An Artificial Immune System for Extracting Fuzzy Rules in Credit Scoring

Ehsan Kamalloo and Mohammad Saniee Abadeh

Abstract—**Various credit scoring models have been proposed to estimate credit risk of loan applicants. Recently, the use of artificial immune systems (AIS) in credit problems has been increased. AIS is inspired from natural immune system which has the ability of determining self from non-self. The aim of this study is constructing an AIS-based model to extract fuzzy rules to predict the likelihood of customers such as good/bad payer. The rules have made our model human-understandable which helps experts to organize their knowledge from the domain. We use Weka data mining software to compare our classifier with several well-known classifiers. The evaluation criterias which have been used in this paper are average correct classification rate, precision, recall and f-measure. The experiments were performed on Australian and German Credit Approval datasets. The results demonstrate that proposed AIS-based classifier has high accuracy and interpretability which makes it competitive to several well-known classification systems.**

I. INTRODUCTION

FINANCIAL institutions and banks widely use credit scoring models to distinguish between good and bad credit and assign credit to good customers. Loans are often the most important base of risk in banks. Using credit scoring can reduce the time of loan approval process [1] and save cost per loan and enhance credit decisions which help lenders to ensure they are applying the same criteria to same group of borrowers [2] and banks can supervise the existing loans easier [3].

Since the rapid growth of auto-financing in the last two decades, data mining is being applied to credit scoring problem. The first study into credit scoring was started by Durand in 1941 to classify credit applications as good or bad payers [4]. Fair and Isaac developed a credit scoring model in the early 60s [5]. Since then, various models have been developed using traditional statistical techniques. With growth of data mining over the last decades, the recent techniques of credit risk assessment treat lending decisions as binary classification problems [4]. A variety of data mining tools and statistical models has been applied to this problem so far, which involve linear discriminant models, logistic regression models, k-nearest neighbor models, decision tree models, neural network models, genetic programming models and support vector machines (SVM) [11].

SVM is one of the popular methods presented in credit problems. Choosing the optimal input feature subset and setting the best kernel parameters are two problems must have been solved to propose a SVM method [11]. Zhang et al. [4] and Huang et al. [11] have used SVM for credit scoring and have shown SVM has high accuracy for the problem.

Hybrid data mining approaches also have proposed for effective credit scoring. Yao [7] has used neighborhood rough set and SVM as a hybrid classifier. In this classifier neighborhood rough set is used for feature selection. Zhang et al. [1] proposed hybrid model based on genetic programming (GP) and SVM. This model used GP to extract if-then rules and for remaining instances of dataset it used discriminator based on SVM. Jiang [8] used decision tree and simulated annealing to build a model. In this paper, authors combine local search strategy of decision tree algorithms and global optimization of simulated annealing algorithm.

In recent years, many bio-inspired algorithms are presented for solving classification problems such as credit card fraud detection [12], credit scoring, security and other applications. Evolutionary algorithms like neural network and genetic algorithm are one group of novel methods that has been competitive to other techniques. Many evolutionary algorithms are proposed for credit scoring like [1,6,7]. Recently artificial immune systems (AIS) are successfully employed in a wide variety of application areas. Artificial immune systems are computational systems inspired by the processes of the natural immune system. Artificial immune systems emerged in the 90s as a new computational model in AI. Hunt and Cooke apply AIS to pattern recognition problems in 1996 [8]. Timmis and de Castro define AIS as "adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving" [16]. Researchers worked mostly on the theories of immune networks, clonal selection and negative selection [9]. Leung et al. [5] propose a simple AIS (SAIS) algorithm that adopts few key concepts of AIS (affinity measure, cloning and mutation) in credit scoring. They found SAIS is a very competitive classifier.

Exploring new techniques in credit scoring to improve the performance can save a lot of money. In this paper, we propose an AIS uses clonal selection. Within the proposed AIS, fuzzy logic has been applied to extract fuzzy rules, so we name the proposed classifier fuzzy AIS (FAIS). Fuzzy logic has been applied to classification problems. Its

E. Kamalloo is a M.Sc. student in the Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran (e-mail: e.kamalloo@modares.ac.ir).

M. Saniee Abadeh is with the Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran (corresponding author to provide phone: 98-2182884349; e-mail: saniee@modares.ac.ir).

This work was supported by Iran Telecommunication Research Center.

advantages are powerful capabilities of managing uncertainty and vagueness [13]. Fuzzy classifiers generate a rule base structure. These rules are represented in linguistic forms that make them interpretable by users [14]. Experts can validate and correct the rules. This increases the interaction with users. Lei and Ren-hou [14] proposed a classifier based on immune principles and fuzzy rules. They apply their algorithm on 15 well-known UCI machine learning repository [15] data sets. They achieve high accuracy. The fuzzy AIS term in our proposed method (FAIS) is similar to [14]. The fitness function used in [14] is simple but in this study we add some useful terms to improve accuracy (details are described in section IV). The major contribution of this study is using fuzzy rule extraction technique in context of credit scoring. We use Weka [10] to compare FAIS with other classifiers.

The rest of this paper is organized as follows. Section II discusses artificial immune system algorithm and the concepts we used in our algorithm. Section III describes pattern classification with fuzzy logic. The proposed algorithm is presented in section IV. Section V provides information of the experiments performed and results achieved and section VI concludes the paper.

II. ARTIFICIAL IMMUNE SYSTEM

Researchers use biology as a basis of inspiration in several different fields. There have been many methods on the biological metaphors, for example neural networks and genetic algorithms. Over the last decade, natural immune system as a metaphor for computation has been increasingly used in several domains. The main features of immune system are learning, adaptability, and memory mechanisms which are making it a rich source for inspiration [16]. In [17] a general review of application areas to which AIS algorithm has been currently applied presented.

A. Natural Immune System

The immune system defends body from assaults of foreign substances called antigens. The immune system uses B-cells which are kind of white blood cells. B-cells produce some proteins called antibodies [8]. The mechanism of immune system is similar to 'hunt and destroy' which works on the cells of our bodies [16]. It has the significant talent of learning about the foreign substances called pathogens. The body responds to these foreign substances by producing antibodies, which can attack the antigens. Antigens are associated with the pathogens.

There are two types of white blood in our bodies: B cells and T cells. A B cell holds antibodies on its shell which can identify the antigens invading the body. The matching between antigen and antibody is complementary and is similar to 'lock and key'. T cells do not interact to antigens directly. They circulate through the body and scan the surface of body cells for the presence of foreign antigens that have been combined with the cell. Then T cells bind to these cells until B cells helping to stimulate them [18].

B. Clonal Selection Principle

The basic features of an immune response to an antigenic stimulus are provided by clonal selection principle. The idea is that only those cells that recognize the antigen select to proliferate and others cannot clone. The properties of this principle are:

1. A cell is duplicated and is subjected to mutation with high rates to form a new cell.
2. Newly differentiated lymphocytes transporting self-reactive receptors eliminate.
3. On contact of mature cells with antigens proliferation and differentiation occur [19].

Many clonal selection based algorithms have been proposed, most of them have been applied to optimization problems [9]. Some of these algorithms are introduced in [9]. The B Cell Algorithm (BCA) [25] is one of the basic algorithms in clonal selection principle and it is applied to optimization problems. An outline of BCA is shown in Fig. 1.

input: g(v) = function to be optimized
output: set of solutions for functions
begin

a. Create an initial population P of individuals in shape-space
b. For each $v \in P$, evaluate g(v) and create clone population C
c. Select a random number of $v' \in C$ and apply the hypermutation operator
d. if g(v') > g(v) then replace v by clone v'
e. Repeat steps b-d until stopping criteria is satisfied

end

Fig. 1. An outline of BCA [9].

III. PATTERN CLASSIFICATION WITH FUZZY LOGIC

Let us assume that the pattern classification problem is a c-class problem in an n-dimensional continuous pattern space and the training data includes M real vectors $x_p = (x_{p1}, x_{p2}, \dots, x_{pn}), p = 1,2,\dots,M$ from c classes ($c \ll M$).

Because the pattern space is $[0,1]^n$, attribute values of each pattern are $x_{pi} \in [0,1]$ for $p = 1,2,\dots,M$ and $i = 1,2,\dots,n$. In experiments of this paper (see section VI), we normalize all attribute values of the data set into the unit interval [0,1].

In our fuzzy classifier system, we use fuzzy if-then rules of the subsequent form for the n-dimensional pattern classification problem.

Rule R_j : If x_1 is A_{j1} and … and x_n is A_{jn}, then Class C_j with $CF = CF_j$.

Where R_j is the label of the *jth* fuzzy if–then rule,

$A_{j1},\ldots,A_{jn}$ are antecedent fuzzy sets in the unit interval [0,1], C_j is the resultant class (i.e., one of the given c classes), and CF_j is the certainty factor of the fuzzy if–then rule R_j. It is a real number in the unit interval [0,1]. It should be noted that some antecedent conditions can be "*don't care*". Introducing "*don't care*" conditions reduces the number of antecedent conditions of rules. These rules are more human-understandable than other rules.

We use a typical set of linguistic values in Fig. 2 as antecedent fuzzy sets. The membership function of each linguistic value in Fig. 2 is specified by homogeneously partitioning the domain of each attribute into symmetric triangular fuzzy sets. We use such a simple specification in experiments to demonstrate the high performance of our fuzzy classifier system, even if the membership function of each antecedent fuzzy set is not tailored. However, we can use any tailored membership functions in our fuzzy classifier system for a particular pattern classification problem.

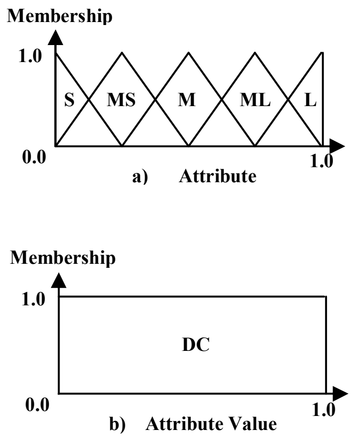

Fig. 2. The used antecedent fuzzy sets in this paper. a) 1: Small, 2: medium small, 3: medium, 4: medium large, 5: large. b) 0: don't care.

In our classifier system, the following steps are applied to calculate the grade of certainty CF_j of each fuzzy if-then rule [14,20,21]:

Step 1: Calculate the compatibility of each training pattern $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ with the fuzzy if–then rule R_j by the following product operation:

$$\mu_j(x_p) = \prod_{i=1}^{n} \mu_{ji}(x_{pi}) \qquad p = 1,2,\ldots,M \qquad (1)$$

Where $\mu_{ji}(x_{pi})$ is the membership function of i^{th} attribute of p^{th} pattern and M denotes the total number of patterns.

Step 2: Calculate the relative sum of the compatibility grades of the training patterns with the fuzzy if–then rule R_j for each class:

$$\beta_{class\,h}(R_j) = \frac{\sum_{x_p \in class\,h} \mu_j(x_p)}{N_{class\,h}}, h = 1,2,\ldots,c \qquad (2)$$

Where $\beta_{class\,h}(R_j)$ is the sum of the compatibility grades of the training patterns in *class h* with the fuzzy if–then rule R_j and $N_{class\,h}$ is the number of training patterns which their corresponding class is *class h*.

Step 3: Find class $\hat{h}$ that $\beta_{class\,h}(R_j)$ is maximum.

$$\beta_{class\,\hat{h}}(R_j) = max\{\beta_{class\,1}(R_j), \ldots, \beta_{class\,c}(R_j)\} \qquad (3)$$

If two or more classes take maximum value and training pattern compatible with the fuzzy if-then rule does not exist, the resultant class of rule R_j cannot be determined.

Step 4: The grade of certainty CF_j is determined as follows:

$$CF_j = \frac{\left(\beta_{class\,\hat{h}}(R_j) - \bar{\beta}\right)}{\sum_{h=1}^{c} \beta_{class\,h}(R_j)} \qquad (4)$$

Where

$$\bar{\beta} = \frac{\sum_{h=1, h\neq\hat{h}}^{c} \beta_{class\,h}(R_j)}{c-1} \qquad (5)$$

Now, we can specify the certainty grade for any combination of antecedent fuzzy sets.

The task of our fuzzy classifier system is to generate combinations of antecedent fuzzy sets for generating a rule set *S* with high classification rate. When a rule set *S* is given, an input pattern $x_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ is classified by a single winner rule R_M in *S*, which is determined as follows:

$$\mu_M(x_p) \cdot CF_M = max\{\mu_j(x_p) \cdot CF_j | R_j \in S\} \qquad (6)$$

That is, the winner rule has the maximum product of the compatibility and the certainty grade (CF_j).

Each fuzzy if-then rule is coded as a string. The following symbols are used for denoting the six linguistic values which are presented in Fig. 1:

0: don't care (DC), 1: small (S), 2: medium small (MS), 3: medium (M), 4: medium large (ML), 5: large (L).

Fig. 3 shows antecedent of a fuzzy rule which is coded as string. The decoded rule becomes "if attribute*1* is *MS* and attribute*3* is *MS* and attribute*5* is *M* and attribute*6* is *L* and attribute*n* is *S* then class is *c*".

2	0	2	0	3	5	0	…	0	0	0	1

Fig. 3. The antecedent of a fuzzy rule coded as a string.

IV. Proposed Credit Scoring System

This section presents an overview of the proposed algorithm which is named FAIS. It is based on the clonal selection principle. The clonal selection principle is used to explain the important features of an adaptive immune response to an antigenic stimulus. The main idea is that only those B-cells that identify the antigens are selected to proliferate. The selected cells are exposed to an affinity maturation process, which develops their affinity to the selective antigens [14]. In this paper, no distinction is made between a B-cell and its receptor, known as an antibody, so that every element of our artificial immune system will be called B-cell.

FAIS has used a population of B-cells. Each B-cell has an age to live in the population. A B-cell dies whenever its age reaches to zero. The objective is to obtain a set of rules with high accuracy. Each B-cell represents a rule. As we mentioned in section IV, each rule is coded like in Fig. 3.

Iterative rule learning approach has been used to learn rules.

A. Fitness Function

In the literature of fuzzy classification, several fitness functions have been proposed [23,20-22]. FAIS have used 3 fitness functions which are computed according to equations (7) to (11). The basic formula which is named $fitness^1$ is introduced in [22]. w^p is weight of instance x^p which reflects the frequency of the instance. We add two terms to $fitness^1$ to make $fitness^2$. NCP is abbreviated of number of classified patterns and NMP is number of misclassified patterns. The final formula is $fitness^3$ which contains a term with rule length. The higher rule length the less fitness is achieved. It led the B-cells to short rule length. So, the final term is useful for comprehensibility. The less rule length the more understandable by human. The desired fitness equation must be chosen to run FAIS.

$$f_P(R_j) = \frac{\sum_{p=1|c^p=c_j}^{K} w^p \cdot \mu_j(x^p)}{\sum_{p=1|c^p=c_j}^{K} w^p} \tag{7}$$

$$f_N(R_j) = \frac{\sum_{p=1|c^p\neq c_j}^{K} w^p \cdot \mu_j(x^p)}{\sum_{p=1|c^p\neq c_j}^{K} w^p} \tag{8}$$

$$fitness^1(R_j) = w_P \cdot f_P(R_j) - w_N \cdot f_N(R_j) \tag{9}$$

$$fitness^2(R_j) = w_P \cdot f_P(R_j) + w_{NCP} \cdot NCP_{normal}(R_j) - w_N \cdot f_N(R_j) - w_{NMP} \cdot NMP_{normal}(R_j) \tag{10}$$

$$fitness^3(R_j) = w_{BF} \cdot fitness^2(R_j) - w_{LEN} \cdot length(R_j) \tag{11}$$

B. FAIS classifier

An overview of FAIS classifier is presented in Fig. 4. The main loop of algorithm is that the classifier learns each class separately. This loop consists of 4 steps: initialization, rule generation, rule learning, and termination test. Rule generation phase uses AIS algorithm to find a rule based on the initial population. In rule learning stage, when a rule is added to rule set, the learning mechanism reduces the weight of those training instances that are covered by the new rule. So, in the run of next rule generation, AIS focuses on those instances that are currently uncovered or misclassified [20]. Weight of each instance at start of the algorithm is assigned to 1. Each step is described in details as follows:

1) Initialization: In this stage, a population of B-cells is generated. The number of initial population is constant. This number is a parameter of FAIS which is named *intialPopulationSize*. For generating a B-cell, an instance of current class from data set is randomly selected, then fuzzy terms for antecedent of generated rule is computed from each attribute value of the instance. Consequent part of rule becomes the class of selected instance. Initial age of B-cell is another parameter denoted *defaultAge*. After generation of initial population, for each B-cell fitness is computed.

2) Rule Generation: In this step, a population of B-cells searches for optimized rule iteratively.

At first step of AIS, some B-cells are selected to be cloned. This selection is based on roulette-wheel selection algorithm. B-cells with higher fitness have more chance to be selected. The number of selection is constant. We define

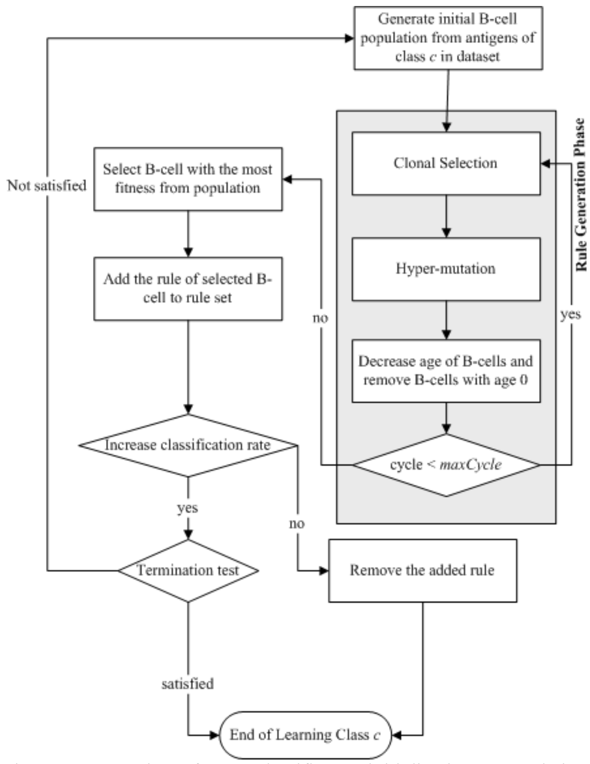

Fig. 4. An overview of FAIS classifier. At initialization, a population of B-cells is generated from instances of class *c*. then some B-cells are selected to proliferate in rule generation phase. The life cycle of B-cells is controlled by age. The best B-cell is added to rule set if the classification rate increases more than a threshold. At last, if the termination test satisfies the classifier learns rules for class *c*.

selectionSize parameter for the number.

Now it's time to proliferate for selected B-cells. Hyper-mutation is occurred during cloning process. A B-cell contains a rule and the rule has antecedents. A change to these antecedents cause change to B-cell which is mutation. Maximum number of simultaneous changes in antecedents of a B-cell is parameter which is named *maxTermChangesNumber*. We need to limit number of changes because modifying the most of antecedents of rule causes the change to the nature of rule. A B-cell is generated in a cycle, and then it clones and lives some time in population. In its life cycle it explores a small part of huge search space. When it mutates, it alters the current search path locally but when the nature of the B-cell is changed, it startles and cannot use its past knowledge. The probability of changing an attribute value to don't care is a parameter that is named *dontcareReplacementRate*. Number of clones produced for each B-cell is another parameter which is called *cloneNumber*. The age of generated B-cells is calculated from the equation (12). The formula controls the population size.

$$newAge = oldAge + defaultAge * fitness, \quad if\ newFitness > oldFitness \tag{12}$$

After cloning, parent B-cells (B-cells which is not produced in this iteration) are getting old. Some B-cells died because their age reaches to 0.

3) Rule Learning: When AIS algorithm is finished, from the final population, the best B-cell based on fitness is selected. The rule which is contained in B-cell is added to rule set. Then the classification rate of current rule set is compared to the old rule set which do not contain the new rule. Classification rate is calculated with equation (13). If the difference is higher than a threshold, the addition is accepted. The threshold is parameter with name *accuracyThreshold*.

$$classification\ rate = \frac{NCP}{number\ of\ patterns} \tag{13}$$

4) Termination Test: If a stopping condition is satisfied, the learning of current class is finished, and the algorithm is going to learn the next class. If the condition is not satisfied, the algorithm tries to learn another rule with initializing a population for next run of AIS. We can use any stopping condition for terminating the loop. We limit the number of learned rules for each class. This is done by a parameter which is called *maxRuleSetSize*.

C. *Reasoning Method*

After extracting rules, FAIS must use rules to predict the class of a test pattern. The usual reasoning method of fuzzy classifiers is done by equation (6) which is explained in section III. At first, we use equation (6) to predict but when all the rules are not applicable for the input pattern. The algorithm finds the most similar rule to this pattern. A rule is created from the instance like the initialization phase of FAIS primary rules are generated from instances. The most similar rule is the rule which has the longest common subsequence (LCS) with the newly generated rule. The length of common subsequence of the selected rule must be greater than minimum length. This value is another parameter which is named *minRuleSimilarityLength*. This method decreases the number of unclassified patterns of the algorithm.

TABLE I
UCI DATASETS USED IN OUR EXPERIMENTS

Dataset	# classes	# attributes	# instances	Classes
Australian	2	14	690	307 negative 383 positive
German	2	24	1000	700 negative 300 positive

Instances of the Australian are almost equally distributed between classes but in the German they are more unbalanced.

TABLE II
PARAMETER SPECIFICATION OF FAIS IN OUR EXPERIMENTS

Parameter	value in Australian	value in German
initialPopulationSize	100	300
maxIteration	50	50
defaultAge	5	5
selectionSize	100	100
cloneNumber	10	10
maxTermChangesNumber	3	2
dontCareReplacementRate	0.5	0.2
maxRuleSetSize	5	10
accuracyThreshold	0.03	0.03
maxRuleSimilarityLength	10	17
minCoveredPercemt	0	30
w_P	0.01	0.35
w_{NCP}	0.69	0.1
w_N	0.01	0.45
w_{NMP}	0.29	0.1
w_{BF}	0.8	0.999
w_{LEN}	0.2	0.001

V. EXPERIMENTAL RESULTS

In this section, two credit data sets were used to evaluate the predictive accuracy of FAIS classifier. Australian Credit Approval and German Credit Approval data sets are available from UCI Machine Learning Repository. In Australian credit approval data set, all names and values have been changed to meaningless to protect confidentiality of the data. Table I illustrates the information of these data sets. In Australian credit data there are 383 instances where assigning credit to them has high risk and 307 instances are creditworthy applicants. The German credit data is more unbalanced, and it consists of 300 instances where credit should not be assigned and 700 instances are creditworthy applicants. Because of simplicity in implementation of FAIS, each value in the Australian and German data sets is normalized between 0.0 and 1.0. Table II represents the parameter settings that have been used in simulations for FAIS. Simulations are done by Weka data mining software.

In Fig. 5 progress of classification rate per rule of FAIS for Australian credit data set and in Fig. 6 the progress for German credit data set is illustrated. The figures show the role of each fuzzy if-then rule that has been evolved by the proposed algorithm. FAIS uses iterative rule learning and it runs AIS per iteration to find a rule (rule generation phase). After the rule is extracted the weight of instances that covered by the rule is decreased. In FAIS the instances are removed from data set which means the weights are set to

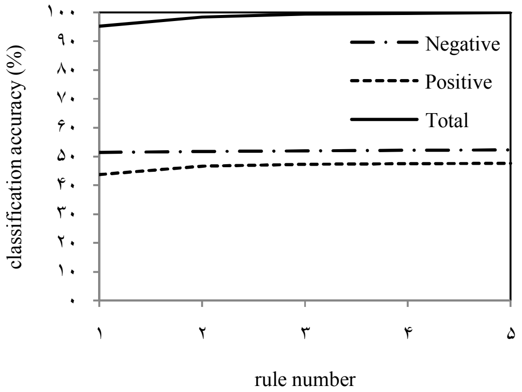

Fig. 5. Progress of classification rate per rule of FAIS for Australian credit data set

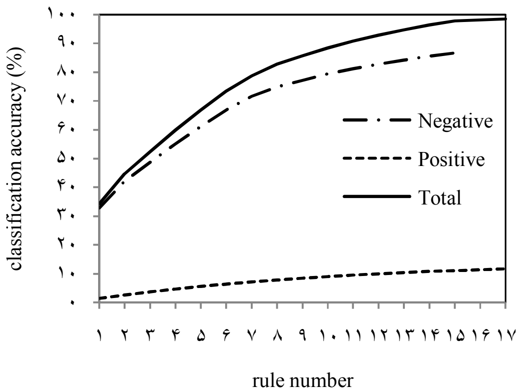

Fig. 6. Progress of classification rate per rule of FAIS for German credit data set

zero. According to figures, the first extracted rules are more general and shorter than later rules. All of the extracted rules participate in decision process and the rules classify almost the whole test data.

In addition to the mentioned explanations, according to the figures we can compare the complexity of data sets. 5 rules for each class have been extracted in Australian data set and in German, we had 12 rules for negative and 17 rules for positive class. The difference in number of extracted rules shows that German data set patterns are more complex than Australian data set. But the figures demonstrate the comparison in another way. According to Figs. 5 and 6, it is obvious that the increasing rate of classification accuracy for Negative class of the German dataset is much more than Negative class of the Australian dataset. Negative class in the German data set has a more complicated signature because the later rules have more effect than Australian data set (slope of the graph is very slow). This fact indicates that knowledge discovery for the instances of Australian data set is much more difficult than German data set. In Australian data set the first rules are very important rules because final classification accuracy is nearly equal to the classification accuracy at those points.

TABLE III
PREDICTIVE ACCURACY OF FAIS AND DIFFERENT CLASSIFIERS

Classifier	Australian	German
DMNBtext	82.58	69.99
DTNB	85.41	71.52
FAIS (this study)	**85.51**	**72**
LibSVM	**85.51**	70.98
LWL	**85.51**	70
Kstar	78.88	69.89
PART	83.32	70.11
SMO RBFKernel	**85.51**	70

The results are measured by Weka machine learning software. The order of classifiers is alphabetical. The most accurate is bold and the second most is underlined. Comparison of predictive accuracies illustrate FAIS is competitive with other classifiers.

In table I, we have demonstrated the distribution of instances over classes. In Australian data set they are approximately equal. Fig. 4 shows this fact too because the accuracy of classes are near each other. In German data set the number of negative class instances is more than instances of positive class. In Fig. 5 we have seen the important role of negative class in final classification accuracy. The extracted rules of positive class have covered very low test data instances.

TABLE IV
CONFUSION MATRIX

Predicted / Actual	Negative	Positive
Negative	TN	FP
Positive	FN	TP

We followed 10-fold cross validation procedure to evaluate the accuracy of FAIS. The classification rate of FAIS is measured with Weka machine learning software and compared with well-known classifiers in Weka including LibSVM, PART, DTNB, Kstar, LWL, DMNBtext, SMO with RBFKernel and J48. In these classifiers, DTNB and PART extract rules, J48 uses decision tree, LWL and Kstar are lazy classifiers, LibSVM and SMO are two implementations of SVM, DMNBtext uses Bayesian decision theory. As we mentioned in previous studies, AIRS is the classifier that were proposed by Leung [5] using AIS in credit scoring. Table III summarizes the prediction accuracies of FAIS and other classifiers.

Other performance measures for comparing FAIS with mentioned classifiers are precision, recall and f-measure. These measures can be obtained using equations (14) to (16).

$$recall^{NEG} = \frac{TN}{TN + FP} \quad recall^{POS} = \frac{TP}{TP + FN} \qquad (14)$$

$$precision^{NEG} = \frac{TN}{TN + FN} \qquad precision^{POS} = \frac{TP}{TP + FP} \qquad (15)$$

TABLE V
COMPARING PRECISION, RECALL AND F-MEASURE OF FAIS AND SELECTED CLASSIFIERS

Classifier	class	Australian pos.	Australian neg.	German pos.	German neg.
DMNBtext	precision	0.82	0.83	0.5	0.71
	recall	0.87	0.77	0.09	0.96
	f-measure	0.85	0.8	0.16	0.82
DTNB	precision	0.87	0.84	0.77	0.53
	recall	0.87	0.84	0.41	0.84
	f-measure	**0.87**	0.84	0.47	0.81
FAIS (this study)	precision	**0.93**	0.79	0.64	**0.80**
	recall	0.8	**0.93**	0.47	0.89
	f-measure	0.86	**0.85**	**0.54**	**0.84**
LibSVM	precision	**0.93**	0.79	**0.78**	0.71
	recall	0.8	**0.93**	0.05	0.99
	f-measure	0.86	**0.85**	0.09	0.83
LWL	precision	**0.93**	0.79	∞	0.7
	recall	0.8	**0.93**	0.0	**1.0**
	f-measure	0.86	**0.85**	∞	0.82
Kstar	precision	0.76	**0.85**	0.5	0.76
	recall	**0.91**	0.64	0.39	0.83
	f-measure	0.83	0.73	0.44	0.79
PART	precision	0.85	0.82	0.5	0.78
	recall	0.85	0.81	**0.49**	0.79
	f-measure	0.85	0.81	0.5	0.79
SMO RBFKernel	precision	**0.93**	0.79	∞	0.7
	recall	0.8	**0.93**	0.0	**1.0**
	f-measure	0.86	**0.85**	∞	0.82

The results are measured by Weka machine learning software. The order of classifiers is alphabetical. The best result is bold and the second most is underlined.

$$f\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \quad (16)$$

According to table V our proposed algorithm FAIS has the most value in f-measure in both positive and negative classes (except in positive class for Australian data set that the value is second most). F-measure is harmonic mean of precision and recall. The significance of precision and recall is dependent to the domain. In some application areas precision is more interested than recall and in some others recall is more important. But with f-measure both of precision and recall are kept high, so, FAIS is a reliable model for classification systems.

VI. CONCLUSION

In this paper, we propose a fuzzy classification system for credit scoring. Using fuzzy logic in AIS is presented in [14]. Credit scoring with AIS is described in [5,24]. The proposed classifier is combined fuzzy logic and AIS concepts. A new fitness formula for fuzzy classification is presented. In this formula, we use three terms, NCP for increasing the accuracy, NMP for decreasing error rate and rule length for increasing the comprehensibility of the knowledge. One of the most compromises in evolutionary algorithms like AIS is balancing between exploration and exploitation. In this algorithm, exploring is done by mutation of B-cells and exploiting is applied with age formula. In equation (12) after cloning when fitness is increased, the age is increased. The age formula controls the population

The future work of this study lies in improving the accuracy and performance of the algorithm. We can use other concepts in AIS like negative selection or immune network.

REFERENCES

[1] H.A. Abdou, "Genetic programming for credit scoring: The case of Egyptian public sector banks," *Expert Systems with Applications*, vol. 36, Nov. 2009, pp. 11402-11417.

[2] L.J. Mester, "What's the point of credit scoring?," *Business review*, vol. 3, 1997, pp. 3–16.

[3] Zhang Lei-lei, Hui Xiao-feng, and Wang Lei, "Application of adaptive support vector machines method in credit scoring," *Management Science and Engineering, 2009. ICMSE 2009. International Conference on*, 2009, pp. 1410-1415.

[4] S. Finlay, "Credit scoring for profitability objectives," *European Journal of Operational Research*, vol. 202, Apr. 2010, pp. 528-537.

[5] K. Leung, F. Cheong, and C. Cheong, "Consumer credit scoring using an artificial immune system algorithm," *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, 2007, pp. 3377–3384.

[6] V.S. Desai, J.N. Crook, G.A. Overstreet, and Jr, "A comparison of neural networks and linear scoring models in the credit union environment," *European Journal of Operational Research*.

[7] D. Zhang, M. Hifi, Q. Chen, and W. Ye, "A Hybrid Credit Scoring Model Based on Genetic Programming and Support Vector Machines," *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, 2008.

[8] J.E. Hunt and D.E. Cooke, "Learning using an artificial immune system," *Journal of Network and Computer Applications*, vol. 19, Apr. 1996, pp. 189-212.

[9] J. Timmis, A. Hone, T. Stibor, and E. Clark, "Theoretical advances in artificial immune systems," *Theoretical Computer Science*, vol. 403, Aug. 2008, pp. 11-32.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, 2009, pp. 10-18.

[11] C. Huang, M. Chen, and C. Wang, "Credit scoring with a data mining approach based on support vector machines," *Expert Systems with Applications*, vol. 33, Nov. 2007, pp. 847-856.

[12] M. Gadi, X. Wang, and A. do Lago, "Credit Card Fraud Detection with Artificial Immune System," *Artificial Immune Systems*, 2008, pp. 119-131.

[13] Xiaoguang Chang and J. Lilly, "Evolutionary design of a fuzzy classifier from data," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, 2004, pp. 1894-1906.

[14] Z. Lei and L. Ren-hou, "Designing of classifiers based on immune principles and fuzzy rules," *Information Sciences*, vol. 178, Apr. 2008, pp. 1836-1847.

[15] A. Asuncion and D. Newman, "UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]," 2007.

[16] J. Timmis and T. Knight, "Artificial Immune Systems: Using the Immune System as Inspiration for Data Mining," *Data Mining: A Heuristic Approach*, Group Idea Publishing, , pp. 209-230.

[17] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future," *Applied Soft Computing*, vol. 8, Jan. 2008, pp. 191-201.

[18] Xian Shen, X. Z. Gao, Rongfang Bie, and Xin Jin, "Artificial Immune Networks: Models and Applications," *Computational Intelligence and Security, 2006 International Conference on*, 2006, pp. 394-397.

[19] D. Dasgupta, "Advances in artificial immune systems," *Computational Intelligence Magazine, IEEE*, vol. 1, 2006, pp. 40-49.

[20] M. Abadeh, J. Habibi, M. Daneshi, M. Jalali, and M. Khezrzadeh, "Intrusion detection using a hybridization of evolutionary fuzzy systems and artificial immune systems," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 3547-3553.

[21] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based

classification systems," *Fuzzy Systems, IEEE Transactions on*, vol. 4, 1996, pp. 238-250.

[22] F. Hoffmann, "Combining boosting and evolutionary algorithms for learning of fuzzy classification rules," *Fuzzy Sets and Systems*, vol. 141, Jan. 2004, pp. 47-58.

[23] O. Cordón and F. Herrera, "A three-stage evolutionary process for learning descriptive and approximate fuzzy-logic-controller knowledge bases from examples," *International Journal of Approximate Reasoning*, vol. 17, Nov. 1997, pp. 369-407.

[24] A. Watkins, J. Timmis, and L. Boggess, "Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm," *Genetic Programming and Evolvable Machines*, vol. 5, 2004, pp. 291-317.

[25] J. Kelsey and J. Timmis, "Immune Inspired Somatic Contiguous Hypermutation for Function Optimisation," *Genetic and Evolutionary Computation — GECCO 2003*, 2003, p. 202.