Turkish Journal of Electrical Engineering and Computer Sciences

Volume 21 | Number 6 Article 14

1-1-2013

Performance evolution of a newly developed general-use hybrid AIS-ANN system: AaA-response

SERAL ÖZŞEN

SALİH GÜNEŞ

TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1703 – 1719

doi:10.3906/elk-1201-22

Performance evolution of a newly developed general-use hybrid AIS-ANN system: AaA-response

Seral ÖZŞEN,* Salih GÜNEŞ
Electrical and Electronics Engineering Department, Faculty of Engineering, Selçuk University, Konya, Turkey

Received: 10.01.2012 • **Accepted:** 23.05.2012 • **Published Online:** 02.10.2013 • **Printed:** 28.10.2013

Abstract: In this study, we have developed a nonlinear recognition system in the artificial immune systems (AIS) field named 'AaA-response (artificial neural network (ANN)-aided AIS-response)', which is different from previous AIS methods in that it uses a different modeling strategy in the formation of the memory response. Because it also uses ANNs in the determination of the correct output, it can be seen as a hybrid system that involves AIS and ANN approaches. Unlike the other AIS methods, AaA-response uses multiple system units (or antibodies) to form an output for a presented input. This property gives the proposed system the ability of producing the desired output values, other than just being a classification algorithm. That is, AaA-response can also be used as a regression method, like ANNs, by producing any output value for the given inputs.

The parameter analyses of the system were conducted on an artificially generated dataset, the Chainlink dataset, and the important points in the parameter selection were emphasized. To investigate the performance of the system for real-world problems, the Iris dataset and Statlog Heart Disease dataset, taken from the University of California - Irvine machine learning repository, were used. The system, which obtained 99.33% classification accuracy on the Iris dataset, has shown an important performance superiority with regard to the classification accuracy to other methods in the literature by reaching 90.37% classification accuracy for the Statlog Heart Disease dataset.

Key words: Artificial immune systems, artificial neural networks, recognition, memory response, learning

1. Introduction

The natural immune system (IS) is an adaptive, complex, distributive, and diverse learning system. These are some properties of the IS, that make it an attractive inspiration source for artificial intelligence designers. In the last 15 years, many studies have been conducted that used the IS in this regard to solve complex problems.

The intelligence and effectiveness of natural systems have attracted the attention of researchers while forming complex systems. Models ranging from neural networks and genetic systems to bee colonies have obtained considerably good results.

The IS consists of interacting elements whose responsibilities are to protect the body. The system involves layers distributed through the body that produce a series of reactions and responses. The IS, with properties like recognition, memory, distributed mechanism and error tolerance, has formed a base for artificial ISs (AISs) to be used in artificial intelligence applications. By modeling many of the IS properties, various AIS algorithms have been developed for the solution of complex problems in different fields. Machine learning, classification, optimization, virus detection, anomaly detection, and scheduling are some of these fields.

*Correspondence: seral@selcuk.edu.tr

In the machine learning applications of the AIS, several studies were conducted. De Castro and Von Zuben modeled the clonal selection mechanism in their study to control the dynamics of their AIS system [1]. On the other hand, Timmis and Neal [2] developed an unsupervised AIS network inspired by the studies of Farmer et al. [3] and Hunt et al. [4]. One of the studies in this field is the study of Watkins [5]. He formed an AIS system called the artificial immune recognition system (AIRS) using metaphors such as competition, clonal selection, and memory acquisition. In addition, Garrett used a different binary representation scheme in his clonal selection model [6]. In AIS algorithms, the interactions are generally computed by some distance measures, but in some applications, this can bring some disadvantages. In their study, Sahan et al. developed an AIS model named attribute-weighted AIS (AWAIS) that uses weighted distance measures to overcome these kinds of shortages [7]. In another work, 2 clonal selection models, CLONALG and opt_IA, were compared [8]. Özşen et al. used their system (AWAIS) to diagnose atherosclerosis disease and reached 100% classification accuracy [9]. In another study, Aydin et al. used an immune classifier with a swarm learning concept [10]. Graff and Engelbrecht, on the other hand, used an IS-based clustering algorithm for clustering data in uncertain environments [11]. Luh and Lin applied an AIS to a human face recognition application [12]. An interesting study related with AISs was done by combining a quantum immune algorithm with the backpropagation neural network by Ji et al. [13]. These are only a few of the conducted studies.

There are many systems that were developed with this concept, but although these studies achieved remarkable results, it cannot be said that they are more successful than the current well-known methods whose successes are certified. Hence, to move AIS to a place above them, a different methodology from that of the previous AIS studies should be carried out. Starting from this point of view, we developed a general-use hybrid AIS-artificial neural network (ANN) system. It differs from the other AIS studies in that it uses a different representation strategy and it can produce real outputs according to the presented input, more than just recognizing input data. Moreover, our system accepts the aid of ANNs to form the correct response.

After giving the background information about ISs and AISs in the following sections, the developed system is explained. The parameter analyses and performance evolution of the system are conducted using the Chainlink, Iris, and Statlog Heart Disease datasets, taken from the University of California - Irvine (UCI) machine learning repository [14], and the application results are given in the subsequent sections. A discussion of the obtained results is provided in a later section.

2. Background

ISs protect the body against harmful microorganisms by defending it with necessary processes. The conducted processes in this defense mechanism are named the 'immune response' as a whole. If this response succeeds, the cells that have a high affinity to the microorganism (that recognize the microorganism at a high level) are saved to be used in the next attacks of similar microorganisms for a more rapid and powerful response. This is the 'immunization' of the body to that microorganism.

Two principal immune mechanisms play a role in immunization: innate immunity and adaptive immunity. The former response is triggered by the common structures of the microbes and it provides a general response. Adaptive immunity, on the other hand, is a second phase of defense and it is specific for different microbes. It is this response that is modeled in most AIS methods for the sake of the memory formation process in it. The processes of adaptive immune response are conducted as follows: When an antigen (Ag, a foreign microbial particle) enters the body, antigen presenting cells (APSs) circulating in the lymph liquid capture the Ag and

divide it into peptides to present to the helper T cells. Moreover, at the same time, if the B cells recognize the receptors of the Ag and if there exist stimulating signals secreted from the helper T cells, the B cells are activated. The activated B cells grow by cloning. Some of these clone populations turn into plasma cells. Plasma cells secrete their antibodies (Ab receptors of the plasma cells and B cells), which are then attached to the receptors of the Ag and form an Ab-Ag complex. This attachment causes the secretion of some chemicals that stimulate some other cells responsible for ingesting the Ab-Ag complex. On the other hand, some other B cells enter the mutation cycle. The affinities of the B cells to Ag are changed in the mutation cycle. Some of the mutated B cells (with the highest affinities) and some of the activated B cells are selected to form a *B-cell* pool. If the receptors of the Ag are not eliminated completely, the B cells in this pool continue to fight with the Ag by entering the cloning-plasma cell–mutation cycle. Each time, the number of B cells that will be collected for the pool is arranged according to the number of remaining receptors. For example, if the number of receptors is bigger than the Ab number, then more B cells are taken into the pool. However, if the Ab number is bigger, fewer B cells enter the pool. This cycle continues until there is no remaining Ag in the body. The last *B-cell* pool involves the B cells whose affinities to the Ag are the highest. These B cells are kept as memory cells [15].

To model this kind of mechanism in the IS to form an AIS, one needs the following [16]:

- A representation scheme for the system units and input data.
- Some computational tools for calculating the interactions of the system units with themselves and the environment.
- Some adaptation procedures for modeling the processes of the IS.

Each AIS is developed based on the application field. A representation method should be selected in accordance with the application field and computational tools are used for that representation method. In the last step of system formation, the researcher should use a mechanism in the IS. For example, one can model the clonal selection procedure in the IS, while another researcher can select the adaptive immune response given briefly in the above paragraph to form the adaptation procedures. The IS involves various mechanisms to model, such as clonal selection, adaptive immune response, innate immune response and negative selection. The key point is correctly modeling the appropriate mechanism for the current problem. So far, various methods have been developed for the steps mentioned above. For example, the shape–space representation scheme is used widely for the first and second step [17]. Many of the AIS systems in the literature have a common point: they represent input data as Ags and system units as Abs or B cells. The system learns by forming memory Abs or B cells by adapting its system units according to their affinities to the presented Ag. Generally, however, they use a one-to-one approach in that formation; that is, one B cell or Ab is produced as the memory unit that best recognizes the presented Ag. However, in our study, we assumed that an Ag contains different receptors in its surface and the elimination of that Ag is realized by the recognition of these receptors completely. Each receptor of the Ag can be recognized by different Abs or B cells. Thus, an Ag response does not involve a single Ab or B cell. Moreover, it is a collective response of an Ab group or a B cell group. The determination of which output will be produced for a memory cell group is conducted with the aid of ANN, as explained in the following sections.

3. Materials and methods

3.1. Developed system

The learning phase of the ANN-aided AIS-response (AaA-response) is composed of 2 stages (Figure 1). In the first stage, the memory B cells that recognize each receptor are formed by modeling the procedures in the adaptive immune response, except for modeling the helper T cells. These memory cells are held in a vector. The second stage is used to determine which memory cell configuration would give which output. For this purpose, an ANN architecture is used.

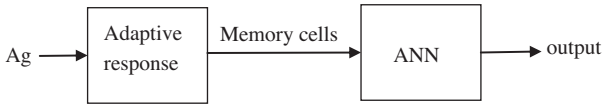

Figure 1. The main structure of the developed system.

The principal working procedure of the system can be explained with an example: suppose that there are 3 input data (that is, 3 *Ag*s in our model) that will be presented to the system with their corresponding outputs, which are expected to be given by the system:

(Ag_1 = [0.3 0.2 0.1], Ag_2 = [0.4 0.5 0.9], Ag_3 = [0.1 0.4 0.3]), corresponding outputs = [2, 3, 5].

In this case, the memory cells that best recognize the features of the *Ag*s are produced. Let these memory cells that are produced be the following:

B_1 = [0.31], B_2 = [0.23], B_3 = [0.09], B_4 = [0.39], B_5 = [0.52], and B_6 = [0.87].

These 6 memory B cells are then used as the inputs of the second stage in the following manner: for a presented *Ag*, the memory cells that recognize its features are determined by the minimum distance criterion. For example, the B cells that recognize the above *Ag*s are determined as follows:

Ag_1: $\{B_1B_2B_3\}$; Ag_2: $\{B_4B_5B_6\}$; Ag_3: $\{B_3B_4B_1\}$.

The outputs for these configurations are then:

$\{B_1B_2B_3\}$ → *output (target value for ANN): 2*,

$\{B_4B_5B_6\}$ → *output (target value for ANN): 3*,

$\{B_3B_4B_1\}$ → *output (target value for ANN): 5*.

The ANN is used to learn the relations between the memory cells by adapting its weights to produce the related outputs when the above B cells are given as inputs. For our example, the inputs and corresponding outputs that will be presented to the ANN will be:

[0.31(B_1) 0.23(B_2) 0.09(B_3)] → 2 (target value),

[0.39(B_4) 0.52(B_5) 0.87(B_6)] → 3 (target value),

[0.09(B_3) 0.39(B_4) 0.31(B_1)] → 5 (target value).

After the system's training procedure is completed in this way, the system learns to produce the output that should be produced for the presented data. When an *Ag* is presented to the system, the memory cells that best recognize its features are determined and the values of these memory cells are given to the ANN as inputs. The output taken from the ANN structure is the response of the system produced for that *Ag*.

3.1.1. Modeling strategy

In the system, input data are represented as *Ag*s, as stated above, while B cells and *Ab*s are used as system units in the training phase. The *Ag*s are normalized vectors in [0–1] with L features, whereas the B cells

and *Ab*s are modeled as real numbers that can take values in the [0–1] interval. In the IS, the *B* cells are differentiated with its receptors, *Ab*s, and so there are no differences between a *B* cell and its *Ab*. Hence, a *B* cell and its *Ab* is represented by the same value in the model. The system has 3 inputs: the input data, the number of receptors, and the target values of each input data that the system tries to find with minimum error. The training phase of the first stage is conducted in sequential order and it continues until the number of receptors is 0 for the current *Ag*. In the natural IS, when an *Ag* enters the body, it copies itself. Thus, the IS tries to eliminate the colony instead of trying to fight with an *Ag* only. The aim in assigning a value for the receptors and using it as a second input is to model that feature in the IS in our system. This number is determined for each *Ag* and held in a vector named *Num_peptides*.

3.1.2. Processes of the first stage

The flow chart of the first stage is shown in Figure 2, where it is seen that a *B* cell population is formed randomly for each feature of the presented *Ag* at the beginning. Moreover, the total receptor number is calculated by multiplying the feature number by the receptor number for that *Ag*, which is taken from the *Num_peptides* vector for the corresponding *Ag*. Next, the immune response cycle is conducted for each receptor until the number

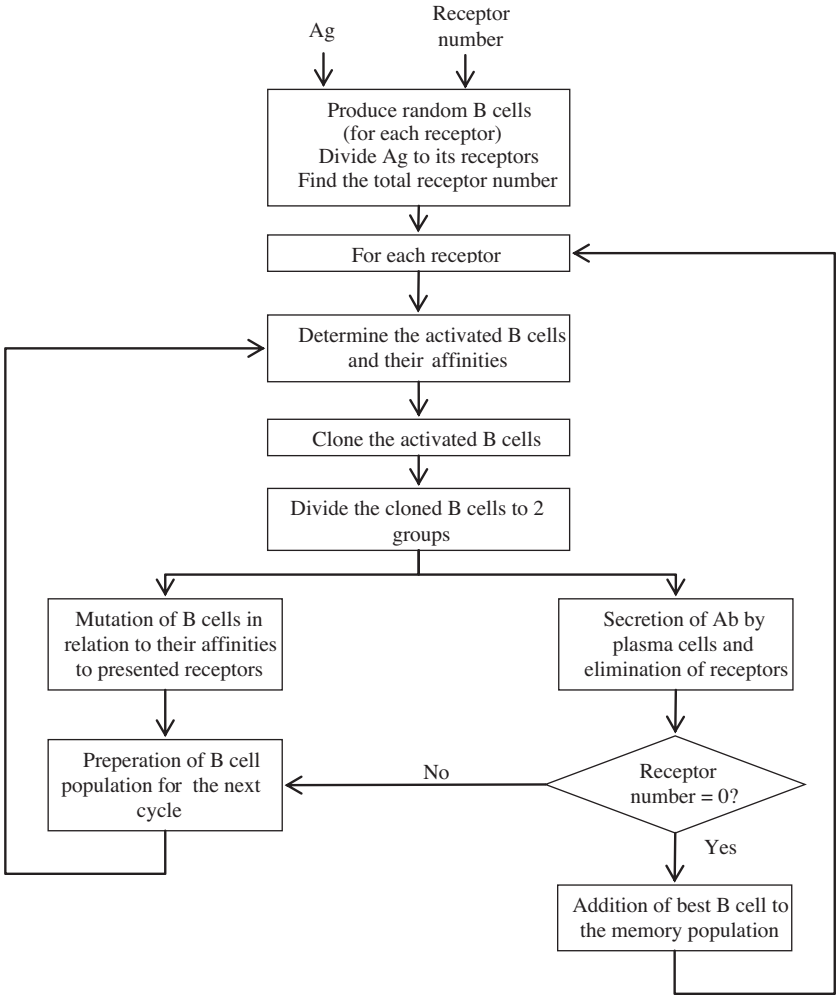

Figure 2. The block diagram of the AIS stage of the AaA response system.

of receptors is 0. The first process of this cycle is the determination of the activated B cells from the B cell population, dedicated to the related receptor and their affinities to the receptor. In the next step, these activated B cells are cloned in proportion to their affinities. The cloned population divides into 2 subpopulations: some of the cloned B cells secrete Abs as plasma cells. One plasma cell can secrete only 1 Ab.

The number of related receptors is decreased with the number of those secreted Abs. On the other hand, the remaining B cells in the cloned population enter the mutation process. At this point, the remaining receptor number is controlled. If the receptor number is 0, it means that there is no receptor in the system to destroy, and the immune response cycle is stopped for that receptor. Otherwise, it continues and a B cell population for the next cycle is prepared. This population consists of some mutated B cells and some activated B cells. To control the population dynamics of the system, the number of B cells for the next cycle is limited to the receptor number and the Ab number as its counterpart in the natural IS: if the number of receptors is bigger than the number of Abs, more B cells are needed for the next cycle. On the other hand, if there exist more Abs than receptors, it means that the receptors have begun to be destroyed and a lower number of B cells should be taken for the next cycle. When the cycle is stopped for the related receptor, the best B cell with the highest affinity to that receptor is taken as a candidate memory cell. This candidate cell is not directly taken to the memory cell pool. Again, to avoid very similar cells from being in the memory pool, the similarities between this cell and the memory cells are calculated. If any of these similarity values are higher than a threshold value ($Thres\_Mem$), the candidate memory cell is not added to the memory cell population. Thus, there must be no similar B cell in the memory population for the candidate cell to be added to that population.

The affinities are calculated in the algorithm as in the following:

$$aff_B = 1 - |(B - cellvalue) - (receptor\ value)|. \tag{1}$$

Here, aff_B is the affinity value between a B cell and a receptor.

Moreover, in the cloning process, a cloning parameter ($clon\_param$) is multiplied with the affinity of the cell that will be cloned. Next, the B cell is cloned for a number of times that is equal to the result of this multiplication.

In the mutation process, on the other hand, a mutation rate (mr) is determined by using $mr = 1 - aff_B$, in which aff_B is the affinity value of the B cell that will be mutated. The mutation process is then conducted according to the formula:

$$Bcell' = Bcell \pm mr \times Bcell. \tag{2}$$

Here, $Bcell'$ is the mutated B cell, $Bcell$ is the B cell before the mutation, and mr is the mutation rate.

For the preparation of the B cell population for the next cycle, 2 points should be considered: what will the number of B cells for this population be, and which cells will be taken? The number of B cells is arranged in accordance with the natural IS, as stated in the above paragraphs, with the following equation:

$$B\_num\_nextB\_num + difference/5. \tag{3}$$

Here, B_num_next is the number of B cells that will enter the population for the next cycle, B_num is the current number of B cells, and the difference is:

$$difference = numberofreceptors - numberofAbs. \tag{4}$$

The number of receptors in the above equation is the number of remaining receptors after the elimination by the Abs. When it comes to the selection of the B cells for the next cycle, again, the natural IS is used as an

inspiration source and a *B-cell* pool is prepared. This pool involves mutated B cells, activated B cells, and randomly generated B cells. They are ordered in the pool according to their affinities in decreasing order. The B cells with the number of *B_num_next* are selected among these cells for the next cycle.

After the formation of the memory B cells for the receptors of the *Ag*s in the input data, it is determined which memory cells best recognize the receptors of each *Ag*. To do this, the distances between the memory cells and receptors of the *Ag*s are calculated by the following formula:

$$Dist_{receptor,B} = |receptor\text{-}B|. \tag{5}$$

Here, $Dist_{receptor,B}$ is the distance value between the related receptor (*receptor*) and the memory B cell (B). The value of the B cell, whose distance to the related receptor is minimum, is held in the *rec_cells* matrix, which involves the information about which memory cells recognize which receptor best. Hence, the size of this matrix is equal to the size of the input data matrix.

3.1.3. Processes of the second stage

The training of the AaA-response system is not over with the first stage. After obtaining memory B cells and the *rec_cells* matrix, which includes the information about which cells recognize the receptors of the input *Ag*s from the first stage, an ANN structure is trained to learn the relation between the recognized cell configurations and the output values. Here, the *rec_cells* matrix is the input of the ANN and the output values given in the beginning of the problem for each *Ag* are the target values for the ANN training. Because an ANN architecture is used in this stage of the algorithm, the performance of the ANN will affect the performance of the system a great deal. For example, the selection of the hidden node number and hidden layer number has an effect, as does the selected learning procedure.

With the training process, the system produces memory B cells and it learns the weights of the ANN, which determines the relation between these memory cells. After training, the system is ready to produce an output value for any presented data by first determining the best recognizing memory cells and then by calculating the relation between these cells with its learned ANN structure. This hybrid structure makes the AaA-response system a multifunctional AIS, which can be used as a classifier as well as a regression method. So far, no AIS method has been developed that can produce any output values for presented inputs. The AaA-response system tries to fill this gap and gives the opportunity of using the AIS method in such problems where there is a need for a real output value for an input rather than just classifying it.

3.2. Used datasets

The performance analyses of the system were conducted with the Chainlink dataset. This dataset contains 1000 data with 2 classes, each containing 500 data. There are 3 features in this nonlinear dataset as the x, y, and z coordinates, as shown in Figure 3.

To see the performance of the system for real problems and to compare it with other methods, the Iris dataset and the Statlog Heart Disease dataset taken from the UCI machine learning repository [14] are used in the applications.

The Iris dataset is a well-known dataset in the literature and is widely used for performance evolution purposes in newly developed systems. It consists of 150 data taken from the Iris flower. There are 3 classes, named Iris Setosa, Iris Versicolor, and Iris Virginica, each having 50 data with the 4 features of sepal length, sepal width, petal length, and petal width.

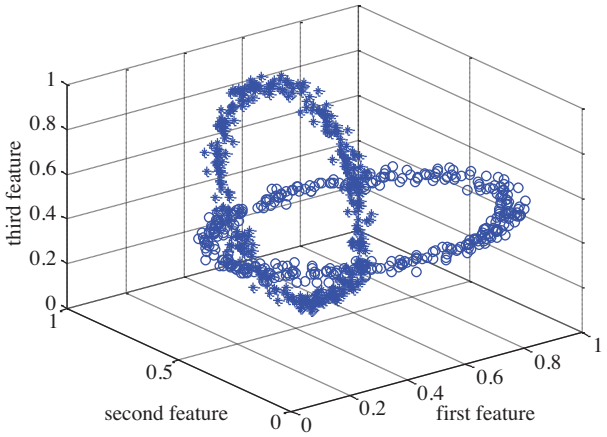

Figure 3. Chainlink dataset.

The Statlog Heart Disease dataset is the other dataset used in our applications. This dataset was formed from data taken from 270 people, of which 150 had heart disease while the remaining 120 people were healthy. The data consist of 13 features, which are: 1) age, 2) sex, 3) chest pain type (4 values), 4) resting blood pressure, 5) serum cholesterol in mg/dL, 6) fasting blood sugar of >120 mg/dL, 7) resting electrocardiographic results (values of 0, 1, and 2), 8) maximum heart rate achieved, 9) exercise-induced angina, 10) oldpeak = ST depression induced by exercise relative to rest, 11) the slope of the peak exercise ST segment, 12) the number of major vessels (0–3) colored by fluoroscopy, and 13) thal: 3 = normal; 6 = fixed defect; and 7 = reversible defect. There are 2 classes within these data and the class information is contained in the 14th feature as 1 and 2 regarding the absence and presence of disease, respectively.

3.3. Performance evaluation criterion

In classification applications, the data should be divided between the training and test sets. Several approaches can be followed while doing this, but the k-fold cross-validation (k-fold CV) is generally used among researchers because it minimizes the bias associated with the random sampling of the training [18]. All of the data were randomly divided into k mutually exclusive and approximately equally sized subsets. The classification algorithm was trained and tested k times. In each case, one of the folds is taken as the test data, and the remaining folds are added to form the training data. The average of these k results gives the test accuracy of the algorithm. In our applications, we use a 3-fold CV for the Iris dataset and a 5-fold CV for the Statlog Heart Disease dataset. The classification accuracy of the system is calculated using the following formula.

Classification accuracy:

$$\begin{aligned} accuracy(T) &= \frac{\sum_{i=1}^{|T|} assess(t_i)}{|T|}, \; t_i \in T \\ assess(t) &= \begin{cases} 1, & if\, classify(t) = t.c \\ 0, & otherwise \end{cases} \end{aligned} \quad . \tag{6}$$

Here, T is the set of data items to be classified (the test set), tT; $t.c$ is the class of item t; and *classify(t)* returns the classification of t.

4. Results

To analyze the parameter effects on the system's performance, the Chainlink dataset is used. As explained in Section 3.1, the AaA-response is a 2-stage learning system. In the first stage, the memory cells are produced

that best recognize the features of the presented data and the second stage determines the output via an ANN, according to the relation between these memory cells. In the parameter analyses, the effects of parameters in these 2 stages are inspected, which are:

Thres_B: the threshold value for B cells to be activated.

Thres_Mem: the threshold value for candidate memory cells to enter the memory population.

In the second stage of the system, because an ANN is used, several parameters affect the performance, such as the hidden layer number, hidden layer nodes, training procedure, and parameters of this procedure. In the parameter analyses, some of these factors are fixed to specific values. These are as follows: the ANN is used with 1 hidden layer and 10 hidden nodes, the logarithmic sigmoid activation function is used in its input nodes and hidden nodes, the linear activation function is used in its output nodes, and the Levenberg–Marquard (LM) backpropagation algorithm is used as a learning strategy. The MATLAB Neural Network toolbox was utilized in the second stage and the default values of the LM algorithm were used as learning parameters, as follows:
net.trainParam.epochs = *5000*;

net.trainParam.goal = *0*;

net.trainParam.min_grad = *1e-20*;

net.trainParam.mu = *0.001*;

net.trainParam.mu_dec = *0.1*;

net.trainParam.mu_inc = *7*;

net.trainParam.mu_max = *1e50*;

net.trainParam.show = *500*.

There are many alternatives to use for the learning strategy and architecture, but because changing between these alternatives is known by many researchers, the emphasis was given to the AIS part of the system.

4.1. Parameter analyses of the Chainlink dataset

To conduct the parameter analyses, the Chainlink dataset was divided into 2 subsets of 700 and 300 data for the training and test sets, respectively. The training data consist of 350 data from 1 class and 350 from the other. Likewise, the test set involves 150 data of 1 class and 150 of the other. The input data are normalized between 0 and 1. During the analyses, an ANN structure with 3 input nodes (Chainlink has 3 features), 1 hidden layer, and 1 output node (to give the class information as 1 or 2) was used with the above mentioned learning strategy (LM with default parameters).

The first parameter whose effect was investigated was the *Thres_B* parameter. To see its impact, the other parameters are fixed at some optimal values as *Num_peptides* = 250 and *hidden node number* = 10. These values are selected using the trial-and-error approach from the experiments done several times. The change of the *Thres_B* parameter is analyzed for 3 values of the *Thres_Mem* parameter, because the 2 parameters work together to form the optimal memory cell. These values are *Thres_Mem* = 0.6, *Thres_Mem* = 0.7, and *Thres_Mem* = 0.75. The reason for selecting these values is that these are the values with which the system performs well. While conducting the investigation on the *Thres_B* parameter, a *Thres_Mem* value is selected and the *Thres_B* value is increased, starting from 0.1 to the value of the *Thres_Mem*. This is because the *Thres_B* value cannot be larger than the *Thres_Mem* value. For example, for the *Thres_Mem* value fixed to 0.6, the *Thres_B* parameter is increased from 0.1 to 0.6 and the classification accuracies of the system are noted for each of these values. The change of the classification accuracy with the changing *Thres_B* parameter for *Thres_Mem* =0.6, *Thres_Mem* = 0.7, and *Thres_Mem* = 0.75 is shown in Figures 4a, 4b, and 4c, respectively.

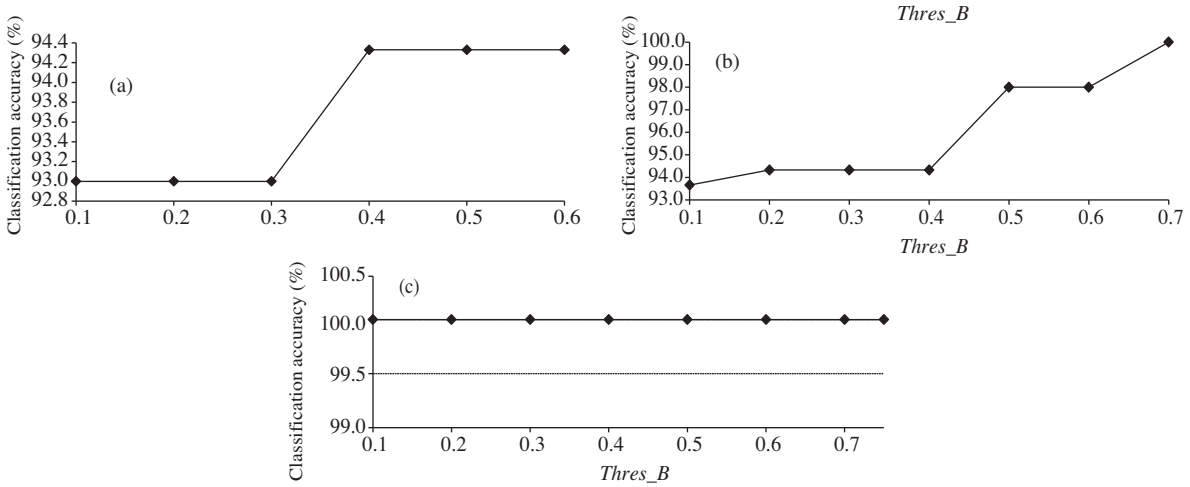

Figure 4. The change of the classification accuracy with the changing value of the *Thres_B* parameter for 3 values of the *Thres_Mem* parameter: a) *Thres_Mem* = 0.6, b) *Thres_Mem* = *0.7*, and c) *Thres_Mem* = 0.75.

As shown from Figure 4, as the value of the *Thres_B* increases, the classification accuracy increases for *Thres_Mem* = 0.6 and 0.7. No comment can be made on the *Thres_B* change for *Thres_Mem* = 0.75 because the classification accuracy is 100% for all of the *Thres_B* values. One point should be noticed here: the classification accuracy increases with the increasing value of the *Thres_B*, but the *Thres_Mem* value also affects this increase in a positive manner.

To investigate the effect of *Thres_Mem*, the other parameters are again fixed at some optimal values as *Num_peptides* = 250, *hidden node number* = 10. The value of *Thres_Mem* is increased for the 2 constant values of the *Thres_B* parameter: *Thres_B* = 0.2 and *Thres_B* = 0.5. Because *Thres_Mem* should be larger than *Thres_B*, it is increased, starting from the value of the used *Thres_B*. The change of the classification accuracy with the changing *Thres_Mem* for *Thres_B* = 0.2 and *Thres_B* = 0.5 is shown in Figures 5a and 5b, respectively.

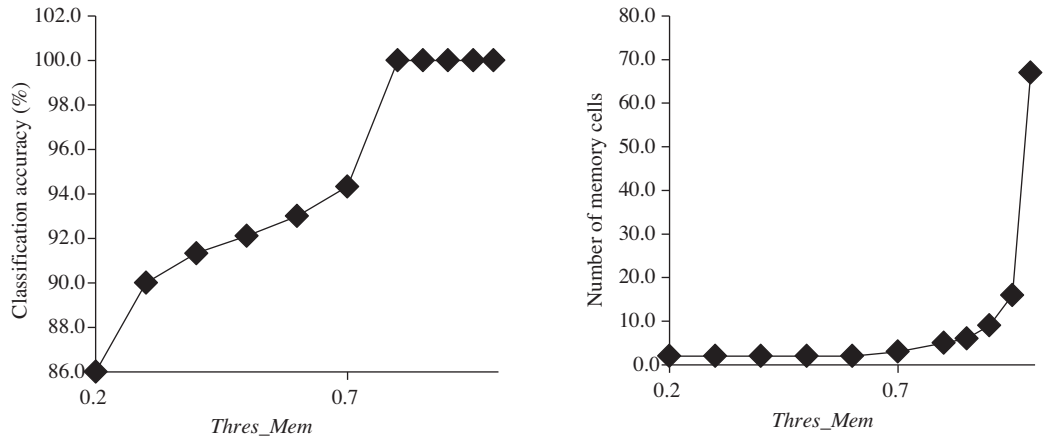

Figure 5. The change of classification accuracy with changing values of the *Thres_Mem* parameter for 2 values of the *Thres_B* parameter: a) *Thres_B* = 0.2 and b) *Thres_B* = *0.5*.

The increase of the *Thres_Mem* parameter also results in an increase of the classification accuracy. However, for a higher *Thres_B* value, the best classification accuracy (100%) is obtained for a lower value

of the *Thres_Mem* parameter. The effects of these 2 parameters on classification accuracy can be explained briefly as follows: the *Thres_B* value adjusts the goodness of the candidate memory cells. Because this is a threshold value that B cells must surpass in order to be activated, if its value is selected to be high, the B cells whose affinities are higher than this value will be activated. Therefore, the overall affinity (or goodness) of the produced candidate memory B cell will be much better. This is the reason for the increasing classification accuracy with the increasing *Thres_B* value. However, care must be taken not to give too high of values to this parameter because, in that case, the system will be too specific. To avoid this, we gradually increase this value to find the optimum *Thres_B* value. On the other hand, the *Thres_Mem* value determines whether the candidate cell will be added to the memory population or not. If the similarity between any memory cell and candidate memory cell is bigger than this value, the candidate cell is not added to the memory population. Hence, if this parameter value is selected to be high, finding similar cells in the population gets harder and the entrance of candidate cells into the memory population gets easier. This results in a more crowded memory population, which presents alternatives to recognize the presented receptors. The increase of the classification accuracy with the increasing *Thres_Mem* value can be explained like this, but again, one must be careful not to give to high of a value to this parameter, because in that case, an over-crowded population arises, which causes over-fitting problems.

4.2. Applications on the Iris dataset

The first dataset to see the performance of the AaA-response system with real-world problems is the Iris dataset. As stated earlier, a 3-fold CV method was applied to determine the training and test sets. In the conducted experiments, the *Num_peptides* and *Thres_B* parameters were fixed at the values of 250 and 0.6, respectively, based on previous experience. The experiments were conducted in the following manner: the *Thres_Mem* parameter is changed to produce a different number of memory cells. For each of these parameter values, the structure of the ANN is changed using a different number of hidden layers and a different number of hidden nodes. Hence, the aim is to reach the optimum ANN architecture giving the highest classification accuracy for that *Thres_Mem* value. In all cases for the Iris dataset, the input node number is 4 and the output node number is 1. The used *Thres_Mem* values for the different memory cell numbers are shown in Table 1.

Table 1. The produced memory cell numbers for the used *Thres_Mem* parameters in the applications on the Iris dataset of the AaA-response system.

Thres_Mem	Number of produced memory cells
0.800	4
0.900	8
0.950	15
0.970	22
0.985	37
0.990	57
0.995	71
0.999	125

The experimental ANN architectures for each of these *Thres_Mem* values are:

- One hidden layer with hidden node numbers 5, 10, 15, 20, 25, and 30.
- Two hidden layers with hidden node numbers (15-3), (15-5), (15-10), (15-15), (30-3), (30-5), (30-10), and (30-15), where the first numbers are the hidden node numbers of the first hidden layer and the second numbers are for the second hidden layer's numbers.

- Three hidden layers with 10 hidden nodes for the first layer, 20 hidden nodes for the second layer, and 3 hidden nodes for the last hidden layer.

The best classification accuracies obtained from the best ANN architecture for each of the *Thres_Mem* parameters are given in Table 2.

Table 2. The highest classification accuracies obtained with the optimum ANN architecture for each *Thres_Mem* value.

Thres_Mem	Best ANN configuration	Highest classification accuracy obtained by using the best ANN configuration (%)
0.8	1st hidden layer: 30 nodes 2nd hidden layer: 10 nodes	87.33
0.9	1st hidden layer: 5 nodes 2nd hidden layer: 5 nodes	95.33
0.95	1st hidden layer: 5 nodes 2nd hidden layer: 15 nodes	98.67
0.97	1st hidden layer: 10 nodes 2nd hidden layer: 20 nodes 3rd hidden layer: 3 nodes	98.67
0.985	1st hidden layer: 10 nodes 2nd hidden layer: 20 nodes 3rd hidden layer: 3 nodes	99.33
0.99	1st hidden layer: 10 nodes 2nd hidden layer: 3 or 5 nodes	99.33
0.995	1st hidden layer: 5 nodes 2nd hidden layer: 3 nodes	98.67
0.999	1st hidden layer: 5 nodes 2nd hidden layer: 15 nodes	98.00

The highest classification accuracies for the experimental *Thres_Mem* values and the obtained memory cell numbers for these values are shown in Figures 6a and 6b, respectively. As stated above, in the explanation of the effect of the *Thres_Mem* parameter, after a point in the *Thres_Mem* increase, the classification accuracy decreases because of the over-fitting problem.

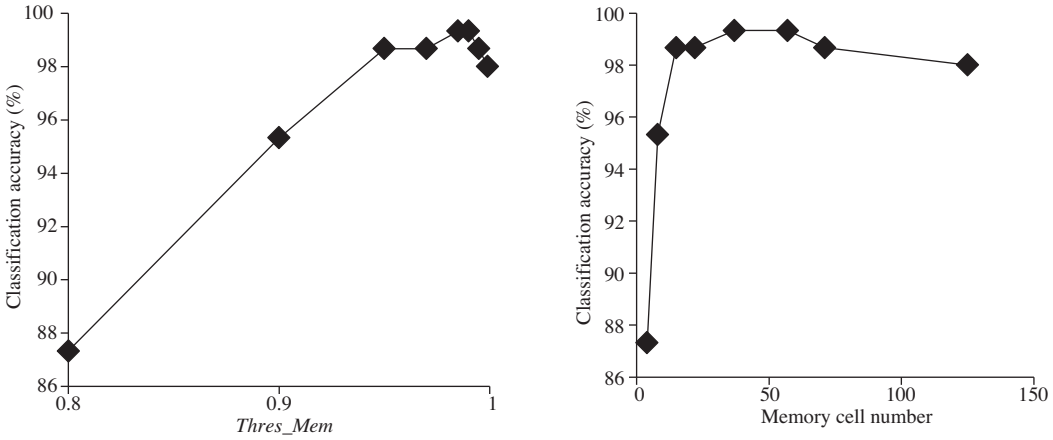

Figure 6. The change of classification accuracy in the applications of the AaA-response system for the IRIS dataset with regard to a) the *Thres_Mem* parameter and b) memory cell number.

In Table 3, the other classification results obtained by some well-known systems in the literature are shown along with the results of our system.

Table 3. The classification results of some methods in the literature, along with the results obtained by the AaA-response system.

Method	Test classification accuracy (%)
Grobian (rough)	100.00
AaA-response	**99.33**
SSV	98.00
C-MLP2LN	98.00
PVM 2 rules	98.00
PVM 1 rule	97.30
AIRS	96.70
FuNe-I	96.70
NEFCLASS	96.70
CART (dec. tree)	96.00
FUNN	95.70

As is seen from Table 3, the AaA-response system has out-performed many well-known effective methods with its high result, which is very close to 100%.

4.3. Applications on the Statlog Heart Disease dataset

The other dataset that is used for the investigation of the AaA-response's performance for real-world problems is the Statlog Heart Disease dataset. For this dataset, the k-fold CV method was used as a 5-fold CV in the division of the training-test sets. Again, as in the case of the Iris dataset, the *Num_peptides* and *Thres_B* parameters were fixed to the values of 250 and 0.6, respectively. However, for this dataset, unlike the Iris applications, the number of hidden layers was not changed during the experiments. Only 1 hidden layer was used, because the performance is decreased a great deal when more than 1 layer is used. The experiments were conducted to find the optimum *Thres_Mem* and *hidden node number* giving the highest classification accuracy. Because this dataset has 13 features, 13 input nodes are used in the input layer. The output layer, on the other hand, consists of the 1 node giving the class information of the presented data. As in the case of the Iris dataset, the applications are done for several values of the *Thres_Mem* parameter, giving different numbers of memory cells. The *Thres_Mem* values used and the produced memory cell numbers are shown in Table 4.

For each of these *Thres_Mem* values, the ANN architecture mentioned above with 1 hidden layer is used and the number of hidden nodes is changed to take the values of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, and 20. The change of the classification accuracy for a fixed *Thres_Mem* value ($Thres\_Mem = 0.999$) with respect to the hidden node number is shown in Figure 7, where it is seen that the classification accuracy first increases and then begins to decrease with the increasing hidden node number. The point at which the maximum accuracy is obtained is the optimum hidden node number for the system to learn the relations between the memory cells. This is the case for all of the *Thres_Mem* values.

The highest classification accuracies obtained with the optimum hidden node number for each *Thres_Mem* value are shown in Table 5.

Table 5 shows that the optimum hidden node numbers for the highest classification accuracies are 3–6 for each of the *Thres_Mem* values. This means that the system learns the relations of the memory cells with an ANN architecture that is not too complex. This results in obtaining output values in relatively a short time in the test processes.

Table 4. The produced memory cell number for the used *Thres_Mem* values for the application of the AaA-response system on the Statlog Heart Disease dataset.

Thres_Mem	Memory cell number
0.8	4
0.95	11
0.98	25
0.985	60
0.99	90
0.995	150
0.998	210
0.9985	234
0.999	270
0.9995	356
0.9997	396
0.9998	450
0.9999	480
0.99995	521
0.99997	588

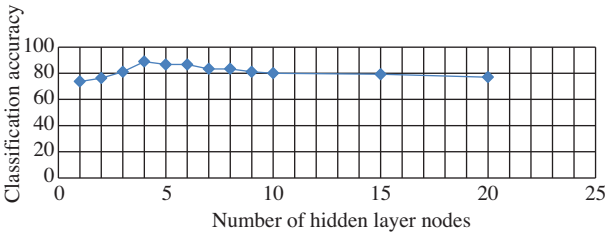

Figure 7. Change of the classification accuracy with respect to the hidden node number for *Thres_Mem* = 0.999.

Table 5. The highest classification accuracies obtained by the optimum hidden node numbers for each of the *Thres_Mem* parameter values in the applications of the AaA-response system on the Statlog Heart Disease dataset.

Thres_Mem	Number of memory cells	Highest classification accuracy (%)	Optimum hidden node numbers giving the highest accuracy
0.8	4	78.89	5, 6
0.95	11	79.62	6, 7
0.98	25	80.00	6
0.985	60	81.11	5, 6
0.99	90	84.44	5
0.995	150	87.03	5
0.998	210	88.51	4
0.9985	234	88.51	5, 6
0.999	270	88.89	4
0.9995	356	88.89	5
0.9997	396	90.37	3
0.9998	450	89.25	3
0.9999	480	88.51	3, 4, 5
0.99995	521	87.03	4
0.99997	588	86.67	5

Figure 8 presents the change of the classification accuracy with increasing memory cell numbers. Similar to the Iris applications, increase of the memory cell number from the optimum point causes over-fitting problems in the system and the performance worsens.

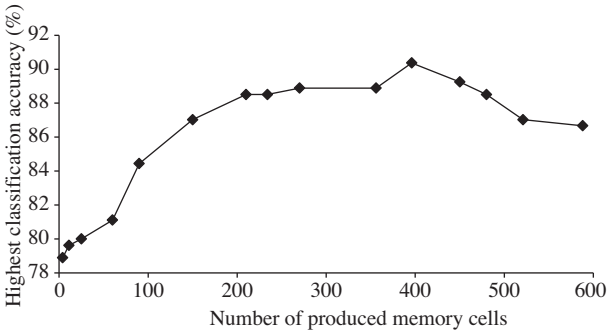

Figure 8. Highest classification accuracy with respect to the number of obtained memory cells in the application of the AaA-response system on the Statlog heart disease dataset.

As a summary, the best classification accuracy obtained by our developed system, AaA-response, for the classification of the Statlog heart disease dataset is obtained as the highest classification accuracy of 90.37%, obtained by:

Thres_B = 0.6, *Thres_Mem* = 0.9997, *Num_peptides* = 250, *produced memory cell number* = 396, *hidden layer number* = 1, *hidden node number* = 3.

To compare this accuracy with the results of other well-known methods in the literature, the results of these methods compared with ours are given in Table 6.

Table 6. The classification results of some methods in the literature, along with the results obtained by the AaA-response system.

Method	Test classification accuracy (%)
AaA-response	**90.37**
AIRS	84.50
Naive Bayes	83.60
AWAIS	82.59
Naive Bayes	81.48
BNND	81.11
C4.5	81.11
BNNF	80.96
Logistic regression	77.00
K*	76.70
IB1c	74.00
1R	71.40
T2	68.10
MLP+BP	65.60
FOIL	64.00
RBF	60.00
InductH	58.50

As seen from Table 6, the highest classification accuracy is obtained by the AaA-response system. Aside from this success, the difference in the classification accuracy between this system and the others is noteworthy.

With these successes in the Statlog Heart Disease and Iris datasets, the developed AaA-response system seems to attract the attention of artificial intelligence researchers for use in a very broad range of application fields.

5. Discussion

Inspired by the mechanisms in natural ISs, several systems have been developed in the field of AIS for different application areas, such as robotics, optimization, classification, virus detection, and scheduling. Although much ground has been covered in this field with some successful studies, it is not possible to say that the AIS is such an AI field that it contains methods that are more successful than the state-of-the-art methods. Several factors can be listed for this deficiency. For example, the processes of the IS have not yet been understood fully, not even by immunologists. This proves the complexity of the IS. One other factor is that the modeled systems are either too problem-oriented or too IS-oriented. In this study, an AIS system named the AaA-response was designed to present the AIS as a successful problem-solving system that can be preferred instead of the current state-of-the-art methods. The AaA-response system is a 2-stage learning system that can produce output values for the presented input samples. In the first stage of this system, memory cells were produced by modeling the adaptive immune response of natural ISs, but with a more realistic model using a different representation scheme. This stage learns its processes on the collective response of the B cells in the IS. That is to say, the response of the IS is formed by the collaborative responses of the produced B cells and this is modeled in the AaA-response system by forming memory B cells and detecting the relations between them with a second stage, which uses an ANN. When any input data are presented to the system, the first stage determines the best recognizing memory cells of their features and the ANN stage produces an output value by remembering the relation in this memory cell configuration.

The developed system was first applied to the Chainlink dataset to conduct the parameter analyses. While the system contains several parameters in its first and second stages, the only analyzed parameters are the *Thres_B* and *Thres_Mem* parameters of the first stage, because they are much more effective in determining the classification accuracy. While analyzing them, the other parameter of the first stage was a fixed value based on experience obtained from previous experimentations. Moreover, the parameters, learning strategy, and architecture of the ANN in the second stage were also not changed.

To see the performance of the system with real problems, the Iris and Statlog Heart Disease datasets were used for classification. Using a 3-fold CV for the Iris data, the applications were conducted by changing the parameter values to find the best classification result, and this result was obtained as 99.33%. This accuracy means that only 1 sample was misclassified; this is an important success for this nonlinear problem because this dataset contains 3 samples whose classifications are very hard. Among the systems using k-fold CV, the classification of 2 of those samples is a noticeable success. On the other hand, the optimum parameter values and optimum ANN architecture were searched for in the Statlog Heart Disease dataset and a classification accuracy of 90.37% was obtained by the optimum parameters. The accuracies of the well-known methods in the literature lie between 58.50% and 84.50%. When this is taken into consideration, the success of the AaA-response for the Statlog Heart Disease dataset becomes more pronounced.

Aside from reaching high classification accuracies, another advantage of the designed system over the previously developed AISs is that it is also appropriate for other application areas, like regression or optimization, in which real values should be obtained as output values. It makes our system a general-use artificial system that can produce the desired output values for the presented inputs. This property has not been addressed in the developed AIS systems so far.

Acknowledgments

This study was supported by the Scientific Research Projects of Selçuk University.

References

[1] L.N. de Castro, F.J. Von Zuben, "An evolutionary immune network for data clustering", Proceedings of the IEEE Brazilian Symposium on Artificial Neural Networks, pp. 84–89, 2000.

[2] J. Timmis, M. Neal, "A resource limited artificial immune system", Knowledge Based Systems, Vol. 14, pp. 121–130, 2001.

[3] J.D. Farmer, N.H. Packard, A.S. Perelson, "The immune system, adaptation, and machine learning", Physica, Vol. 22, pp. 187–204, 1986.

[4] J.E. Hunt, D.E. Cooke, "An adaptive and distributed learning system based on the immune system", Proceedings of the IEEE International Conference on Systems Man and Cybernetics, pp. 2494–2499, 1995.

[5] A. Watkins, AIRS: A Resource Limited Artificial Immune Classifier, MSc, Mississippi State University, Mississippi, USA, 2002.

[6] S.M. Garrett, "A paratope is not an epitope: implications for immune network models and clonal selection", Lecture Notes in Computer Science, Vol. 2787, pp. 217–228, 2003.

[7] S. Sahan, H. Kodaz, S. Güneş, K. Polat, "A new classifier based on attribute weighted artificial immune system (AWAIS)", Lecture Notes in Computer Science, Vol. 3280, pp. 11–20, 2004.

[8] V. Cutello, G. Narzisi, G. Nicosia, M. Pavone, "Clonal selection algorithms: a comparative case study using effective mutation potentials", Lecture Notes in Computer Science, Vol. 3627, pp. 13–28, 2005.

[9] S. Özşen, S. Kara, F. Latifoğlu, S. Güneş, "A new supervised classification algorithm in artificial immune systems with its application to carotid artery Doppler signals to diagnose atherosclerosis", Computer Methods and Programs in Biomedicine, Vol. 88, pp. 246–255, 2007.

[10] I. Aydin, M. Karakose, E. Engin, "Artificial immune classifier with swarm learning", Engineering Applications of Artificial Intelligence, Vol. 23, pp. 1291–1302, 2010.

[11] A.J. Graaff, A.P. Engelbrecht, "Clustering data in an uncertain environment using an artificial immune system", Pattern Recognition Letters, Vol. 32, pp. 342–351, 2011.

[12] G.C. Luh, C.Y. Lin, "PCA based immune networks for human face recognition", Applied Soft Computing, Vol. 11, pp. 1743–1752, 2011.

[13] X. Ji, Y. Liu, X. Yu, J. Wu, "Alternative combination of quantum immune algorithm and backpropagation neural network", Proceedings of the 7th International Conference on Natural Computation, pp. 66–70, 2011.

[14] University of California - Irvine, UCI Machine Learning Repository, available at http://archive.ics.uci.edu/ml/ (last accessed 22 May 2012).

[15] A.K. Abbas, A.H. Lichtman, Cellular and Molecular Immunology, 4th ed., Philadelphia, Saunders Publishing, 1994.

[16] L.N. de Castro, J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Heidelberg, Springer-Verlag, 2002.

[17] A.S. Perelson, G.F. Oster, "Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self–non-self discrimination", Journal of Theoretical Biology, Vol. 81, pp. 645–670, 1979.

[18] E. Micheli-Tzanakou, Supervised and Unsupervised Pattern Recognition, Boca Raton, Florida, USA, CRC Press, 1999.