In AWCSA, we calculate attribute weights using statistical information in dataset such as standard deviation and mean value of attributes. But research shows that this not only increases the complexity of the algorithm, but also sometimes can not get good appropriate weights. Then for this disadvantage we utilize the differential evolution (DE)[5][6] for determining attribute weights which are then used in AWCSA classification, namely DE-AWCSA. We also design experiments to compare the performance of DE-AWCSA to that of CSA and AWCSA form the part of classification accuracy.

This work was supported by National Natural Science Foundation of China under Grant Nos.61075063, the Fund for Outstanding Doctoral Dissertation of CUG Nos. 2235122, Self-Determined and Innovative Research Funds of CUG Nos. 1210491B16.

★corresponding author.

doi:10.4304/jcp.7.12.3013-3019

This paper is organized as follows. In Section 2, immune systems including natural and artificial immune system are introduced, and then is their product clonal selection algorithm. Section 3 is allocated for the AWCSA. The DE-AWCSA is proposed in Section 4. Section 5 contains the experimental methods and results. In Section 6, experimental results are concluded and future works are emphasized.

II. RELATED WORK

A. Natural and Artifical immune System

Natural immune system is a kind of pattern detection system whose main duty is to provide body to stay in homeostasis with neural and endocrine system. Throughout a person's lifetime, the body is exposed to a huge variety of pathogenic (potentially harmful) material. The immune system contains lymphocyte cells known as B-cells and T- cells, each of which has a unique type of molecular receptor (location in a shape space). The receptors of these cells take important roles in immune responses. Especially B cells enjoy a great importance because of their secreted antibodies (Abs) that take very critical roles in adaptive immune response [7].

Inspired by natural immune system, Artificial Immune System emerged in the 1990s as a new computational research area. It is a machine-learning algorithm that embodies some of the principles and attempts to take advantages of the benefits of natural immune systems to tackle with complex problem domains. Clonal Selection Algorithm (CSA) is one such system inspired by the clonal selection theory of acquired immunity, which has shown success on a broad range of engineering problem domains.

B. Clonal Selection Theory

The clonal selection theory was proposed by Burnet, Jerne, Talmadge, with the use of describing the functioning of acquired immunity, and is a specific theory to describe the diversity of antibodies used to defend the organism from invasion [4]. An antibody is a molecule produced by B lymphocyte cells that can neutralize a single antigen. Each B lymphocyte (white blood cell) creates unique or customized antibodies of a specific type. The theory, when originally proposed, was a point of contention and competed with another model called template theory. Today, the clonal selection theory is regarded as a fact with the overwhelming amount of empirical evidence supported.

The theory specifies that the organism has a pre-existing pool of heterogeneous (individually unique) antibodies that can recognize all antigens with some level of specificity. When an antigen matches (selects) an antibody, it causes the cell to chemically bind to the antigen, replicate and produce more cells with the same receptor. During the cell proliferation stage, genetic mutations occur in the clone of cells that promote the match or affinity with the antigen. This allows the binding ability of the cells to improve with time and expose to the antigen. This selection of replication cells by antigens can be viewed as a type of Darwinian microcosm where the fittest cells (best match with antigens) are selected for survival, and genetic mutation provides cell variation.

C. Clonal Selection Algorithm(CSA)

The Clonal Selection Algorithm, usually called CSA, is said to be inspired by the following elements of the natural clonal selection theory [8]:

- Maintenance of a specific memory set
- Selection and cloning of most stimulated antibodies
- Death of non-stimulated antibodies
- Affinity maturation (mutation)
- Generation and maintenance of diversity

The goal of the algorithm is to develop a memory pool of antibodies that represents a solution to an engineering problem [9]. In this case, an antibody represents an element of a solution or a single solution to the problem, and an antigen represents an element or evaluation of the problem space.

Using the clonal selection algorithm for classification involves two steps. The first stage is the training of the system with a set of sample data. After the training is completed, the test data set will be given for classification.

The Figure 1 provides an overview of the steps of the clonal selection algorithm：

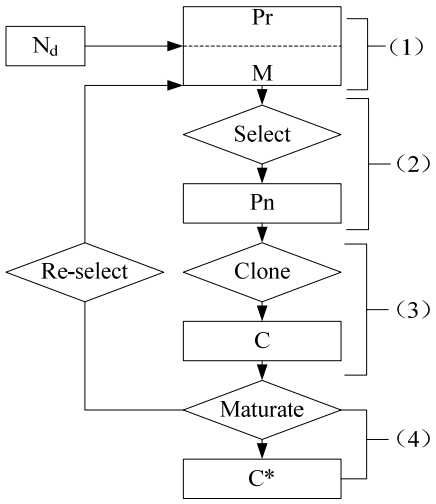

Figure 1. Block diagram of the clonal selection algorithm.

- Training

The algorithm works as in Figure 1 (after each six steps we have one cell generation):

1) Generate a set (P) of candidate solutions, composed of the subset of memory cells (M) added to the remaining (Pr) population (P = Pr + M);
2) Determine (Select) the n best individuals of the population (Pn), based on an affinity measure(distance between antibody and the antigen);
3) Reproduce (Clone) these n best individuals of the population, and form a temporary population of clones (C). The clone size is an increasing function of the affinity with the antigen;
4) Submit the population of clones to a hyper-mutation scheme, where the hyper-mutation is proportional to

the affinity of the antibody with the antigen. A maturated antibody population is generated (C*);

5) Re-select the improved individuals from C* to compose the memory set M. Some members of P can be replaced by other improved members of C*;

6) Replace d antibodies by novel ones (diversity introduction). The lower affinity cells have higher probabilities of being replaced.

- Classification

After training has completed, the evolved memory cells Ab{M} are available for the use for classification. Each memory cell is presented with a data item. By calculating the affinity between memory cell and a test data, the test data is classified into the class that has the maximum affinity.

III. AWCSA

In clonal selection algorithm, the affinity or distance between antibody and the antigen is calculated by the Euclidean distance criteria (Eq. (1)):

$$D = \sqrt{\sum_{i=1}^{L} (ab_i - ag_i)^2} \quad (1)$$

Where *ab* and *ag* are two points represented by a vector respectively and L is the length of these vectors. According to this formula, all of the attributes share the same effect in determining distance. However, there are such data sets that some attributes of them have no effect on the class of data while some other attributes are more important in determining class. The AWCSA algorithm proposed for minimizing the effect of this problem is a supervised Artificial Immune System based on attributes weighted distance criteria. AWCSA is a two-level classification system in which attribute weights of each class are formed in one level and a training procedure with these weights takes place in the other.

A. Attribute Weighting

The imbalance of attributes has great influence on the performance of algorithm [10][11]. So, if it is assigned higher weights to the attributes that are more important in determining one class and if these weights are used in calculation of distance, it can be prevented to make a misclassification of the two distant data according to the Euclidean norm in the same class [12]. Starting from this point, the proposed attribute weighting depends on the following base: if one attribute doesn't change very much among the data of one class, this attribute is one of the characteristic attributes of related class and it must have a higher weight than others.

The applied attribute weighting procedure in the AWCSA is as follows:

1) Normalization of each attribute in data set between 0-1.

2) Determine the antigens of each class
$\rightarrow Ag\_class_j (j:1,.....n, n: number\ of\ class)$

3) For each class do:
For $Ag\_class_{(L\times Nc)}$ to be a matrix that involves the antigens of that class;

(L : attribute num, Nc : ag num of class)

(3.1) For i^{th} attribute do: *(i : 1....L)*

Evaluate standard deviation of i^{th} attribute with Eq.(2):

$$std\_dev_i = \sqrt{\frac{1}{Nc}\sum_{k=1}^{Nc}(Ag_{k,j} - mean(Ag_i))^2} \quad (2)$$

Here $Ag_{k,i}$ is the i^{th} attribute of k^{th} Ag in j^{th} class; mean (Ag_i) is the mean of i^{th} attribute of all Ags in j^{th} class.

Calculate the weights as follows:

$$w_{j,i} = 1/std\_dev_i, (i = 1,....L;\ j = 1,....n) \quad (3)$$

(3.2) normalize the weights of j^{th} class.

The calculated $w_{n\times l}$ matrix is a normalized weight matrix involving the weights of each attribute for each class and this matrix is used in distance calculations of the training algorithm of AWCSA. Here, in the attribute weighting procedure, a means of normalization of attributes for each class by standard deviation is performed. By doing so, each class has its own set of attribute weights.

B. AWCSA

AWCSA is a simple and supervised AIS which uses weighted distance criterion while calculating distance between system units (Antibody-Ab) and input data (Antigen-Ag). The system can be shown basically as in Fig. 2.

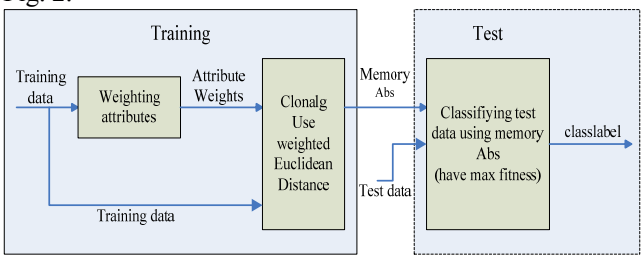

Figure 2. AWCSA classification system

The training phase of the system begins with a pre-processing step which is used for determining weights of attributes. In this phase, each attribute's standard deviation in one class is calculated and reciprocal of this value is taken as the weight of that attribute for related class. By this way, a weight matrix is obtained containing weights of each attribute in each column and for each class in each row. This weight matrix is then used in the main training procedure of AWCSA while calculating Euclidean distances of Abs to the presented data (Ag) in the following way:

$$Affinity' = \sqrt{\sum_{k=1}^{L} w_{j,k}(Ab_{j,k} - Ag_{i,k})^2} \quad (4)$$

Here $Ab_{j,k}$ and $Ag_{i,k}$ are the k th attribute of Ab_j and Ag_i ,respectively; $w_{j,k}$ is the weight of k th attribute that belongs to the class of Ab_k .

The training procedure of the learning algorithm conducts the following steps:

(1) For each Ag_i do : *(i : 1,....,N)*

(1.1) Determine the class of Ag_i . Call memory Abs of that class and calculate the distances between Ag_i and the memory Abs with Eq. (1).

(1.2) If the minimum distance among the calculated distances above is less than a threshold value named as suppression value (supp) return to step 1.

(1.3) Form a memory Ab for Ag_i :

At each iteration do:

(1.3.1) Make a random Ab population with Ab = [Ab_mem; Ab_rand] and calculate the distances of these Abs to Ag_i .

(1.3.2) Select m nearest Abs to Ag_i ; clon and mutate these Abs (Ab_mutate).

(1.3.3)Keep the m nearest Abs in the Ab_mutate population to Ag_i as Ab_mem temporary memory population.

(1.3.4) Define the nearest Ab to Ag_i as Ab_cand, candidate memory Ab for Ag_i and stop iterative process if the distance of Ab_cand to Ag_i is less that a threshold value named as stopping criterion (sc).

(1.3.5) Concatenate Ab_cand as a new memory Ab to memory matrix of the class of Ag_i .

(1.4) Stop training.

After training is completed, the evolved memory cells Abs are available for the use for classification.

IV. DE-AWCSA

Differential evolution (DE) is arguably one of the most powerful stochastic real-parameter optimization algorithms in current use [13-15]. Regarded as a standard evolutionary algorithm (EA), DE operates through similar computational steps. However, unlike traditional EAs, DE is a new heuristic approach mainly having three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using few control parameters. Since its inception in 1995, DE has drawn the attention of many researchers all over the world, resulting in a lot of variants of the basic algorithm with improved performance [16].

In our study, we utilize DE in the attribute weighting stage of AWCSA in the following manner: because our objective in attribute weighting is to reduce the classification error of the classifier, we used the DE processes to find optimum weight configuration for features that gives the minimum classification error when used in the classifier. Thus, we formed a population consisting of the individuals, which represents the weights for attributes in each class and apply DE processes (crossover–mutation–selection) to these individuals. The best individual is the one causing lowest test classification error when used in the AWCSA with some training and test data. The procedure in finding optimum weights via DE is given in detail as follows:

A. Form Initial Population of Individuals Which Represents the Weights of Features for Each Class:

The number of features denotes l and there will be m classes. Here, a population consisting of individual is formed. Each individual is a string with continuous values. These values represent the weights and because there will be l weight for each class, $l \times m$ values form each individual, which means, each individual will be represented as a string with a length of $l \times m$.

B. Calculate the Fitness of These Individuals:

With some training data, AWCSA is trained to use the individual whose cost value is to be calculated. Then, some test data is presented to the trained AWCSA and classification error is calculated in percentage form. The cost value of given individual is this test classification error. This calculation is conducted for each individual in population. The best individuals are the ones which have minimum cost values.

C. Select a Number of Individuals Having Minimum Cost Values:

The population members are ranked in ascending order with regard to their cost values from minimum to maximum and the first k individuals are selected to be subjected to crossover and mutation processes.

D. Crossover and Mutation:

The above selected k members of the population are subjected firstly to crossover operation. There are various routines for this process [16] and we selected single point crossover routine in this operation. Then, the resulted individuals are presented to the mutation process with a constant mutation rate.

E. Convergence Test:

The resulted individuals after crossover and mutation processes form a part of the new population after one generation. Some randomly generated individuals are also added to this new population to provide diversity and generalization. The fitness (cost values) of these new individuals is calculated as previously in step 2. Then it is decided whether to stop generations or not by looking for the minimum cost value. If the minimum cost value stays the same throughout a predefined number of generations then the algorithm is stopped. Otherwise, the processes in steps 2–5 continue to be done until maximum generation number is reached.

The above procedure is conducted only to determine the weights of features for each class. The best individual causing minimum test classification error in AWCSA is determined and this individual is then used in AWCSA as the weights of features for each class in the distance calculation both in training and test procedures of AWCSA.

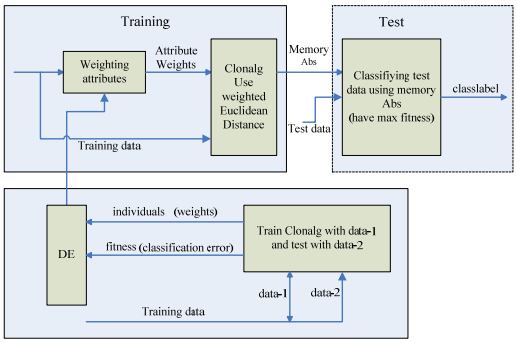

Figure 3. DE-AWCSA classification system

V. Experimental Methods and Results

The previous section describes the theory of CSA, AWCSA and DE-AWCSA. The main purpose of the current research is to conduct a detailed comparison among them and determine their effects on the classification accuracy. This section describes the experimental data, the methodology, and results of the experiment.

A. Experimental Data

We run our experiments on 6 standard UCI data sets [17] in Weka [18]. We choose these 6 data sets from 36 standard UCI data sets according to the fact that AWCSA is inappropriate for the data set which has many attributes. The data characteristics are listed in Table 1. Before classification, we should handle the data with following three steps:

1. Dealing with the missing attribute values. For the special datasets, we use the unsupervised attribute filter Replace Missing Values in Weka to replace all the missing attribute value.

2. Discretize the numeric attribute values. For the special datasets, we use the unsupervised filter Discretize in Weka to handle all numeric attribute values in each data set.

3. Removing some of the useless attributes [19]. There is just one such attribute in the above-mentioned 6 data sets: "Hospital Number" attribute in data set "colic.ORIG". In order to remove these useless attributes we adopt the unsupervised filter named Remove in Weka.

TABLE 1

Description of data sets used in the experiments

Dataset	Instances	Attributes	Classes	Missing	Numeric
colic	368	23	2	Y	Y
heart-statlog	270	14	2	N	Y
hypothyroid	3772	30	4	Y	Y
splice	3190	62	3	N	N
vote	435	17	2	Y	N
liver-disorders	345	7	2	N	N

B. Experimental Methods

In all, three different algorithms were tested on all of the selected data sets. The particular factors of interest, as mentioned above, were classification accuracy. In order to obtain independent test data and reliable results, each original data set was split randomly (90/10) into a training and a test data set. Each time, one of the folds is selected as test data and other 9-folds are combined as training data. The training data is then divided into two parts to determine the weights by DE procedure as explained in Section IV. After determination of the weights, the training data is again presented to AWCSA to train system with obtained weights for attributes. The test fold is used to determine the test classification accuracy. These processes conducted for all 10-folds and 10 test results are averaged to have the resulted test classification accuracy of the system. The configuration of the system is shown in Fig. 3 for DE-AWCSA for one fold.

The used parameters for AWCSA and DE in experiment are given in Tables 2 and 3, respectively. Here in Table 2, 'Itnum' is the maximum iteration number for the training of AWCSA, 'Supp' is the threshold value used in the generation of memory Abs in the training phase of AWCSA, and 'Sc' is the stopped criterion for the AWCSA training.

TABLE 2

Used parameters of AWAIS

Parameter	Value
Itnum	200
Supp	0.08
Sc	0.0001

The parameters given in Table 3 are 'Maxgen', 'Nind' and 'Gen' used in DE. 'Maxgen' is the maximum generation number; 'Nind' is the population size. The parameter 'Gen' refers to the number of generations used in the convergence test in that if minimum cost value does not change in 'Gen' generations, the DE is stopped.

TABLE 3

Used parameters of DE

Parameter	Value
Maxgen	100
Nind	50
Gen	10

C. Experimental Results

Classification accuracy is a very important criteria for evaluating a performance of a classifier [20]. Thus, we use the classification accuracy to evaluate the performance of CSA, AWCSA and DE-AWCSA.

Classification accuracy, equaling to the percentage of instances correctly classified, refers to the predictive ability of a classification algorithm in terms of classifying an independent set of test data.

In this paper, experiments are designed on Weka in order to compare the performance of AWCSA and DE-AWCSA to CSA. The comparison results on accuracy values of the three algorithms on each data set are shown in Table 4. The average values are summarized at the bottom of the table.

TABLE 4

DESCRIPTION OF DATA SETS USED IN THE EXPERIMENTS

Dataset	CSA(%)	AWCSA(%)	DE-AWCSA(%)
colic	73.13± 8.78	***74.40± 9.20***	***75.41± 10.50***
heart-statlog	82.22± 11.15	***84.07± 9.57***	***84.44± 8.63***
hypothyroid	83.56± 2.9	***88.15± 27.59***	***91.99± 0.9***
liver-disorders	76.57± 8.11	***77.13± 7.79***	***79.99± 5.92***
splice	87.55± 3.30	81.85± 1.37	***89.97± 2.57***
vote	85.29± 4.98	***87.57± 9.02***	***90.84± 5.97***
Average	81.39	82.20	85.44

Then we can know from Table 4 that the attributed weighted clonal selection algorithm (AWCSA, DE-AWCSA) got higher classification accuracy than traditional clonal selection algorithm.

For 'splice' data set, it has 3190 instances and 62 attributes. This problem is too complex for AWCSA, so the classification accuracy is not so good. But DE-AWCSA also performs well on this complex problem. An increase with 2.42% in the classification accuracy was obtained for 'splice' by DE-AWCSA.

For 'hypothyroid' data set, attribute imbalance has great effect on the performance of CSA. The improved algorithm AWCSA and DE-AWCSA enhance the classification accuracy significantly. An increase with 4.59% in the classification accuracy was obtained for 'hypothyroid' by AWCSA. A better increase was reached by DE-AWAIS with 8.43% for 'hypothyroid' dataset.

As can be seen from Table 4, the highest average classification accuracy was obtained by DE-AWCSA with a classification accuracy of 85.44%, compared to the average classification accuracy of CSA 81.39%. And indeed such an increase is very important for a classifier from the application perspective.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we first proposed a new algorithm named AWCSA according to the shortage of CSA classifier. Then, we proposed a different strategy for the attribute weighting. We used a basic real-valued DE procedure to find optimum weights giving minimum classification error for the attributes. In order to evaluate the classification performance of the algorithms (CSA, AWCSA, DE-AWCSA), we applied them to the six standard classification datasets from the UCI Machine Learning Repository.

Experimental results show that AWCSA can enhance the classification accuracy for most problems, but is powerless for complex problems. Compared to AWCSA, taking advantage of DE's strong global search ability and high convergence rate, DE-AWCSA has obtained very good results with high performance in classification accuracy for all data sets.

In summary, there exists a number of ways to improve the performance of clonal selection algorithm in classification and future work will explore these methods.

REFERENCES

[1] J. H. Carter, "The immune system as a model for pattern recognition and classification," *Journal of the American Medical Informatics Association*, vol. 7, no. 3, pp. 28–41, 2000.

[2] L. N. De Castro and F. J. Von Zuben, *AiNet: An Artificial Immune Network for Data Analysis*. Data Mining, A Heuristic Approach. Idea Group Publishing, USA, 2001.

[3] L.N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, 2002.

[4] J. Brownlee, *Clonal Selection Theory &Clonalg the Clonal Selection Classification Algorithm*. Swinburne University of Technology, 2005.

[5] R. Storn and K. Price, "Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, " *Global Optimization*, vol. 11, pp. 341-359, 1997.

[6] A. Nobakhti and H. Wang, "A simple self-adaptive differential evolution algorithm with application on the alstom gasifier," *Applied Soft Computing*, vol. 8, pp. 350-370, 2008.

[7] C.G, Xue, L.L, Dong and G.H. Li, "An Improved Immune Genetic Algorithm for the Optimization of Enterprise Information System based on Time Property," *Journal of Software*, vol. 6, no. 3, pp. 436-443, 2011.

[8] L. N. De Castro and F. J.Von Zuben, "Learning and Optimization Using the Clonal Selection Principle," *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, vol. 6, pp. 239-251, 2002.

[9] A. S. Perelson and G. F. Oster, "Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliabilty of Self-Nonself Discrimination," *Journal of Theoretical Biology* ,vol. 81, pp. 645–670, 1979.

[10] J.Wu, Z. Cai and S. Ao, "Hybrid Dynamic K-Nearest-Neighbor and Distance and Attribute Weighted Method for Classification," *International Journal of Computer Applications in Technology*, vol. 43, no. 4, pp. 378-384, 2012.

[11] S. Şahan, H. Kodaz, S. Güneş, and K. Polat, "A new classifier based on attribute weighted artificial immune system," in Proc. ISCIS 2004, C. Aykanat et al. Eds. Springer-Verlag, pp.11–20.

[12] J.Wu and Z. Cai, "Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB)," *Journal of Computational Information Systems*, vol. 7, no. 5, pp.1672-1679, 2011.

[13] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Computing*, vol. 10, pp. 673-686, 2006.

[14] J.F. Yan, C.H. Guo and W.Y. Gong, "Hybrid Differential Evolution with Convex Mutation" *Journal of Software*, vol. 6, no. 11, pp. 2321-2328, 2011.

[15] A. Salman, A. Engelbrecht and M. Omran, "Empirical analysis of self adaptive differential evolution," *European Journal of Operational Research*, vol. 183, pp. 785-804, 2007.

[16] S. Das, and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evolution Computing*, vol. 15, no. 1, pp. 4-31, 2011.

[17] C. Merz, P. Murphy and D. Aha, UCI repository of machine learning databases. Department of ICS, University of California, Irvine,1997.

http://www.ics.uci.edu/mlearn/MLRepository.html.

[18] H. W. Ian and F. Eibe, "Data mining: practical machine learning tools and techniques, 2nd edn," Morgan Kaufmann, San Francisco, 1996.

http://prdownloads.sourceforge.net/weka/datasets-UCI.jar

[19] S. J. Hong, "Use of Contextual Information for Feature Ranking and Discretization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, no. 5, pp. 718-730, 1997.

[20] L.Jiang, Z. Cai, D.Wang and H.Zhang, "Improving Tree augmented Naive Bayes for class probability estimation," *Knowledge Based System*, vol. 26, pp. 239-245, 2012.

Jia Wu received the B. Eng degree in computer science from China University of Geosciences (CUG), Wuhan, China, in 2009. He is currently studying in CUG for his PhD degree in Geology Information Technology. His current research interests include data mining and knowledge discovery, machine learning, evolutionary computation, and hyperspectral remote sensing.

Zhihua Cai received the PhD degree from China University of Geosciences, in 2003. Dr. Cai is currently a faculty member at School of Computer Science, China University of Geosciences, Wuhan, China. His main research areas include data mining, machine learning, evolutionary computation, and their applications.

Xiaolin Chen received the B. Eng degree in computer science from China University of Geosciences (CUG), Wuhan, China, in 2010. She is currently studying in CUG for her Msc degree in Computer Science. Her current research interests include data mining and machine learning, evolutionary computation.

Shuang Ao received the B. Eng degree in computer science from China University of Geosciences (CUG), Wuhan, China, in 2009. He is currently studying in UWO for his PhD degree in Computer Science. His current research interests include data mining and knowledge discovery.

Yongshan Zhang is currently studying in CUG for her B. Eng degree. Her current research interests include data mining and machine learning.