IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 4- Number 4- September 2012 (55-67)

Artificial Immune Classifier (aiCLS): An Immune Inspired Supervised Machine Learning Method

S. Amir Ehsani
Faculty of Electrical, Computer and IT Engineering
Qazvin Islamic Azad University
Qazvin, Iran
amir@ehsani.ir

Amir Masoud Eftekhari Moghadam
Faculty of Electrical, Computer and IT Engineering
Qazvin Islamic Azad University
Qazvin, Iran
eftekhari@qazviniau.ac.ir

Received: March 20, 2012- Accepted: August 28, 2012

Abstract— **Artificial immune systems have been proven to be efficient in pattern recognition, data clustering and data classification. The proposed method is a novel artificial immune classifier called aiCLS based on aiNET. Artificial immune network (aiNET) is an efficient data analysis and clustering algorithm capable of clustering simple datasets through complex ones. Hidden capabilities of aiNET for supervised learning were significantly considered by aiCLS. The proposed method takes a local optimization approach to classification problem. It generates local optimum cells to recognize any given training antigen. Concatenation of these cells results in a global optimum classifier. The novelty of aiCLS has been discussed from both computational and immunological aspects. From the computational aspect, aiCLS is a fast one-shot learner algorithm with regard to the proposed "iterative clonal selection". From the immunological aspect, aiCLS introduces a novel clonal suppression method called "dissimilarity proportional clonal suppression (DPCS)", which increases data reduction and convergence to local optimum for any given antigen. DPCS alters convergence through a greedy suppression, which takes antibody-antigen affinity into account. The experimental results show that aiCLS outperforms artificial immune recognition system (AIRS) on UCI benchmark datasets in both classification accuracy and data reduction.**

Keywords-Component; Artificial Immune System; Immune Network; Classifier; Dissimilarity proportional clonal suppression; Supervised Learning

I. INTRODUCTION

Artificial immune systems are natural inspired soft computing paradigm. The immune system of vertebrates has been the source of this inspiration which aims at recognizing the pattern of pathogens (antigens) to limit the damage they may cause. Therefore, the concepts and metaphors of artificial immune system have been defined primarily based on pattern recognition (1,2). After that, artificial immune systems have been applied in a wide variety of domain areas, such as pattern recognition, classification, optimization, data analysis, computer security and robotics(3).

Artificial immune systems can be intuitively classified through their application domains and employed metaphors. From the application point of view, artificial immune systems have been used in a wide range of applications from network security to robotics (3). In this paper, the most relevant applications such as data analysis, optimization and classification are mentioned. CLONALG was one of the first machine learning and optimization algorithms which employed immune concepts (4). Then it was extended by employing immune network metaphor to aiNET, which is a data clustering and data analysis algorithm (5). Another immune clustering algorithm, called RLAIS (or RAIN), employs immune network theory (6). While aiNET models immune cells as antibody, RLAIS introduces an artificial concept called Artificial Recognition Ball (ARB). ARB is a set of B-Cells with identical antibody, in which the number of B-Cells determines antibody stimulation.

Due to the intrinsic association between data clustering and data classification and capabilities of RLAIS, AIRS has been developed as a data classification algorithm based on RLAIS (7). AIRS is a stable and robust classifier that produces around average results (8). The first version of AIRS did not apply the principle of dissimilarity proportional mutation (9); however, the recent version of the algorithm overcame its deficiencies. In addition, it removed unnecessary computational complexities while maintaining classification accuracy of the algorithm (10).

SAIS is another classifier algorithm based on immune concepts. It is different from other immune algorithms in many ways (11). SAIS models immune cells as B-Cell; but only one B-Cell is taken into account so that each class is modeled as a hyper sphere instead of multiple hyper spheres. SAIS ignores immune networks both similarity proportional cloning and dissimilarity proportional mutation. However, removal of these concepts and metaphors results in generation of a compact immune classifier which outperforms AIRS in half of in common datasets.

Classification application of artificial immune systems is not limited to general domain classifier algorithms such as AIRS and SAIS. There are some other domain specific classifier algorithms which have been introduced in literature (12,13,14). Although the aiNET algorithm is characterized as a clustering algorithm, it is also applied in classification problem. (15,16). The present authors examined the concerned aspects with regard to the significant lack of research in generating supervised learning algorithm based on aiNET.

This paper attempts to address the above-mentioned deficiencies. With regard to the capability of aiNET in modeling simple to complex clusters, it is considerable that aiNET can be used as a classifier algorithm. This work was motivated by previous researches of the authors. In (17) and (18), aiNET has been applied in classification problems though acceptable results have been achieved; however, it was still far from the optimum level and computational costs has been high to be a general domain classifier. Therefore, an extension of aiNET, named aiCLS is proposed as a general domain classifier algorithm.

The proposed artificial immune classifier (aiCLS) is a classifier algorithm, which adopts all immune metaphors and concepts employed in aiNET; however, it's different from aiNET in two aspects: 1)- the proposed method is a one-shot learning algorithm so that the learning process will be faster and there isn't more maximum iterations parameter; 2)- clonal suppression of aiNET replaced with a novel Dissimilarity Proportional Clonal Suppression (DPCS), which takes antigen-antibody affinity into account at clonal suppression process. DPCS is a population control mechanism that results in greater data reduction and greater efficiency.

The remainder of the paper is organized as follows: a summary of the related works is discussed in Section 2. Section 3 describes the proposed method. Section 4 presents experimental results and concluding remarks are given in Section 5.

II. Related Works

In this section, the most relevant works reported in the literature are first presented. This paper skipped the detailed discussion of natural immune concepts and interested readers can refer to (3) and (19) for more information on immune systems. This section starts with a brief review of both versions of AIRS algorithm, followed by the introduction of some artificial immune inspired classifiers. The section ends with the discussion of standard aiNET as a clustering algorithm and classifier aiNET used in the literature.

A. Artificial Immune Recognition System (AIRS)

AIRS is one of the first supervised artificial immune systems (7). The main algorithm revised in (10) and some unnecessary computational complexities were removed from it and the mutation subroutine changed to dissimilarity proportional mutation. These changes did not reduce efficiency of the first version of algorithm. Hence, it reduced computational complexity and increased data reduction in some datasets.

As a successor of RLAIS, AIRS models immune cells as Artificial Recognition Ball (ARB). Each ARB consists of an antibody (feature vector coupled with its class) and a scalar value which determines the number of resources. The number of resources is considered as a measure of fitness; higher number of resources means higher fitness. This fitness measurement is later used in a competition for resources to make a fitness proportional selection in population control mechanism.

The revised version of AIRS is different from that of the primary AIRS in two aspects; 1)- AIRS2 employs dissimilarity proportional mutation; 2)- some unnecessary computational complexities are removed from the algorithm, which include mutating class of ARB, complex stopping condition of cell proliferation et Figure 1 illustrates the summary of AIRS algorithm.

```
1. Initialization
2. For each antigen
2-1. ARB generation: select the best matching memory cell and generate ARB. Clone and mutate ARB.
2-2. Resource allocation: allocate resources to ARBs; number of resources is proportional to similarity (the closer they are to antigen, the more the resources)
2-3. Resource competition: if the allocated resources are more than available resources, remove resources from the least similar ARB. ARBs with no resources will be deleted.
2-4. Cell proliferation: Clone ARBs proportional to similarity and mutate them proportional to dissimilarity.
2-5. Stopping condition: If ARBs' similarity to antigen has not met certain criteria, repeat 2-2 to 2-5.
2-6. Memory cell introduction: if new ARB is closer to antigen than the best matching cell, add a new ARB to memory cells set; if similarity of the new ARB and the best matching cell is more than a certain threshold, the best matching cell removes the best matching cell.
```

Figure 1. Summary of AIRS2 algorithm

B. Other Artificial Immune Inspired Classifiers

Simple Artificial Immune System (SAIS) is a novel immune inspired supervised learning algorithm. Almost none of the immune metaphors have been employed by SAIS. However, its classification accuracy is more than AIRS in some cases (11). This algorithm has a global optimization approach for the classification problem. In other words, instead of generating optimized B-cell for each antigen, SAIS generates a B-cell for the whole class. So, it models the whole class as a single hyper sphere and reduces classification accuracy on complex datasets in comparison with algorithms such as AIRS, which model the class as multiple hyper pheres.

Attribute Weighted Artificial Immune System (AWAIS) is another immune inspired classifier. AWAIS employs more immune metaphors than SAIS but still it does not take the immune network into account (20). The results reported for this algorithm in literature are limited to Wine dataset. So, one cannot judge its classification capabilities.

The above named methods are general domain classifiers, which employ immune metaphors. Many domain specific classifiers have been also reported in the literature, such as remote sensing image classifiers (12), spam detection (13), anomaly detection (21) and so on.

C. Artificial Immune Network (aiNET)

Artificial Immune Network (aiNET) is a data clustering algorithm (5) extended from CLONALG (4). Network theory is employed in aiNET to generate a population control mechanism, which is based on limiting the minimum distance of any antibody pair. The output of aiNET is an edge-weighted graph composed of a set of nodes and a setoff edges. Each node is an antibody and each edge has a weight. Weight is affinity (similarity) between the node pair (5). Due to employing network theory population control mechanism, aiNET can be used to cluster a wide variety of datasets from simple to complex. The key parameter of this population control is suppression threshold (σ_s). The suppression threshold controls the specificity of cells' smaller value of $\sigma_{s,}$ makes more antigen specific cells and larger value makes generalized cells. Figure 2 is the summary of aiNET algorithm.

Standard aiNET is a clustering algorithm but it has been used as a classification algorithm, too. In (15), two aiNETs have been used to discover clusters in two classes of cancer outcome prediction. In terms of immunology, nodes of aiNET are internal images of antigens (training data). Therefore, clustering each class's training data using aiNET results in the generation of internal images of training data. Internal images could be defined as a compressed form of training data. Internal images preserve shape of cluster and the number of cells in internal images is less than antigens (training data). After the training process, internal images (also, known as memory cells) are ready to classify test data. Classification could be done through a variety of methods.

III. Artificial Immune Classifier (aiCLS)

The proposed method is an extension and modification of aiNET. In other words, aiCLS is an aiNET, which optimized for supervised learning and classification problem. These modifications reduces computational complexity of aiNET and preserves all the concepts and metaphors employed in aiNET, such as immune network, mutation proportional to dissimilarity and similarity proportional cloning.

```
1. Randomly generate initial antibodies
2. While stopping condition didn't met
2-1. For each antigen
2-1-1. Calculate affinity (similarity) of all antibodies and antigen and select n best matches
2-1-2. Affinity maturation: clone proportional to affinity (similarity), mutate proportional to distance (dissimilarity)
2-1-3. Calculate affinity of new antibodies and antigen
2-1-4. Natural death: eliminate cells whose affinity is interior to Natural Death Threshold ($\sigma_d$)
2-1-5. Calculate Ab-Ab affinity and eliminate self-recognize cells (affinity < $\sigma_s$)
2-1-6. Add the remaining antibodies to the network
2-2. Calculate affinity of each two memory cells in network and eliminate self-recognize cells (affinity < $\sigma_s$)
2-3. Reproduce some random cells and add them to the network
```

Figure 2. Summary of aiNET algorithm

There are two main different aspects between aiNET and aiCLS: 1)- aiCLS is a one-shot algorithm and is faster than aiNET; 2)- aiCLS employs a novel clonal suppression method called *Dissimilarity Proportional Clonal Suppression* to increase efficiency of algorithm and for rapid convergence.

It is clear that just a single pass through the data does not guarantee the generation of optimal cells to classify or cluster the data (11). There are two different approaches to optimal cell generation. The most popular approach is to process data consequent iterations for a certain number. This approach has been used by aiNET. It randomly generates memory cells before the first iteration and then, during the first iteration, affinity maturation process generates closer memory cells to antigen. Memory cells generated in the first iteration will be used in the following iteration as initial memory cells so that any iteration slightly increases affinity of memory cells and this process continues until a certain stopping condition, such as the maximum number of iterations, is met.

Another approach is a one-shot learning (10), meaning that the algorithm passes any selected antigen to the training process just once; however, it is not the same as the one-pass. In this approach, training for an antigen is kept on consequently until a certain stopping condition is met. The stopping condition of these methods is a certain affinity criterion, not the maximum number of iterations. This process led the immune cells to the local optimum for each antigen. Therefore a population control mechanism was needed to suppress redundant immune cells while maintaining optimum immune cells required recognizing antigen.

If suppressing subroutines of the algorithm maintains the optimum cells found for an antigen through learning process for other antigens, the algorithm results in the optimum memory cells. However, if it eliminates the immune cells of an antigen without any appropriate replacement, it results in biasing for the latest antigens. However, computational complexity of the one-shot approach is less than iterative approach if and only if the iterations needing the satisfied stopping condition of affinity maturation are less than the *maximum iterations* of iterative algorithm.

The proposed method is a one-shot learning algorithm. Since network suppression of aiNET applied once for *iteration*, after training on all antigens, aiCLS cannot employ it as a main population control mechanism. Hence, in the one-shot learning, network suppression proceeds just once at the final stage of algorithm. Therefore, newly generated cells are accumulated on the immune network and size of network rapidly increases during the clonal selection process. Therefore, in order to control population and maintaining diversity and local optimum cells generated through affinity maturation, a novel suppression method has been proposed.

Immune network theory declares two stimulating terms and one suppressing term. Recognizing antigen and other antibodies stimulates antibody and being recognized by other antibodies which suppress it (3). Clonal suppression of aiNET is based on antibody-antibody affinity (5). Although all antibodies, which survive clonal selection, can recognize antigen, their antigen affinity is not the same. Therefore, aiNET may eliminate most stimulated antibody which leads to ignoring antibody-antigen affinity in clonal suppress.

Dissimilarity proportional clonal suppression (DPCS), proposed as a population control mechanism in this paper, takes both antibody-antibody affinity and antibody-antigen affinity into account. For any two antibodies that recognize each other, DPCS eliminates antibody with lower affinity to antigen. In addition, to prevent exponential growth of immune network, the best matching cells selected as the initial cells for the current antigen are moved from immune network to clone before clonal suppression. This process is a *local network suppression* that is included in DPCS. DPCS will be explained in Section 3-2.

The proposed algorithm can be thought as a five stage algorithm. Data normalization and generation of seed cells takes place in the initialization stage. The second stage generates initial antibodies. These initial antibodies go through *iterative clonal selection* to generate optimum antibodies in stage three. These stages are followed by *dissimilarity proportional clonal suppression* in stage four. Whenever these four stages are achieved for all antigens, network suppression, as the final stage, eliminates self-recognizing memory cells.

Once the training process is completed, the memory cells will be available to classify test data. The class of most similar memory cell to the test data will be selected as a class of test data. Figure 3 illustrates the summary of the training process of aiCLS.

A. Definitions

In this section, a brief definition of concepts and metaphors of immune systems are introduced. In addition, Table I shows conceptual association between concepts of immune systems and aiCLS.

Antibody: B-Cell receptor, Y-shaped receptor molecules bounds on the surface of a B-Cell with the primary role of recognizing antigen of Pathogens.

Antigen: A small portion of pathogens, molecules which should be recognized by immune system.

Affinity: Degree of binding of an antibody to an antigen. Affinity has been declared as both similarity and dissimilarity in the literature. However, in this paper, affinity was declared as similarity and distance is measurement of dissimilarity. By the way, to avoid confusion, similarity and dissimilarity have been used frequently instead of affinity and distance, respectively.

Clonal Selection: Process of selecting the best matching cells, cloning, somatic hyper mutation and selecting surviving cells

Somatic hyper mutation: High probability mutation proportional to dissimilarity

Affinity maturation: Mutation followed by survival selection

```
1. Initialization and data normalization
2. For each antigen
2-1. Generate initial antibodies
2-2. Generate optimum antibody through iterative clonal selection
2-3. Dissimilarity proportional clonal suppression
3. Network suppression
```

Figure 3. Summary of aiCLS algorithm

TABLE I. MAPPING BETWEEN AICLS AND IMMUNE SYSTEM CONCEPTS

Immune System	The Proposed Method
Antibody	Combination of feature vector and its affinity to the currently processing antigen
Antigen	Set of feature vectors of training data
Distance	Euclidean distance between antibody and antigen
Affinity	1 – Distance
Memory cell	Combination of feature vector and its relevant class

B. Overview of aiCLS

The proposed method consists of five main stages; each stage consists of some detailed steps. Figure 3 shows main stages of aiCLS algorithm while Figure 4 illustrates all steps of the algorithm in detail. The remainder of this section describes main stages in more detail; then, followed by description of data classification is done using aiCLS.

1) Initialization

The primary step of algorithm is data normalization. Normalization method highly depends on distance measurement. Distance of any two normalized vectors has to be between 0 and 1 inclusively and none of attributes of feature vectors may exceed the range [0, 1]. In this paper, Euclidean distance has been used as dissimilarity measurement; therefore, the following equation was employed to normalize the data:

$$F_i = \frac{f_i - Min_i}{Max_i - Min_i} \qquad (1)$$

$$G_i = f_i \Big/ \sqrt{N} \qquad (2)$$

where F is an N dimensional feature vector consisting of ($f_{1,}$ f_2, …, f_N) and Max_i and Min_i are possible maximum and minimum values of i-th feature in F, respectively.

Once normalization is done, generation of initial antibodies can be preformed. Although it is an optional step and could be ignored by setting *Seed Count* to zero, *Seed Count* is the parameter that determines the quantity of randomly selected antigens and their class attribute should be passed to memory cells in the initial step.

```
1. Data normalization: Distance of each two antigens should be within [0, 1].
2. Generating seed cells: add a certain number (SeedCount) of randomly chosen antigens to memory cell set (M).
3. For each antigen (Ag)
3-1. antibody set (AB) Generation
3-1-1. Select all memory cells from M where class of M is equal to class of Ag and add it to $M_{Active}$.
3-1-2. If $M_{Active}$ is empty, add Ag to AB. Else, for each m in M, generate an Ab where feature vector of Ab is equal to feature vector of M and affinity of Ab is zero; then, insert it to AB set.
3-1-3. Calculate and update affinity of each Ab in AB.
3-2. Select n best matching cells with the maximum affinity from AB and add them to BestMatches and $MC_{best}$.
3-3. Clone best matches proportional to affinity and add new cells to clone set.
3-4. Mutate clone proportional to distance.
3-5. Calculate and update affinity of each Ab in clone.
3-6. Eliminate each Ab in clone where affinity of Ab is less than Natural Death Threshold ($\sigma_d$).
3-7. Select n best matching cells with the maximum affinity from clone and add them to Best Matches to be used in the next iteration of clonal selection.
3-8. While Average Affinity of Clone is less than Affinity Threshold ($T_A$), repeat 3-3 to 3-8.
3-9. Add $MC_{best}$ to clone.
3-10. Eliminate memory cells of $MC_{best}$ from M
3-11. Calculate distance of each two Ab in clone.
3-12. Dissimilarity proportional clonal suppression: for $Ab_i$ and $Ab_j$ ($i \neq j$) in clone; if distance of $Ab_i$ and $Ab_j$ is less than the Suppression Threshold ($\sigma_s$), eliminate Ab with the minimum affinity to antigen.
3-13. For each Ab in clone, generate m where antibody of m equals to Ab and class of m equals to class of Ag, and add m to M
4. Calculate distance of each $m_i$ and $m_j$ ($i \neq j$) in M where class of $m_i$ and $m_j$ is equal.
5. Network suppression: eliminate memory cells with less than suppression threshold ($\sigma_s$) distance.
```

Figure 4. aiCLS Algorithm

Initialization of seed cells also could be performed by using randomly generated memory cells; but, these cells have to be in the range of normalized vectors.

2) Initial Antibody Generation

After initialization, training process begins from the first antigen to the last one. In the very first step of training process, all memory cells of the class concerned with antigen are selected as active memory cells. Antibodies of active memory cells are collected in antibody set (AB). If there is no active memory cell, initialization process adds the antigen as the internal image of antigen to antibody set. At this point AB is a set of antibodies, the *affinity* of which attribute is zero. Thus, affinity calculation takes place here; affinity of each antibody in AB with antigen is calculated to update *affinity* attribute of antibody.

Note that, there is no antibody pool in aiCLS. Therefore, aiCLS has no need to maintain the class of antibodies because living antibodies are generated for the current antigen and their class is the same as the class of antigen.

3) Iterative Clonal Selection

The heart of aiCLS is *iterative clonal selection*. As mentioned earlier, aiCLS is a one-shot learner. Therefore, in one pass of algorithm, aiCLS has to generate a set of high affinity antibodies for any given antigen. So, regular clonal selection is extended to iterative clonal selection, which continues affinity maturation until meeting a certain stopping criterion. Iterative clonal selection leads antibodies to a local optimum solution for the antigen.

The proposed clonal selection is very similar to the clonal selection used in [4] and [5]. Iterative clonal selection starts with the selection of the best matches for antigen. In other words, N antibodies with higher affinity to antigen are selected. These selected antibodies are called *best matches*. Since set *of best matches* is supposed to change during iterative clonal selection, its antibodies are also maintained in another set, named MC_{best} , for further use in DPCS. The next step clones *best matches*, proportional to antigen affinity using Equation 3. The higher cell affinity causes a larger number of clones. Since affinity is in the range [0, 1], the number of clones is in the range 0 and *clone rate* parameter, inclusively.

$$NumClone = CloneRate - Affinity(Ab, Ag) * CloneRate \quad (3)$$

Step 3-4 of the algorithm mutates any antibody in clone. Mutation is done according to the directed mutation introduced in (5), which is shown in Equation 4; where Ab_m is mutated antibody, Ab is antibody, Ag is antigen and α is mutation rate calculated by Equation 5.

$$Ab_m = Ab - \alpha(Ab - Ag) \quad (4)$$

$$\alpha = Rand * Dist(Ab, Ag) * M \quad (5)$$

In Equation 5, *Rand* is a uniform random number between 0 and 1, *Dist* is the Euclidean distance and *M* is hyper mutation rate or learning rate. The default value of hyper mutation rate in aiNET is 4.0 and, as a successor of aiNET, this value is adopted here. The lower and higher values of hyper mutation rate reduce and increase convergence speed, respectively. But, the increasing rate of hyper mutation may result in divergence.

Directed mutation is a dissimilarity proportional mutation. Therefore, for high affinity antibodies, α is small and, since cloning is similarity proportional, a large number of clones are generated. Thus, many antibodies are generated and each of them slightly changes during mutation. Hence, exploitation is done through cloning and mutation. In addition, for low affinity antibodies, α is large and a small number of clones that have been generated. Therefore, clonal selection results in exploration; since there are a small number of exploring cells, it would not lead population to divergence. The next step calculates antibody-antigen affinity for new antibodies and sets *affinity* attribute of the new antibodies.

The following step in the standard clonal selection is survival selection, which selects a percentage of high affinity cells to survive and eliminate others. This step is removed from iterative clonal selection to maintain more diversity. Therefore, after affinity calculation, natural death eliminates antibodies, whose affinity is less than *natural death threshold* (σ_d). Natural death is the last step in the standard clonal selection; however, there are two more steps in the iterative clonal selection. First of all, the *best matches* are selected again from the antibodies remaining in the clone to be used in the next iteration of clonal selection; then, stopping criteria are considered. If average affinity of antibodies in clone is more than or equal to *affinity threshold* (T_A), learning on this antigen is complete and the next step will be Step 3-9. If stopping condition did not occur, clonal selection process would be continued and Step 3-3 to Step 3-8 would be repeated; but this time, *the best matches* are the cells selected from clone in Step 3-7. Therefore, in each iteration of the clonal selection, initial cells are the best cells of previous iteration. Therefore, iterative clonal selection leads antibodies to a rapid greedy local optimum convergence..

Affinity threshold (T_A) determines how high affinity of antibodies should be after iterative clonal selection. So, this parameter has a significant effect on efficiency of algorithm. While 4-2 discusses the effect of this parameter in detail, it should be noted here that the optimum value for this parameter is in the range [0.9, 0.98] for all the tested datasets.

4) Dissimilarity Proportional Clonal Suppression

Population control is a challenge for evolutionary algorithms including artificial immune systems. Two population controls are used in artificial immune systems. The mechanism used by (6) and (10) is a ranking system based on artificial recognition balls (ARBs) to eliminate low affinity cells. ARBs could be thought as a feature vector and its rank in clone so that the competition between ARBs in AIRS and RAIN is the population control mechanism used to eliminate low rank cells. By the way, AIRS eliminates all cells but one in clonal selection. So, it reduces diversity. Another well-known approach to this challenge is the elimination of self-recognized cells. The aiNET algorithm removes all cells, which distance from another cell is less than a certain threshold. This method preserves diversity but it may eliminate high affinity cells.

Dissimilarity proportional clonal suppression (DPCS) is a diversity preserving population control mechanism, which eliminates low affinity antibodies. DPCS eliminates low affinity antibodies recognized by a higher affinity antibody. Therefore, high affinity antibodies and low affinity antibodies, which are not recognized by other antibodies, may survive clonal suppression process. Hence, both exploration and exploitation are possible at DPCS.

As was mentioned before, in order to prevent from the rapid increment of Immune Network during the learning process, DPCS is responsible for local network suppression, too. This is done by eliminating the best matching memory cells (MC_{best}) from network and concatenating them in clone. So, these cells have to join to antibody competition in order to stay in the memory cell pool.

The primary step of DPCS is to concatenate clone cells and MC_{best}. Then, DPCS eliminates MC_{best} from memory cells' pool. The next step is the calculation of distance of any antibody pairs. After that, for any antibody pair, whose distance is less than *suppression threshold* (σ_s), DPCS eliminates antibody with lower affinity from antigen. Therefore, the best one of self recognizing group of antibodies survives. After that, Step 3-13 generates a memory cell from surviving antibodies and adds it to memory cells' pool.

Here, the effects of local network suppression of DPCS are mentioned. There are two possible scenarios: 1)- consider that, MC_{best} has low affinity to antigen. In this case, new antibodies have high affinity to antigen (regarding the iterative clonal selection). Therefore, new antibodies are close to antigen and far from MC_{best} and MC_{best} causes DPCS to survive. Therefore, DPCS would not lead cells for biasing new data; 2) let both of new antibodies and the best matches have high affinity to antigen. In this case, the best one survives.

DPCS is the last stage in learning process for an antigen. When DPCS is completed, aiCLS starts again for the next antigen. If there is no more antigens, network suppression reduces the size of network as the final step of learning process of aiCLS.

5) *Network Suppression*

With regard to the local network suppression applied for each antigen in DPCS, network suppression of aiCLS is not a mass elimination of cells. However, network suppression consists of two steps: the first one is the calculation of inter cell distance of memory cells which its concerned class is the same. After that, for any of the same class memory cell pairs, the distance is less than *the suppression threshold* (σ_s). Network suppression eliminates one of them randomly. Therefore, one cell out of a group of self-recognizing cells survives (randomly selected). Note that elimination of all self-recognizing cells results in losing the efficiency of classifier because removing all of them results in forgetting the corresponding antigens.

6) *Classifying Input/Test Data*

Once the training process is done, classification is a straightforward task. After calculating affinity of all memory cells in memory cell pool and input data, the class of memory cell with the highest affinity to input data is the output of classification algorithm.

C. *Comparative Analysis of Artificial Immune Systems*

This section presents a summary of the proposed method and other artificial immune systems in a table. Table II presents comparative analysis of the concepts of some artificial immune systems.

IV. EXPERIMENTAL RESULTS

This section introduces experimental results of the proposed method. In order to compare aiCLS with other classifiers, a variety of machine learning datasets were applied. The remainder of this section is organized as follows. Subsection 1 describes test bed of the algorithm. Subsection 2 analyses parameters of algorithm from the viewpoint of efficiency and data reduction. After that, the comparative analysis of data

TABLE II. COMPARATIVE ANALYSIS OF SOME AIS SYSTEMS

Immune System	aiCLS	aiNET	AIRS	AIRS2	SAIS
Application Domain	Classification	Clustering	Classification	Classification	Classification
Immune Network	Yes	Yes	Yes	Yes	No
Clonal selection	Yes	Yes	Yes	Yes	No
Immune cell model	Antibody	Antibody	ARB	ARB	B-Cell
Immune cell concept	Feature vector and affinity	Feature vector	Feature vector, class and rank	Feature vector, class and rank	Multiple feature vector
Memory cell concept	Feature vector and class	Feature vector	Feature vector and class	Feature vector and class	Multiple feature vector
Number of clones	Similarity proportional	Similarity proportional	Similarity proportional	Similarity proportional	Constant value
Mutation	Dissimilarity proportional	Dissimilarity proportional	Uniform random	Dissimilarity proportional	Uniform random
Population control	Ag-Ab affinity, Ab-Ab affinity	Ab-Ab affinity	Ab-Ag ranking, Ab-Ab affinity	Ab-Ag ranking, Ab-Ab affinity	N/A
One-shot	Yes	No	Yes	Yes	No
Shape of each class	Multiple hypersphere	Multiple hypersphere	Multiple hypersphere	Multiple hypersphere	Single hypersphere

Classification accuracy is introduced in Subsection 3. Followed by that, brief discussion on the comparative analysis of data reduction and computation complexity is given in Subsections 4 and 5.

A. *Test Bed*

In order to determine the classification performance of aiCLS, five benchmark datasets from UCI repository were applied (22). These five datasets were Wine, Iris, Sonar, Ionosphere and Pima Indians Diabetes. Specifications of all datasets are shown in Table III. Table IV demonstrates optimum parameters used in the experimental results. At the end of this text, if the value of a parameter was not mentioned explicitly, the value form Table IV was used.

Benchmark datasets of test bed were different in many aspects; number of classes, training method, etc. Hence, it was a challenge for aiCLS. Sonar is a high dimensional dataset, which was selected to assess the effect of curse of dimensionality. Iris have three classes, one of which is linearly separable and the other two are not linearly separable; so, performance of the proposed method on this dataset determined capabilities of the method for classification of non-linear classes. Wine is not a challenging dataset and it is usually used for initial tests of novel classification methods.

There are standard training and testing methods for benchmark datasets. Four out of five datasets, which were mentioned here, should be examined using K-Fold cross validation. In a k-fold cross validation, instances of dataset are divided into k disjoint sets. Then, the concatenation of k-1 set is used as a training set and another set is used to test the trained classifier. Therefore, there were K training-test sets, each of which would be the test set once. The average of classification accuracy of all K training sets was considered as a result. The results reported in this paper are average of three given runs.

B. *Analysis of Effect of Parameters*

This subsection discusses the effect of parameters of algorithm. *Affinity threshold* and *suppression threshold* are the most effective parameters of aiCLS. Therefore, 4-2-1 and 4-2-2 analyze effect of these parameters and determines how their changes alter the performance of classification. Then, 4-2-3 briefly discusses other parameters.

1) *Affinity Threshold Analysis*

Affinity threshold (T_A) is the parameter that determines the time immune response is triggered. Whenever the average affinity of antibodies in clone exceeds T_A, *iterative clonal selection* is completed and immune response begins.

Considering the facts about T_A, theoretically, it can be concluded that smaller values of T_A convert aiCLS to a one pass algorithm and significantly reduce efficiency. Figure 5 illustrates classification accuracy and data reduction of aiCLS while T_A changes from very low value (0.5) through high (0.99);

TABLE III. SPECIFICATION OF BENCHMARK DATASETS

Dataset	Wine	Iris	Sonar	Diabetes	Ionosphere
Size of feature vector	13	4	60	8	34
Classes	3	3	2	2	2
Instances	178	150	208	768	351
Training Standard	10 fold	5 fold	13 fold	10 fold	First 200 as train set, remainder as test set

Table V shows both data reduction and classification accuracy for T_A in the range of [0.8,0.99].

Figure 5 indicates divergence for small values of *affinity threshold*. Whenever aiCLS fails to converge, it achieves low classification accuracy and low data reduction. Small values of T_A, causes early satisfaction of stopping criteria of *iterative clonal selection* and generation of low affinity antibodies. Ionosphere for T_A equals 0.8, results a good sample of divergence of aiCLS.. Over 4500 memory cells were generated and the accuracy was 90% while the best obtained consequence was 126 memory cells and 95.8% accuracy achieved for T_A was equal to 0.95.

The graphs of Figure 5 can be divided to three parts. The first part of the graph is where increment of T_A results in slightly data reduction and almost no improvement in accuracy (e.g. from 0.5 through 0.8 on Wine); the average number of execution of *iterative clonal selection* is close to one at these values. This part of graph demonstrates divergence. After that, there is always a jump of accuracy in graph and, at the same time, a big step in data reduction. This part indicates start of convergence; after that, there will be slight improvement in accuracy and almost no change in data reduction of algorithm or slight increment of data compared with the lower values of T_A. Until the third part of the graph, which shows up only for some datasets. Datasets such as Ionosphere and Iris, have a significant decrease in accuracy in the last steps (T_A = 0.99 for Iris). Therefore, the learner is over fitted. The classifier is too specialized for training data and loses its generalization for input data.

TABLE IV. OPTIMUM PARAMETERS OF AICLS ALGORITHM FOR BENCHMARK DATASETS

Dataset	Wine	Iris	Sonar	Diabetes	Ionosphere
Number of best matches (N)	10	5	1	1	7
Affinity Threshold (T_A)	0.95	0.97	0.9	0.96	0.95
Suppression Threshold (σ_s)	0.08	0.06	0.04	0.05	0.07
Clonal Rate	10	10	10	10	10
Natural Death Threshold (σ_d)	0.5	0.5	0.5	0.5	0.5
Number of Seed Cells	0	0	0	0	0

TABLE V. EFFECT OF AFFINITY THRESHOLD ON MEMORY CELL COUNT (M) AND CLASSIFICATION ACCURACY (P)

T_A	0.8		0.9		0.95		0.97		0.99	
Dataset	M	P	M	P	M	P	M	P	M	P
Wine	112	90.3%	49.3	95.0%	**66.4**	**98.2%**	82.2	97.8%	109.8	98.1%
Iris	48.13	95.4%	24. 3	95.1%	26.9	97.3%	**29.9**	**98%**	35.8	96.1%
Sonar	419.7	90.4%	**129.6**	**92.7%**	151.6	93.7%	167.9	93.7%	182.7	93.9%
Diabetes	157	74.9%	109.8	75.2%	215.3	77.4%	259.7	78.3%	468.9	79.8%
Ionosphere	4531	90.1%	414.6	91.7%	**126**	**95.8%**	127	95.2%	136.3	94.5%

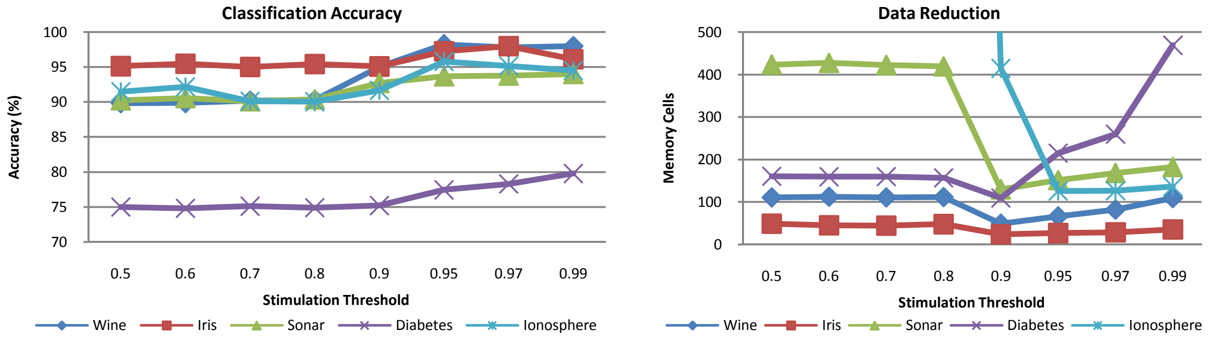

Figure 5. Effect of affinity threshold: classification accuracy and data reduction

It should be noted here low values of *affinity threshold* and *natural death threshold* depend on each other in some aspects. Let values of T_A and σ_d be equal to 0.5. Thus, natural death eliminates all antibodies with the affinity less than 0.5; then, *iterative clonal selection* calculates average affinity of clone. Therefore, if there is a living antibody, stopping condition is met although antibodies still have low affinity. More discussion of σ_d is held in 4-2-3.

While the relationship of *affinity threshold* and *natural death threshold* is the cause of divergence of the first part of graphs in Figure 5, the relationship between high values of *affinity threshold* and *suppression threshold* causes over-fitting of classifier. High *affinity threshold* reduces diversity of population and generates self-recognizing antibodies, combined with a high value of σ_s ; all of these cells are eliminated and only the best one survives. This process is repeated for the whole training antigens and the result of training process is a set of antibodies, which is too specialized to antigens and has no generalization. Therefore, the generated classifier is over-fitted and memory cells are increased in number. A good example of the over-fitted classifier is Sonar dataset when T_A is 0.99. In this case, classification accuracy on training data was 100% while, for test data, it was less than 88% for an average of three runs.

As a result of the experimental results and normalized data, the best values for T_A were in the range [0.9, 0.98] while the value of natural death stayed in [0.2, 0.6] and suppression threshold was in the range of *(1-T_A)* to *2*(1-T_A)*, inclusively.

2) *Suppression Threshold Analysis*

Both network suppression and clonal suppression are based on *suppression threshold* parameter. Therefore, this parameter is the key in population control mechanism. *Suppression threshold* could be defined as the minimum distance of any given antibody pair. Hence, lower values of σ_s result in smaller recognition regions and generation of more memory cells and vice versa; larger σ_s reduces the number of memory cells and increases recognition region. Therefore, small σ_s makes memory cells more localized and, similar to the antigen and its larger values, it makes memory cells more generalized. Localization leads the classifier toward over-fitting and generalization reduces classification accuracy on the overlapped complex classes. Therefore, there is a trade-off between number of memory cells and classification accuracy.

TABLE VI. EFFECT OF SUPPRESSION THRESHOLD (Σ_S) ON MEMORY CELL COUNT (M) AND CLASSIFICATION ACCURACY (P)

σ_s	0.4		0.5		0.6		0.7		0.8	
Dataset	M	P	M	P	M	P	M	P	M	P
Wine	205.3	97.9%	154.3	98.1%	117	98.3%	94.7	98.2%	**80.93**	**98.2%**
Iris	45	97.5%	36.4	97.9%	**28.9**	**98%**	23	27.8%	20.2	96.1%
Sonar	**129.6**	**92.7%**	116.3	92.1%	114.3	91.2%	103.4	90.3%	94.5	89.6%
Diabetes	324.6	80.0%	**259.7**	**78.3%**	202.6	74.4%	163.1	73.4%	131.3	72.5%
Ionosphere	186.3	96.3%	168	96.5%	141.3	96.1%	**126.6**	**95.8%**	115.6	92.9%

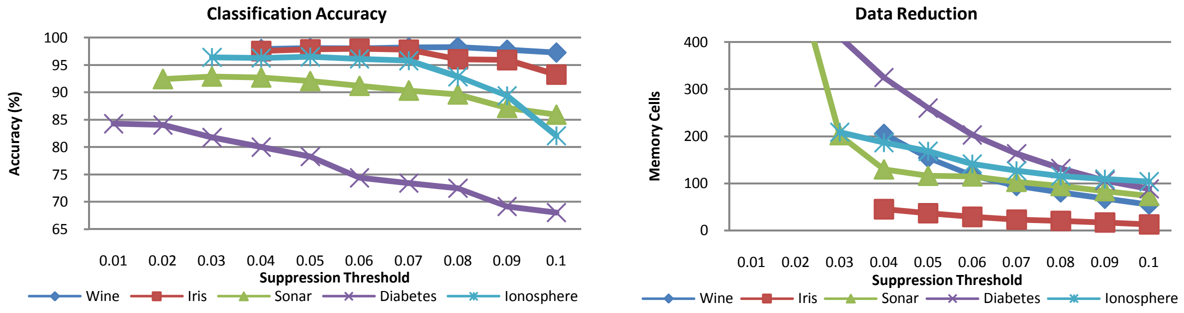

Figure 6. Effect of suppression threshold (σ_s) on classification accuracy and data reduction

Figure 6 shows the effect of this parameter on classification accuracy and data reduction and Table VII illustrates numeric values of accuracy and data reduction in the range of [0.4, 0.8].

Table VI gives the trade-off between data reduction and classification accuracy. It can be observed that increment of *suppression threshold* always increases data reduction; however, increment of classification accuracy is not guaranteed.

1) *Analysis of Other Parameters*

Here, effectiveness of other parameters of aiCLS is discussed. Parameters such as *clonal rate*, *number of best matches* (*N*), *number of seed cells* and *natural death threshold* (σ_s), which have lower dependence on input data were compared with the previously discussed parameters. Therefore, the same values were used for these parameters for all datasets, except for one of them, which is *number of best matches* (*N*).

Number of best matches and *clonal rate* were both effective in computational complexity of the algorithm through altering the number of clones generated in the clonal selection. Increment of the number of clones increased computational complexity of clonal selection; however, in addition, population converged faster. Due to the experimental results, the values less than five for clonal rate resulted in the divergence or reduction of convergence speed since it reduced size of population and led to the lack of exploitation and exploration.

Number of best matches was also effective in *dissimilarity proportional clonal selection*. It was mentioned that this parameter was not completely independent from input data. That was because of the dependence of the parameter on *suppression threshold*, which was a data-dependent parameter. Small values of *suppression threshold* limited the minimum distance of antibodies; therefore, larger values of *N* increased the probability of biasing to earlier antigens. Although the parameter had no significant effect on classification accuracy, for best achieving results, the value of *N* were determined according to the value of *suppression threshold*.

Natural death threshold determined recognition region. If the affinity of an antibody to the antigen is less than σ_d, the antibody does not recognize antigen; so it's not stimulated and dies in natural death. Theoretically, the value of this parameter can be between the range of 0 and 1. Small values of σ_d slightly increases computational complexity and large values decreases diversity of population. Due to the experimental results, value of σ_d was better to be set lower than half of T_A. The final parameter discussed in this section is the number of seed cells. This parameter increased accuracy and convergence speed in some datasets, but it did not alter the results as much as other parameters. Since the compared algorithms did not employ this parameter in order to have a fair comparison with other classifier method, the value of the parameter was set to zero for all data sets.

C. *Comparative Analysis of Classification Accuracy*

The results of aiCLS have been compared with three artificial immune classifiers. The primary one was AIRS2 (10). The most popular and highly cited one is artificial immune supervised learning algorithm. However, since the results of AIRS2 and the first version of AIRS (7) are not completely equal, both versions of AIRS have been included in the comparative analysis. The third algorithm is SAIS, a simple and compact immune classifier, which outperformed AIRS in Diabetes and Iris but had lower efficiency than AIRS in Ionosphere. Table VII shows mean and standard deviation of classification accuracy of aiCLS and other immune classifiers in percentage.

Table VII demonstrates that outperforms results other immune classifiers on datasets of test bed. Note that, the maximum difference between aiCLS and AIRS is on Sonar dataset, which is a high dimension dataset; meaning that aiCLS is more robust to the curse of dimensionality. In addition, Table VIII compares aiCLS and a variety of popular classifiers and some of state-of-art classification methods such as C4.5+m+cf (23) ,AdaBoost SVM (24).

TABLE VII. COMPARISON OF AICLS AND OTHER IMMUNE CLASSIFIERS' CLASSIFICATION ACCURACY ON BENCHMARK DATA

Classifier	Wine	Iris	Sonar	Diabetes	Ionosphere
aiCLS	98.2% (1.2)	98.0% (2)	92.7% (6.2)	78.3% (5.5)	95.8% (1.2)
AIRS2	-	96.0% (1.9)	84.9% (9.1)	74.2% (4.4)	95.6% (1.7)
AIRS	-	96.7% (3.2)	84.0% (9.6)	74.1% (4.4)	94.9% (0.8)
SAIS	97.1% (4.7)	97.3% (4.7)	-	77.4% (5.7)	87.5% (4.4)

TABLE VIII. COMPARISON OF AICLS AND OTHER CLASSIFIERS' RESULTS ON BENCHMARK DATA

Rank	Wine		Iris		Sonar		Diabetes		Ionosphere	
1	IncNet	98.9	Grobian	100	**aiCLS**	**92.7**	**aiCLS**	**78.3**	3-NN + simplex	98.7
2	SSV opt-prune	98.3	**aiCLS**	**98.0**	TAP MFT Bayesian	92.3	Logdisc	77.7	3-NN	96.7
3	**aiCLS**	**98.2**	SSV	98.0	Naive MFT Bayesian	90.4	IncNet	77.6	IB3	96.7
4	kNN	97.8	C-MLP2LN	98.0	SVM	90.4	DIPOL92	77.6	MLP + BP	96.0
5	SSV opt-node	97.2	PVM 2-rules	98.0	Best two-layer MLP+BP 12hidden	90.4	DPA	77.5	**aiCLS**	**95.8**
6	SAIS	97.1	SAIS(10-fold)	97.3	AIRS2	84.9	SAIS	77.4	AIRS2	95.6
7	C4.5+m+cf	97.0	PVM 1-Rule	97.3	MLP+BP, 12 hidden	84.7	SMART	76.8	AIRS	94.9
8	FSM	96.1	AIRS	96.7	MLP+BP, 24 hidden	84.5	GTO DT	76.8	C4.5	94.9
9			AIRS2	96.0	1-NN Manhattan	84.2	kNN	76.7	RAIC	94.6
10			aiNET	96.0	AIRS	84	ASI	76.6	SVM	93.2
11			CART	96.0	MLP+BP,6 hidden	83.5	AdaBoost SVM	76.6	Nonlinear perceptron	92.0
12			FUNN	95.7	FSM-method?	82.6	Fisher DA	76.5	DB CART	91.3
13					1-NN Euclidean	82.2	MLP+BP	76.4	Linear Perceptron	90.7
…										
18			C4.5+m+cf	93.1					C4.5+m+cf	89.8
20									SAIS	87.5
…										
25							AIRS2	74.2		

D. Comparative Analysis of Data Reduction

Computational cost of the classification process is O(m*n), where m is count of generated memory cells and n is size of feature vector. Therefore, reducing the count of memory cells increases classification speed. Table IX compares data reduction capability of aiCLS, AIRS2 and AIRS. It have been shown that aiCLS outperforms AIRS in classification accuracy and data reduction. Therefore, the classifier trained by aiCLS is faster and more accurate than AIRS.

E. Computational Complexity

It has been mentioned that converting an iterative algorithm to one-shot learning algorithm does not guarantee reduction of computational complexity. Computational cost of aiNET for each iteration is $O(p^3)$, where p is length of input vectors (5). Therefore, computational cost of aiCLS is $O(p^3*k)$, where k is the average number of iterations of *iterative clonal selection* required for meeting stopping criteria. According to the experimental results for benchmark datasets, average execution of *iterative clonal selection* has been less than ten times. Therefore, k is too smaller than p^3 and it is ignorable in the computational cost of aiCLS. Therefore, the computational cost of aiCLS is assumed as $O(p^3)$. Table X shows the average run of Iterative clonal selection.

TABLE IX. COMPARISON OF DATA REDUCTION CAPABILITY OF AICLS AND OTHER IMMUNE CLASSIFIERS

Dataset	Wine	Iris	Sonar	Diabetes	Ionosphere
Size	161	120	192	691	200
aiCLS: Memory cells	**80.9/49%**	**28.9/76%**	**129.6/32.5%**	**259.7/62%**	126.6/37%
AIRS2: Memory cells		30.9/74%	177.7/7%	273.4/60%	**96.3/52%**
AIRS: Memory cells		42.1/65%	144.6/25%	470.4/32%	140.7/30%

TABLE X. AVERAGE RUNS OF ITERATIVE CLONAL SELECTION

Dataset	Wine	Iris	Sonar	Diabetes	Ionospher e
Average Iterations (*k*)	3.7	4.5	1.6	4.1	3.9

V. CONCLUSION

A new artificial immune classifier (aiCLS) has been proposed in this paper. Novelty of aiCLS lays in its iterative clonal selection and dissimilarity proportional clonal suppression (DPCS). For any given antigen, iterative clonal selection generates optimum antibodies, which was done through successive cloning and affinity maturation until a certain stopping criterion was met. Once the stopping criterion has been met, iterative clonal selection is completed and optimum antibodies are generated to recognize the antigen. Since this process repeats for any training antigens, aiCLS generates optimum antibodies for all training antigens. Therefore, classification process is supposed to be accurate if and only if suppression process maintains these optimum cells. Hence, DPCS stage is proposed to reduce redundant self-recognizing antibodies while maintaining optimum antibodies generated for the antigen. Therefore, DPCS stage is effective in data reduction, iterative clonal selection stage increases convergence speed of algorithm and both of them are effective in classification accuracy. The proposed method outperforms the accuracy of famous classifiers for the cited datasets found in UCI repository. In addition, the results of aiCLS on Sonar dataset determines that this method is significantly more efficient than AIRS on the datasets with high dimensional feature vector.

REFERENCES

[1] S. Forest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-Nonself discrimination in a computer," in proceedings IEEE symposium on Research in Security and Privacy, Oakland, CA, pp. 202-212, 1994.

[2] D. Dasgupta, and N. Attoh-Okine, "Immunity-based systems: A survey," in proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, pp. 369-374, 1997.

[3] L. N. de Castro, and J. Timmis, "Artificial immune systems as a novel soft computing paradigm," Soft Computing, vol. 3, no. 8, pp. 526-544, 2003.

[4] L. N. de Castro, and F. J. Von Zuben, The Clonal Selection Algorithm with Engineering Applications, in GECCO'00, 2000, 36-37.

[5] L. N. de Castro, and F. J. Von Zuben, "An evolutionary immune network for data clustering," in proceedings ofVI Brazilian Symposium on Neural Networks (SBRN'00), Rio de Janeiro, pp. 84, 2000.

[6] J. Timmis, and M. Neal, "A resource limited artificial immune system for data analysis," Knowledge-Based Systems, vol. 14, no. 3-4, pp. 121-130, 2001.

[7] A. Watkins, A Resource Limited Artificial Immune Classifier, Master's Thesis, Mississippi State University, MS. USA., December 2001.

[8] L. Meng, P. v. d. Putten, and H. Wang, "A comprehensive benchmark of the artificial immune recognition system (AIRS)," in Advanced Data Mining and Applications. Springer Berlin/Heidelberg, vol. 3584, 2005, pp. 575-582.

[9] A. B. Watkins, and L. Boggess, "A resource limited artificial immune classifier," Evolutionary Computation, vol. 1, Honolulu, HI, pp. 926-931, 2002.

[10] 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, no. 3, pp. 291-317, 2004.

[11] K. Leung, F. Cheong, and C. Cheong, "Generating compact classifier systems using a simple artificial immune system," IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, vol. 37, no. 5, pp. 1344-1356, 2007.

[12] Y. Zhong, L. Zhang, B. Huang, and P. Li, "An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery," IEEE Transactions on Geoscience and Remote Sensing, vol. 44, no. 2, pp. 420-431, 2006.

[13] T. Oda, and T. White, "Immunity from spam: An analysis of an artificial immune system for junk email detection," in Artificial Immune Systems. Heidelberg: Springer Berlin, 2005, pp. 276-289.

[14] Draman, Noor Azilah, Campbell Wilson, and Sea Ling. "Modified Ais-based Classifier for Music Genre Classification." In Proc. ISMIR. 2010.

[15] D. Tsankova, and V. Rangelova, "Modeling cancer outcome prediction by aiNet: Discrete artificial immune network," in Control & Automation, Mediterranean Conference (MED '07), Athens, 2007, pp. 1-6.

[16] Martinelli, Eugenio, Gabriele Magna, Saverio De Vito, Raffaele Di Fuccio, Girolamo Di Francia, Alexander Vergara, and Corrado Di Natale. "An adaptive classification model based on the artificial Immune system for chemical sensor drift mitigation." Sensors and Actuators B: Chemical (2012).

[17] A. Ehsani, and A. M. Eftekhari, "Content semantic model generation using aiNET to enhance accuracy of image retrieval systems," in CSI Computar Conference, Tehran, 2009 (in Persian).

[18] A. Ehsani, and A. M. Eftekhari, "Salt and paper noise detection based on negative selection and aiNET," in CSI Computer Conference, Tehran, 2009 (in Persian).

[19] J. Timmis, M. Neal, and J. Hunt, "An artificial immune system for data analysis," Biosystems, vol. 55, no. 1-3, pp. 143-150, 2000.

[20] S. Sahan, H. Kodaz, S. Gunes, and K. Polat, "A new classifier based on attribute weighted artificial immune system (AWAIS)," in Computer and Information Sciences – ISCIS, vol. 3280, Springer Berlin / Heidelberg, 2004, pp. 11-20.

[21] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection," in Evolutionary Computation, Vancouver, 2006, pp. 664-671.

[22] C. L. Blake, and C. J. Merz, UCI Repository of Machine Learning, 1998, Accessable at: http://www.ics.uci.edu/mlearn/MLRepository.html

[23] J. Demsar, "Statistical comparisons of classifiers over multiple datasets," Journal of Machine Learning Research, vol. 7, pp. 1-30, 2006.

[24] X. Li, L. Wang, and E. Sung, "AdaBoost with SVM-based component classifiers," Engineering Applications of Artificial Intelligence, vol. 21, pp. 785-795, 2008.

S. Amir Ehsani was born in Tehran, Iran in 1981. He received his B.Sc. degree in Software Engineering in 2006 and his M.Sc. degree in A artificial Intelligence in 2009 from Qazvin Islamic Azad University, Qazvin, Iran. His research interests include soft computing, data analysis, knowledge management and image retrieval. He is currently working as CKO in Sanaye Varaq Iran Company.

Amir Masoud Eftekhari Moghadam was born in Tehran, Iran in 1964. He received his B.Sc. and M.Sc. degrees in computer engineering from Iran University of Science and Technology in 1992 and 1995, respectively and his Ph.D. degree in computer engineering in 2002 from Science and Research Branch, Islamic Azad University, Tehran, Iran. In 1996 he was a lecturer of computer engineering, and since 2002 he has been assistant professor at the Qazvin branch, Islamic Azad University, Qazvin, Iran. His research interests include image processing, image retrieval, image/video coding, semantic video analysis, data mining and image security.