An Artificial Immune System for Classification with Local Feature Selection

Grzegorz Dudek

***Abstract*—A new multiclass classifier based on immune system principles is proposed. The unique feature of this classifier is the embedded property of local feature selection. This method of feature selection was inspired by the binding of an antibody to an antigen, which occurs between amino acid residues forming an epitope and a paratope. Only certain selected residues (so-called energetic residues) take part in the binding. Antibody receptors are formed during the clonal selection process. Antibodies binding (recognizing) with most antigens (instances) create an immune memory set. This set can be reduced during an optional apoptosis process. Local feature selection and apoptosis result in data-reduction capabilities. The amount of data required for classification was reduced by up to 99%. The classifier has only two user-settable parameters controlling the global-local properties of the feature space searching. The performance of the classifier was tested on several benchmark problems. The comparative tests were performed using *k*-NN, support vector machines, and random forest classifiers. The obtained results indicate good performance of the proposed classifier in comparison with both other immune inspired classifiers and other classifiers in general.**

***Index Terms*—Artificial immune system, classification, dimensionality reduction, local feature selection, supervised learning.**

I. Introduction

Natural immune systems, as the defense system of animal organisms against pathogens, were the inspiration behind the artificial immune systems (AIS). The interest of researchers is generated by such immune system features as: recognition of antigen (AG) characteristics, pattern memorization capabilities, selforganizing memory, adaptation ability, immune response shaping, learning from examples, distributed and parallel data processing, multilayer structure and generalization capability. The application areas for AIS can be summarized as follows [1]:

1) learning (clustering, classification, recognition, robotic and control applications),
2) anomaly detection (fault detection, computer, and network security applications),
3) optimization (continuous and combinatorial).

The most important types of AIS are based on the concepts of negative selection, clonal selection, and the immune network. Negative selection AIS was introduced by Forrest *et al.* [2]. The idea is to generate change detectors and then remove those that recognize "self" elements. The remaining detectors can later be used to detect anomaly. This mechanism has been applied in network security [3], milling operations [4], fault detection and diagnosis [5], [6], network intrusion detection [7], DNA computer [8], and in the detection of changes made to computer programs by viruses.

Clonal selection describes the basic feature of adaptive immune response: only those cells that recognize the AG proliferate. In consequence, they are selected over those that do not. These clones have mutated from the original cell at a rate inversely proportional to the match strength. Fukuda, Mori and Tsukiyama first developed an algorithm that included clonal selection to solve computational problems [9], [10]: scheduling and resource-allocation optimization problems. Clonal selection was popularized by de Castro and Von Zuben, who developed an algorithm called CLONALG [11], which currently exists in two versions: for optimization and for pattern recognition. Another form of the clonal selection algorithm is artificial immune recognition system (AIRS) [12], which was developed from the AINE immune network [13]. Typical applications for clonal selection include the following [14]: unimodal, combinatorial, multimodal, and non-stationary function optimization [15], [16], initializing the centers of radial basis functions [17], various types of pattern recognition [18], graph coloring problems [19], multiple character recognition problems, automated scheduling [20], and document classification [21]. In [22] and [23], a clonal selection algorithm was applied to time series forecasting.

The immune (idiotypic) network theory was proposed by Jerne [24]. In artificial models based on this theory, immune cells can match other immune cells as well as AGs. This leads to the creation of a network between the immune cells. The recognition of an AG results in network activation and cell proliferation, whilst the recognition of a cell receptor by another cell receptor results in network suppression. Several immune network models have been developed [10], [25]–[28]. The CLONALG algorithm mentioned above was extended by employing the metaphor of the immune network theory and then applied to data clustering. This led to the development of the aiNet algorithm [29]. Another immune network theory inspired by AIS is AINE, which is applied to data clustering. A model combining the ideas of aiNet and AINE has been also proposed [30]. Apart from data clustering, immune network models have been applied to [14]: detecting gene promoter

Manuscript received August 16, 2010; revised February 20, 2011 and June 24, 2011; accepted October 7, 2011. Date of publication February 10, 2012; date of current version November 27, 2012.

The author is with the Department of Electrical Engineering, Czestochowa University of Technology, Czestochowa 42-200, Poland (e-mail: dudek@el.pcz.czest.pl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2011.2173580

sequences [27], data mining [31] and diagnosis [5]. The idiotypic network theory is no longer widely accepted by immunologists [32].

Another concept applied in AIS is the controversial immunological danger theory proposed by Matzinger [33]. In this approach, self-nonself dichotomy is replaced by danger-nondanger dichotomy. The Dendritic Cell Algorithm [34] inspired by this theory has distinct advantages when applied to real-time computer security problems, as it has very low CPU processing requirements and does not require extensive training periods.

This paper presents a supervised learning algorithm based on an AIS using clonal selection. Even though some AIS for classification have been developed, this model has a unique feature, it includes the local feature selection mechanism. The aim of feature selection is to reduce the dimension of the input vector by the selection of a feature (variable) subset which describes the object in the best manner and ensures the best quality of the learning model. In this process irrelevant, redundant and unpredictive features are omitted. Popular feature selection methods are global, i.e., they determine one feature set for all training data. But one can imagine that different features are important in different regions of the input vector space. The proposed approach allows the detection of many relevant feature sets (a separate relevant feature set is created for each learning point and its neighborhood). This method of feature selection is inspired by the binding of an antibody (AB) to an AG, which occurs between amino acid residues forming an epitope and a paratope. Only certain selected residues (so-called energetic residues) take part in the binding. This approach reduces the curse of dimensionality that affects many machine learning methods. The new algorithm is Artificial Immune System with Local Feature Selection (AISLFS).

This paper is organized as follows: in Section II, the biological inspirations behind AIS are discussed. In Section III the existing AIS for data classification are described, their limitations and some new inspirations for AIS are presented. In Section IV, the proposed classifier algorithm with a local feature selection procedure is defined. In Section V, we perform an empirical analysis study of the proposed classifier and compare the results to algorithms described in Section III as well as to other popular classifiers: the *k*-nearest neighbor (*k*-NN), support vector machines (SVM), and random forest (RF). In Section VI, an overview of the work is given.

II. Biological Inspirations

The immune system is a multifunction defense system which has evolved to protect animals against infections from viruses, bacteria, fungi, protozoa, and also worms and tumor cells. The immune systems of vertebrates consist of many types of proteins, cells, organs, and tissues which interact in an elaborate and dynamic network. The first barrier of defense against pathogens is the innate or non-specific immune system, which exists in an organism from its birth and does not adapt during its lifetime. It consists of cells and molecules quickly reacting to infection (e.g., phagocytes which "eat" bacteria). The second barrier is the adaptive (specific) immune system, which, unlike the innate system, demonstrates the ability to recognize and remember specific pathogens (immunological memory) and to improve its working in response to infection. This leads to quicker recognition and elimination of the pathogen that reinfect the organism.

The immune system operates on two levels: cell-mediated immunity and humoral immunity. The mechanisms of cell-mediated responses developed for pathogens that infect cells, whereas the mechanisms of the humoral responses developed for pathogens in body fluids. The immune system is able to distinguish self and nonself molecules (AGs).

The proposed AIS is inspired by the humoral responses and is able to recognize not only self and nonself AGs but many different types of AGs (many classes of patterns).

The cells responsible for AG recognition, lymphocytes, are equipped with surface receptors specific for a given nonself AG. When lymphocytes identify an invader, they bind to it and are activated to proliferation, which induces the generation of clones.

There are two main classes of lymphocytes in the adaptive immune system: B cells (bone cells) and T cells (thymus cells). T cells on the AG stimulation are involved in cell-mediated immunity. B cells in reaction to the AG contact proliferate and differentiate into effector cells (plasma cells), which generate and secrete a lot of specific ABs binding to the AGs.

In our algorithm the representatives of the immune system are ABs, which work as the recognition units. An AB as well as an AG belongs to one of the classes.

AB molecules have a Y-shaped structure [35], with two variable regions that bind to an AG. Within these variable regions some polypeptide segments, named hypervariable regions, show exceptional variability. These hypervariable regions, located at the tips of the Y, form a unique topography called a paratope corresponding to the antigenic determinant known as an epitope. The two variable regions contain identical AG-binding sites that are specific for only one type of AG. The amino acid sequences of the variable regions of different ABs are extremely variable. Therefore, the AB molecules in the body provide an extremely large repertoire of AG-binding sites.

In the proposed algorithm each AB is equipped with a specific receptor (each AB different) which allows recognizing an AG by its selected features. An AG represents the data instance (feature vector belonging to one of the classes). An epitope is formed of those features which are present in a paratope.

The source of the vast repertoire of AB molecules that can be synthesized by an individual lies in the way of encoding the amino acid sequences of the variable domains into DNA chains as well as the random selection and recombination of the gene segments. Information for the variable domains is present in the libraries of the gene segments that recombine at the level of the DNA. During the development of a B cell these genes are rearranged (independently of an AG) and undergo repeated rounds of random mutation (somatic hypermutation, depending on the AG) to generate B cells expressing structurally distinct receptors. Some of these B cells bind to AGs with increased affinity and undergo differentiation to generate plasma cells, which secrete large amounts of ABs of higher affinity (*clonal expansion*). This process is called the *affinity maturation*. As a result of clonal expansion, B cells operating as memory cells are also generated. Thanks to these cells the immune response is more effective and rapid if subsequent reinfection with the same AG occurs (*secondary immune response*).

The clonal expansion is used in our AIS to generate new recognition units which are optimized during training. The number of clones is the parameter of the algorithm. The affinity maturation leads to the increasing number of recognized AGs. During training the immune memory is improved and as a result AGs are recognized with the lower error.

The selective treatment of lymphocytes by AGs and the activation of only those, which can bind to them, is called *clonal selection*. During the creation of the new lymphocytes the cells of high affinity to self AGs are generated as well. Further development of these cells will lead to an attack from the immune system against the self cells. In order to prevent this, the lymphocytes that recognize the self cells are eliminated (*negative selection*).

The clonal selection is used in the algorithm to searching the paratope space to determine the best recognition units which form the immune memory. The concept of negative selection is not used here.

A protein antigenic molecule has parts on its surface (antigenic determinants or epitopes) against which a particular immune response is directed. Proteins are built from a repertoire of 20 amino acids which form chains which do not exist in nature as straight chains (called primary structure), but as folded whorls with complex loops (tertiary structure). Most ABs recognize a conformational epitope whose protein structure has a specific 3-D shape. The AG-AB interaction results from the formation of multiple non-covalent bonds. These attractive forces consist of [35]: hydrogen bonds, electrostatic bonds, van der Waals forces and hydrophobic forces. The strength of a non-covalent bond depends critically on the distance between the interacting groups. For a paratope, to combine with its epitope, the interacting sites must be complementary in shape, charge distribution, and hydrophobicity, and, in terms of donor and acceptor groups, capable of forming hydrogen bonds.

In the proposed AIS a paratope should be similar in shape to an epitope (not complementary, see Fig. 1) to ensure a strong bond, i.e., the features which form a paratope and an epitope should have similar values. Amino acid residues, which are on the surface of the molecule in the tertiary structure and which take part in the binding, represent the selected features of the pattern.

The strength of the interaction between an AG and an AB is loosely referred to as *affinity*. Because an AG can have many different epitopes on its surface (if so it is called multivalent), it can be bound to many different ABs. The combined strength of multiple bond interactions between an AB and an AG is called *avidity*. If similar or identical epitopes occur on different AGs, the AB generated against one of them will be *cross-reactive* with others.

The affinity is a function of the distance between a paratope and an epitope in the proposed AIS. High affinity (small distance) between selected AB/AG features indicates that the AG is recognized by the AB. An AG having many epitopes can be bound to many ABs. The avidity to an AG, which is calculated for each class c, is the total binding strength of all ABs of class c. An AB having the cross-reactivity threshold can recognize many AGs with similar epitopes.

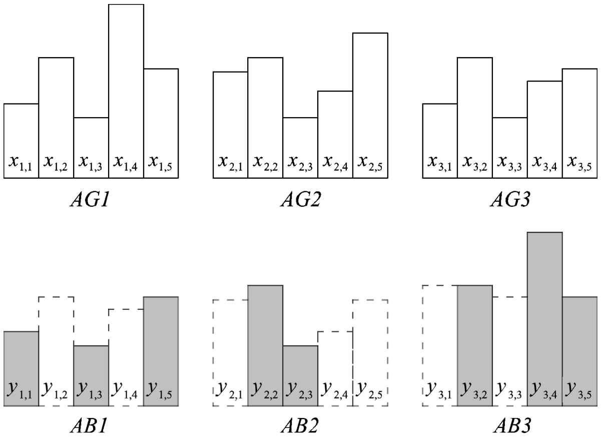
$x_{1,1}$ $x_{1,2}$ $x_{1,3}$ $x_{1,4}$ $x_{1,5}$
$x_{2,1}$ $x_{2,2}$ $x_{2,3}$ $x_{2,4}$ $x_{2,5}$
$x_{3,1}$ $x_{3,2}$ $x_{3,3}$ $x_{3,4}$ $x_{3,5}$
AG1
AG2
AG3
$y_{1,1}$ $y_{1,2}$ $y_{1,3}$ $y_{1,4}$ $y_{1,5}$
$y_{2,1}$ $y_{2,2}$ $y_{2,3}$ $y_{2,4}$ $y_{2,5}$
$y_{3,1}$ $y_{3,2}$ $y_{3,3}$ $y_{3,4}$ $y_{3,5}$
AB1
AB2
AB3

Fig. 1. Different antigens and antibodies (n = 5).

Antigenic proteins have structural epitopes consisting of 15–22 amino acid residues that constitute the binding face with an AB which has a similar number of residues taking part in the binding [36]. Each structural epitope has a functional epitope of about two to five residues that dominate the strength and specificity of binding with the AB. The remaining residues of a structural epitope provide supplementary interactions that increase the stability of an AG-AB complex. The binding energy of an AG-AB complex is primarily mediated by a small subset of contact residues (*energetic residues*) in the epitope-paratope interface. Based on the hypothesis, developed from molecular modeling of crystallized AG-AB complexes, that functional epitopes are represented by patches of surface-exposed nonself amino acid residues surrounded by residues within a 3Å radius, Duquesnoy used the term “eplet” [36], [37] to describe polymorphic residues within 3.0–3.5Å of a given sequence position on the molecular surface. Many eplets represent short linear sequences but others have residues in discontinuous sequence positions separated in the primary sequence but clustered together on the molecular surface by folding of the native protein. Thus, the functional epitopes/paratopes can generally be defined by small numbers of amino acid residues among all amino acid residues forming an AG/AB.

In our AIS by the energetic residues we mean the selected features of the patterns. The set of these features, which form a paratope and an epitope, ensure the best recognition properties of the recognition unit. The goal of our algorithm is to construct the immune memory composed of ABs with suitably shaped receptors that recognize correctly all AGs from the training set.

Since the mechanisms of cell-mediated responses are not used in our AIS, they are not described here. It is worth mentioning only that T cells do not react directly on AGs, like ABs, but on the complexes composed of a MHC molecule (expressed on the surface of antigen presenting cells) and a short peptide AG. The MHC molecule inside the cell takes a fragment of the synthesized protein and displays it on the cell surface. If the T cell recognizes the protein as nonself, it can kill the infected cell. T cell receptors have a similar structure to AB receptors and are characterized by a high degree of variability as well.

III. Artificial Immune Classification Systems: Review, Criticism, and Some Inspirations

Key features of AIS, such as feature extraction, recognition, and learning are very useful in classification and clustering tasks. The focus of early AIS research seems to have been on the development of unsupervised learning algorithms rather than the supervised or reinforcement kind. One of the first works to attempt to apply AIS to supervised learning concerns a real-world problem: the recognition of promoters in DNA sequences [27]. The results obtained are consistent with other approaches such as neural networks and Quinlan's ID3 and are better than the nearest neighboring algorithm. The primary advantages of AIS are that it only requires positive examples, and the patterns it has learnt can be explicitly examined. In addition, because it is selforganizing, no effort is required to optimize the system parameters.

Some of the artificial immune classification systems are described below. In the experimental part of this paper we compare results of these AIS to our AISLFS.

Carter proposed a pattern recognition and classification system called Immunos-81 [38]. This was created using software abstractions of T cells, B cells/AB, and their interactions. Artificial T cells control the creation of B cell populations (clones), which compete for recognition of "unknowns." An AG represents a single data instance and may have multiple epitopes (attributes, variables). An interesting feature of Immunos-81 is its potential ability to learn "on line," which refers to the ability to add new AG-types or classes to the AIS without having to rebuild the entire system. The learning process in Immunos-81 is rapid because it needs only one pass of the training data.

Some researchers have noted that Immunos-81 is highly complex and the description of this technique in [38] is somewhat incomplete or lacking in details and not sufficient to replicate [39], [40]. Given the description of Immunos-81's training procedure and the identification of potentially useful and desirable elements of the system, Brownlee implemented this method in three variants Immunos-1, Immunos-2, and Immunos-99 [40] with the goal of repeating the results observed in the original work. The first two are basic and naive implementations and the third one exploits those elements of Immunos-81 that appear beneficial and unique to the system, and integrates cell-proliferation and hypermutation techniques from other immune-inspired classification systems.

The most popular immune classification system is the AIRS [41], [42]. The goal of this algorithm is to develop a set of memory cells that can be used to classify data. Memory cells are evolved from a population of artificial recognition balls (ARBs). An ARB represents a number of identical B cells and is a mechanism employed to reduce duplication and dictate survival within the population. The AIRS maintains a population of memory cells and ARBs for each class of AGs. The first stage of the algorithm is to determine the affinity (based on the Euclidean distance) of memory cells to each AG of a certain class. The next stage is to identify the strongest ARBs, based on affinity to the training instance. They are used to create the established memory set used for classification. This is achieved via a resource allocation mechanism. The stimulation level of an ARB is calculated not only from the antigenic match, but also from the class of the ARB. This provides reinforcement for ARBs that are of the same class as the AG being learnt and that match well the antigenic pattern, in addition to providing reinforcement for those that do not fall into that class and do not match well with the pattern. Once the stimulation of an ARB is calculated, the ARB is allowed to produce clones which undergo mutation.

The specification of the AIRS algorithm is reasonably complex at the implementation level. Two versions of this algorithm were proposed: basic AIRS1 and revised AIRS2, which is simpler and has greater data reduction capability [42]. The parallel version of AIRS designed for distribution across a variable number of processes is presented in [43].

Another idea is the multiclass classifier M-NSA based on negative selection and its new version called multiclass iteratively refined negative selection classifier (MINSA) [44]. In this approach, the self cells are the immune system equivalent of the training patterns. The classifier solving C-classes classification problem is built of C-elements. Each of them contains one receptor set corresponding to one class. Receptors in a particular set get more stimulated by cells (patterns) belonging to classes not corresponding to the set and get less stimulated by self cells. Thus, an element contains an effective receptor set being the internal representation of a class corresponding to the element. The pattern is classified to the class corresponding to the least stimulated receptor set. The stimulation can be determined by taking into account the number of activated receptors or their average stimulation. The classification process with the use of the M-NSA algorithm consists of two phases: selection and detection (see [44] for details).

The AB-AG representation and affinity metric is a crucial parameter for developing competitive immune-inspired algorithms. The affinity measure is usually based on the distance measure between immune cell receptor and AG representing the feature vector. From a geometrical point of view, receptors and AGs are represented as points in n-dimensional space, with a contiguous recognition region surrounding each point to account for imperfect matching [45]. Such a problem representation, coming from Perelson's shape-space formalism [46], dominates all classical immune-inspired algorithms. Some researchers criticize this representation and show that the biological validity of this mathematical abstraction is a controversial issue [45], [47]. The dimensionality of the biological shape-space is likely orders of magnitude smaller than typical machine-learning datasets [48].

In [49], several undesirable properties of hyperspheres (as antibody recognition regions) in high dimensions are shown. The volume of a hypersphere tends to zero when dimensionality increases, and the entire volume of a hypersphere is concentrated immediately below the surface. This means that the recognition space (covered by ABs) is nearly zero. This phenomenon induces general fundamental limitations on the real-valued negative selection for high-dimensional problems. Another problem with high-dimensionality is that any metric defined across an increasing volume of the shape-space becomes increasingly meaningless as data points tend to become equidistant [45]. The two key assumptions in instance-based methods based on low-dimensional intuitions: 1) that there are dense regions in the space that can be generalized, compressed or sparsely represented, and 2) that the distance between points is a meaningful proxy for comparison, discrimination and localization, cannot be used in high-dimensional space because their validity is a rapidly decreasing function of dimensionality [45]. These all are examples of the curse of dimensionality: procedures that are analytically or computationally manageable in low-dimensional spaces can become completely impractical in high-dimensional spaces [50], [51].

Recently, McEwan and Hart [45] proposed an alternative representational abstraction in AIS considering both the biological and machine learning perspectives. They noted that an epitope is not a predefined object. It is an arbitrary discontinuous region on the 3-D surface of a molecule. It comes into being as an epitope by virtue of binding to a receptor, that is, in the context of a particular interaction. The epitope is now defined as a subset of surface correlated peptides, and the immune repertoire is not a population of centroids, prototypes, or support vectors, but an overcomplete dictionary of basis functions; an ensemble of weak learners. In a maximal simplification each cell receptor defines a different m-dimensional subspace. The system is composed of many simple classifiers with weak representational capabilities. Each classifier is represented by a separate immune cell. The search space for the immune repertoire is enriched to the space of classifiers, and the regression function becomes a weighted vote amongst an ensemble of classifiers. As in Boosting, a set of weak learners can be aggregated into an arbitrarily strong learning algorithm. An increase in representational power is achieved through the diversity of single classifiers (defined in low-dimensional subspaces) and an increase in stability through their integration.

Cohen *et al.* [52] noted that specificity is a property of a collective of cells and not of single clones. They use the term “degeneracy,” which refers to the capacity of any single antigen receptor to bind and respond to (recognize) many different ligands (poly-clonality, poly-recognition). In their concept specificity is not an intrinsic but emergent property of the immune system and the functional unit is not an individual clone but a clone collective. Emergent properties arise from collective interactions between individual components (immune agents). The specificity of the collective meta-response emerges from the web of mutual interactions that influences each cell, despite its degenerate receptor.

A number of properties of Cohen’s cognitive immune model can be useful computationally in AIS. These ideas, paradigms and processes that describe the functioning and behavior of the immune system were identified as possible areas of inspiration for novel AIS [53]. In [54], the authors began initial investigations into degenerate detectors for AIS. They presented the conceptual framework approach and built an abstract computational model in order to understand the properties of degenerate detectors. To investigate the idea of degeneracy and following the conceptual framework approach, in [55] a computational model to facilitate the investigation is proposed. The authors identified that it is possible to recognize patterns using such degenerate receptors, and when compared to a non-degenerate system, recognition appears quicker.

In the AIS described below in the next section the dimensionality of the problem is reduced by local feature selection. Moreover, the recognition region of each AB (strictly speaking the number of AGs of the same class in the AB recognition region) is maximized. This leads to better coverage of the recognition space. Activated ABs form an ensemble which is local, i.e., for different input points to be classified the ensemble created is composed of different ABs, which cover these points in different subspaces. The idea of many simple, small sized recognition elements which recognize patterns independently and affect the output in similar ways is close to the concept of the Cohen’s degeneracy. The assumed benefit of an AIS with degenerate detectors will be to provide greater scalability and generalization over existing AIS [53].

IV. Artificial Immune System for Classification with Local Feature Selection

A. Outline

The concept of energetic residues described in Section II was the inspiration behind the AIS for classification with local feature selection. The algorithm creates the immune memory composed of ABs with appropriately formed paratopes. A paratope is composed of such residues that ensure the correct classification of AGs lying in the AB recognition region. The resulting population of ABs recognizes correctly the classes of all training AGs. This algorithm is equipped with a mechanism for local feature selection. A different feature set is associated with each element of the classifier (AB), which ensures the correct classification of all training points (AGs) in the AB neighborhood.

An AG represents a single data instance. It is a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ of standardized features, which correspond to amino acid residues (Fig. 1). The AG belongs to one of C classes. The epitope is not fixed, it is formed from the selected residues and depends on the paratope. The selection of residues takes place during the creation of the immune memory. An AG can have many epitopes and can be bound to many ABs. The population of training/testing AGs is identical to the training/testing set of points.

The representatives of the immune system are ABs. An AB, built similarly to an AG, is a vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ of standardized features and belongs to one of C classes. The feature values of an AB and its class are determined during initialization at the start of the algorithm and do not change during training. The AB population is initialized by the training points in the same way as the AG population, so

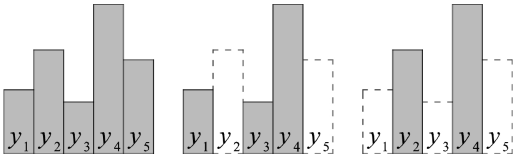
y_1 y_2 y_3 y_4 y_5
y_1 y_2 y_3 y_4 y_5
y_1 y_2 y_3 y_4 y_5

Fig. 2. Different paratopes of the same antibody (n = 5).

the AG and the AB populations are the same. Similarly to an epitope, a paratope is a set of selected features. An AB has only one paratope that is formed during training (Fig. 2). Each AB is characterized by the cross-reactivity threshold r adjusted individually during training and defining the AB recognition region in Ω-subspace (this is the subspace of the n-dimensional feature space with coordinates stored in Ω). This recognition region is represented by the hypersphere of radius r with center at the point $\mathbf{y} = [y_i]$, $i \in \Omega$. Inside this region there are AGs belonging solely to the same class as the AB. The AB is thus cross-reactive with AGs of the same class structurally similar to the AG initializing this AB.

The strength of binding of an AB to an AG (affinity) is dependent on the distance between vectors **x** and **y** measured using only selected features encoded in the paratope. The avidity to an AG, which is calculated for each class $c = 1, 2, \ldots, C$, expresses the strength of binding of all ABs of class c, which contain the AG in their recognition regions.

Fig. 1 shows three different AGs and ABs. The bar heights correspond to the feature values. The selected features (gray bars) form the paratops. *AG1* can be bound to all these three ABs (corresponding bars in the AB epitopes and *AG1* have similar heights), *AG2* can be bound to *AB2*, and *AG3* can be bound to *AB1* and *AB2*. The epitopes are dependent on the paratopes. *AG1* has three epitopes for these ABs: [x_1, x_3, x_5], [x_2, x_3] and [x_2, x_4, x_5], *AG2* has only one epitope: [x_2, x_3] and *AG3* has two epitopes: [x_1, x_3, x_5] and [x_2, x_3].

An AB causes secreting of clones, which inherit the features, paratope, and class from the parent AB. The number of clones is set *a priori* and is independent of the AB stimulation. The clones undergo hypermutations which modify their paratopes (set of features but not the feature values). The clone binding to the greatest number of AGs replaces the parent AB. This process is cyclic and leads to the creation of high quality ABs (memory cells), recognizing a large number of AGs by their selected features. In this combinatorial optimization process the number of AGs in the AB recognition region is maximized and the number of selected features (the paratope size) is minimized.

Optionally, the redundant memory cells, i.e., ABs which do not cause a decrease in the classifier efficiency, are eliminated. This leads to vast data reduction capabilities of the algorithm.

B. Algorithm

The pseudocode of the AISLFS classifier is given in Algorithm 1.

The symbols that appear in the following description of the algorithm are listed in Table I.

Step 1. The AG population (the standardized dataset) is divided into training and test parts in a proportion dependent on the cross-validation fold.

Algorithm 1 Pseudocode of the AISLFS Algorithm

```
Training
1. Loading of the training set of
   antigens.
2. Generation of the initial antibody
   population.
3. Do for each antibody.
  3.1. Do until the stop criterion is
       reached (clonal selection loop).
    3.1.1. Generation of clones.
    3.1.2. Clonal hypermutation.
    3.1.3. Calculation of the affinity
           of clones for antigens.
    3.1.4. Evaluation of clones.
    3.1.5. Selection of the best clone
           and replacing the parent
           antibody by it.
4. Apoptosis of the redundant antibodies
   (optionally).
Test
5. Antigen presentation and calculation
   of the avidity of antibodies for
   antigen.
6. Assignment of the highest avidity
   class to antigen.
```

TABLE I
LIST OF SYMBOLS

Symbol	Description
Ω	Set of selected features describing the AB paratope
Ψ	Set of stimulated ABs
$a(\mathbf{y}_k, \mathbf{x}_j, \Omega_k)$	Affinity measure of the kth AB for the jth AG
$A_c(\mathbf{x}^*)$	Avidity of activated ABs from class c for the presented AG
$c = 1, 2, \ldots, C$	Class number
$d(\mathbf{y}_k, \mathbf{x}_j, \Omega_k)$	Distance measure between the kth AB and jth AG
l	Tournament size in the tournament searching algorithm
n	Total number of features
N	Number of the training points
r	Cross-reactivity threshold
t	Maximum number of the successive iterations without result improvement (to stopping criterion of the clonal selection loop)
$\mathbf{v}$	n-element binary vector corresponding to the Ω-set used in hypermutation
$\mathbf{x} = [x_1, x_2, \ldots, x_n]$	Feature vector, antigen
$\mathbf{y} = [y_1, y_2, \ldots, y_n]$	Feature vector, antibody

Step 2. The initial AB population is created by copying all AGs from the training set (ABs and AGs have the same structure). This method of initialization prevents putting ABs in empty regions without AGs. The class labels of AGs are copied as well. The AB number is the same as the AG number.

Step 3.1. In this loop the kth AB paratope is constructed. This is a combinatorial optimization problem. The goal is to find the set of selected features Ω_k for which the criterion function (4) is maximized. To do so, a stochastic search mechanism called tournament searching is employed [56].

This simple method explores the solution space starting from an initial solution and generating new ones by perturbing it using a mutation operator. This operator switches the value of one randomly chosen bit (but different for each candidate solution) of the parent solution. When the set of new l candidate solutions is generated (l represents the tournament size), their evaluations are calculated. The best candidate solution (the tournament winner), with the highest value of the criterion function, is selected and it replaces the parent solution, even if it is worse than the parent solution. This allows us to escape from local maxima of the criterion function. If l is equal to 1, this procedure comes down to a random search process. On the other hand, when $l = n$ this method becomes a hill climbing method where there is no escape from the local maxima.

This algorithm turned out to be very promising in the feature selection problem, better than a genetic algorithm and simulated annealing, as well as deterministic sequential forward and backward selection algorithms [56].

The process of cyclic cloning and hypermutations is stopped when there is no result improvement in t successive iterations, i.e., the process is stopped after Tth iteration when the evaluation of clones generated in iterations from $T - t$ to T is not higher than the evaluation of the parent solution in iteration $T - t$.

Step. 3.1.1. Each AB generates l clones, where l depends only on the size of the solution space n. The number of clones $l = 1, 2, \ldots, n$ is the parameter controlling the global-local search properties of the algorithm. In the experimental part of this paper, reported in Section V, it is assumed that the number of clones is equal to round(*n/3*).

Step 3.1.2. The goal of hypermutation is to improve the AB paratope, i.e., defining the set of selected features Ω_k which ensures maximization of function (4). Each clone undergoes the hypermutation and a different Ω-set is created for it in the clonal selection loop of the algorithm. Initial Ω-sets are created at random.

Let $\mathbf{v}_k$ be an n-element binary vector corresponding to the kth clone. The components of this vector are determined as follows:

$$v_{k,i} = \begin{cases} 1, & \text{if } i \in \Omega_k \\ 0, & \text{otherwise} \end{cases} \qquad i = 1, 2, ..., n \tag{1}$$

where Ω_k is the set of amino acid residues forming the paratope of the kth AB, i.e., the set of the selected feature indices.

The indices of features corresponding to ones in $\mathbf{v}_k$ are elements of Ω_k. Hypermutation modifies a clone by switching the value of one randomly chosen bit of $\mathbf{v}_k$, different for each clone among l generated. As a result, the set Ω_k is changed, i.e., the clone paratope is somewhat modified at random.

Step 3.1.3. The affinity measure of the kth AB for the jth AG is evaluated based on the distance measure between them

$$d(\mathbf{y}_k, \mathbf{x}_j, \Omega_k) = \left(\sum_{i \in \Omega_k} |\mathbf{y}_{k,i} - \mathbf{x}_{j,i}|^p \right)^{\frac{1}{p}} \tag{2}$$

where $p = 1$ for the Manhattan metric and $p = 2$ for the Euclidean metric.

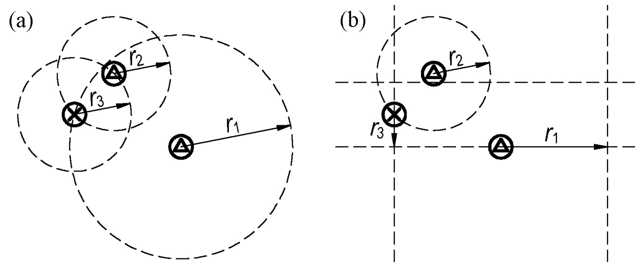

Fig. 3. Antibodies (small circles) and their recognition regions (dashed lines) for (a) $\Omega_k = \{1, 2\}$, $k = 1, 2, 3$ and (b) $\Omega_1 = \{1\}$, $\Omega_2 = \{1, 2\}$ and $\Omega_3 = \{2\}$. Antigens from class 1 are marked by triangles and from class 2 by crosses.

The distance is calculated by taking into account only selected features forming the paratope; the remaining features are omitted. Instead of (2) any other metric can be used which allows the features to be left out.

The affinity of AB for AG is inversely proportional to the distance between them in the feature space

$$a(\mathbf{y}_k, \mathbf{x}_j, \Omega_k) = \begin{cases} 0, & \text{if } d(\mathbf{y}_k, \mathbf{x}_j, \Omega_k) > r_k(\Omega_k) \text{ or } r_k(\Omega_k) = 0 \\ 1 - \dfrac{d(\mathbf{y}_k, \mathbf{x}_j, \Omega_k)}{r_k(\Omega_k)}, & \text{otherwise} \end{cases} \tag{3}$$

where $a(\mathbf{y}_k, \mathbf{x}_j, \Omega_k) \in [0, 1]$.

AB cross-reactivity threshold r_k is adjusted after a new AB is generated, such that an AB representing class c covers the region to the greatest extent possible without containing any AG of a different class. Thus, r_k is the distance between kth AB of class c and the nearest AG belonging to a different class. The hypersphere-shaped recognition regions of different ABs are defined in subspaces of different dimensions, determined by elements of Ω_k.

Fig. 3 presents AB recognition regions of three ABs in 2-D space where: (a) both features are selected for all ABs (i.e., $\Omega_k = \{1, 2\}$, $k = 1, 2, 3$), and (b) the first feature is selected for the first AB, both features are selected for the second AB and the second feature is selected for the third AB (i.e., $\Omega_1 = \{1\}$, $\Omega_2 = \{1, 2\}$, and $\Omega_3 = \{2\}$).

Step 3.1.4. After the threshold r_k is determined, the number of AGs contained in the AB recognition region is summed up. This number, depending on set Ω_k, informs about the AB representativeness and is used as the AB evaluation $F(\mathbf{y}_k)$, which is maximized in the process of clonal selection

$$F(\mathbf{y}_k) = |\{\mathbf{x}_j : d(\mathbf{y}_k, \mathbf{x}_j, \Omega_k) \le r_k(\Omega_k)\}| \to \max . \tag{4}$$

Step 3.1.5. The clone with the highest value of evaluation function (4) is selected and it replaces the parent AB, even if it is worse than the parent AB. If many AB have the same value of function (4), one of them, the one with the smallest paratope (the smallest number of selected features), is selected as the winner. This leads not only to maximization of the AG number in the AB recognition region but also to minimization of the selected feature number.

Step 4. This step is optional. In nature, most of the enormous number of lymphocytes generated during the immune response process die after the danger passes. But a relatively small number of the cells remain in the immune system as memory cells. The AISLFS immune memory consists of as many ABs as AGs. The recognition regions of adjacent ABs can overlap or one may contain another one. Some ABs can be removed if this does not deteriorate the classifier efficiency. These redundant ABs are eliminated in the deterministic sequential backward selection procedure [57], in which ABs are sequentially removed from the immune memory until the removal of next AB decreases the criterion. In each iteration the procedure selects one AB for elimination, without which the classifier accuracy is largest and not lower than the accuracy before this elimination. If the largest accuracy of the classifier is obtained for several ABs, the one of them with the smallest number of AGs in its recognition region is chosen for elimination. As a result, the final immune memory contains mostly ABs with large recognition regions, which ensure that the classification accuracy of the training set is 100%, as in the case of the original immune memory containing all ABs.

Steps 5 and 6. The test procedure corresponds to the secondary immune response where a new AG is presented to the trained immune memory. In the test procedure an AG of an unknown class, representing feature vector $\mathbf{x}^*$, is presented to the memory cells. Some ABs recognize this AG using their paratopes shaped in the training process, i.e., their recognition regions include this AG. Let Ψ be a set of these stimulated ABs. The set Ψ consists of subsets Ψ_c containing ABs belonging to one of classes $c = 1, 2, \ldots, C$. The avidities of ABs from the sets Ψ_c for the presented AG are determined. These avidities express the strength of the binding of ABs of class c to the AG. Three methods of avidity calculation are proposed. The first one relies on summing up ABs in sets Ψ_c

$$A_c^1(\mathbf{x}^*) = |\Psi_c(\mathbf{x}^*)|. \tag{5}$$

The second method relies on summing up the AB affinities in sets Ψ_c

$$A_c^2(\mathbf{x}^*) = \sum_{\mathbf{y}_k \in \Psi_c(x^*)} a(\mathbf{y}_k, \mathbf{x}^*, \Omega_k). \tag{6}$$

The third method uses the probabilistic OR operation to the AB affinities in sets Ψ_c

$$A_c^3(\mathbf{x}^*) = s_{m_c}(\mathbf{x}^*) \tag{7}$$

where m_c is the number of ABs in Ψ_c, and s_{m_c} is the probabilistic OR (also known as the algebraic sum) defined recursively as follows:

$$s_{m_c}(\mathbf{x}^*)=a(\mathbf{y}_{m_c}^c, \mathbf{x}^*, \Omega_{m_c}^c) + s_{m_c-1}(\mathbf{x}^*) - a(\mathbf{y}_{m_c}^c, \mathbf{x}^*, \Omega_{m_c}^c) s_{m_c-1}(x^*) \tag{8}$$

where $\mathbf{y}_{m_c}^c \in \Psi_c, s_1(\mathbf{x}^*) = a(\mathbf{y}_1^c, \mathbf{x}^*, \Omega_1^c)$.

The class of the highest avidity is assigned to the AG. If no AB is stimulated by the AG ($\Psi = \emptyset$), the class of the AG is unrecognized. If for several sets Ψ_c the avidity is equal, it can be assumed that the AG belongs to the most numerous class in Ψ for avidity definitions (6) and (7), or to the most numerous class in the set of all memory cells for definition (5).

C. Discussion

It is worth noticing that ABs mature independently of each other (loop 3 of Algorithm 1) and the clone evaluation function (4) is based on the number of AGs of the same class in the recognition regions of ABs, not on the individual AB discriminatory abilities (classification accuracy). The final decision about the class of an AG to be recognized is made collectively by stimulated ABs after the immune memory is created. This method of decision making resembles the concept of combining individual classifiers in ensembles, to generate more certain, precise and accurate results. The random subspace method of combining models [58] is a good reference for AISLFS. The difference between these two approaches is that in AISLFS the feature subsets are not created by random and the component models (ABs) with different feature subsets are local not global.

In comparison to other AIS, in AISLFS the optimization problem is shifted from the AB location optimization in the continuous n-dimensional feature space to the optimization of the feature subspaces, where the search space is binary. In this optimization problem there are two criterions: 1) the number of AGs of the same class in the AB recognition region which is maximized, and 2) the number of features which is minimized. The maximization of the first criterion is a priority. Criterion 2) is the secondary one. It concerns solutions which are equally evaluated using criterion 1). Both criterions lead to better coverage of the recognition space. The way in which the feature space is covered is specified below. AB covers the whole subspace W and the hypersphere defined in the subspace V where:

1) V and W are complementary subspaces of the n-dimensional Euclidean space U with Cartesian coordinates $\Theta = \{1, 2, \ldots, n\}$;
2) V is the m-dimensional subspace with coordinates $\Omega \subseteq \Theta$ (these coordinates form the paratope), W is the $(n–m)$-dimensional subspace with coordinates $\Theta \backslash \Omega$;
3) the hypersphere in V has a radius of r and center at the point $\mathbf{y} = [y_i]$, $i \in \Omega$.

The possible AB recognition regions for $n = 3$ in Fig. 4 are shown. (In this picture the nearest AG of different class than AB, marked with triangles, is the same AG in all subspaces. But it can be different in different subspaces.) The regions are bigger in U for smaller Ω-subspaces. Bigger regions, containing more AGs of the same class, are favored in optimization process. It means that the maximization of the primary criterion leads to the dimensionality reduction as well as the minimization of the secondary criterion.

D. Runtime Complexity Analysis

The runtime complexity of the AISFLS algorithm is as follows.

Step 1: Initialization of the AG population by copying N n-dimensional datapoints: $O(Nn)$.

Step 2: Initialization of the AB population by copying N n-dimensional datapoints: $O(Nn)$, initialization of the AB paratopes by random: $O(Nn)$.

Step 3: Do for each of N ABs: $O(N)$.

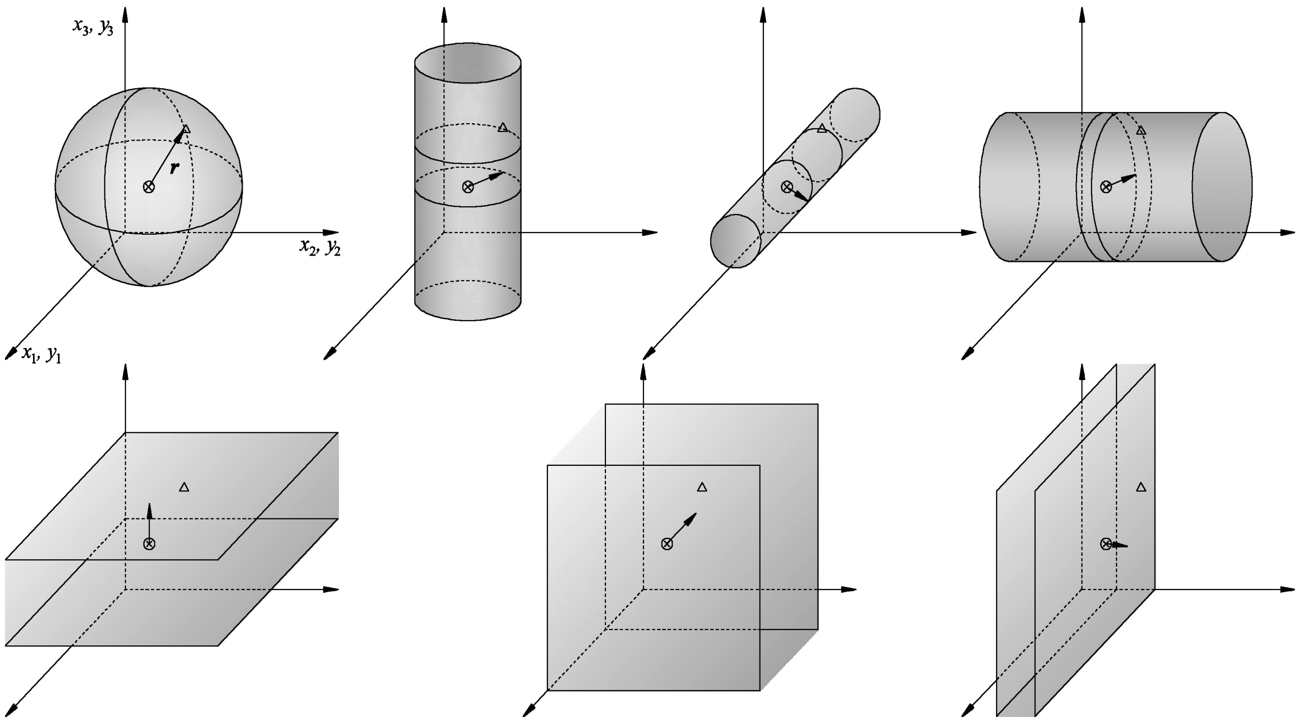

Fig. 4. Antibody (small circle) and its possible recognition regions in R^3. Triangle denotes the nearest AG of different class than AB.

Step 3.1: Do until the stop criterion is reached (T times): $O(T)$.

Step 3.1.1: Generation of l clones by copying the parent AB: $O(ln)$.

Step 3.1.2: Hypermutation of each clone: O(ln).

Step 3.1.3: Affinity calculation between clones and AGs: $O(lNn)$.

Step 3.1.4: Evaluations of clones, i.e., selection of the nearest AG of different class than the clone and determination of the clone cross-reactivity thresholds: $O(lN)$, determination of the number of AGs in the clone recognition regions: $O(lN)$.

Step 3.1.5: Selection of the best clone and replacing the parent AB by it: $O(l)$.

Given this, the asymptotic on the training routine without apoptosis would be $O(Nn + \sum_{i=1}^{N} T_i (ln + lNn + lN + l)) = O(\sum_{i=1}^{N} T_i lNn)$.

The apoptosis process consists of:

Step 4.1: Preparation of an $N \times N$ table with affinities of ABs for AGs: $O(N^2n)$.

Step 4.2: Sequential backward selection loops: $O(N^2)$.

Step 4.2.1: Determination of the classes and affinities of stimulated ABs for each AG: $O(N^2)$.

Step 4.2.2: Calculation of the avidities of ABs from each class for each AG and determination of the AG class: $O(N^2C)$.

This gives us a running time of apoptosis process: $O(N^2n + N^2(N^2 + N^2C)) = O(N^2n + N^4C)$.

The test procedure for M test points and N_A ABs (Steps 5 and 6) consists of initialization of the test AG population by copying M n-dimensional test datapoints: $O(Mn)$, preparation of an $N_A \times M$ table with affinities of ABs for AGs: $O(N_AMn)$, determination of the classes and affinities of stimulated ABs for each AG: $O(N_AM)$, calculation of the avidities of ABs from each class for each AG and determination of the AG classes: $O(N_AMC)$. The overall runtime complexity of the test procedure will be $O(Mn + N_AMn + N_AM + N_AMC) = O(N_AM(n+C))$. When apoptosis is performed usually $N_A << N$, and without apoptosis $N_A = N$.

TABLE II
DESCRIPTION OF DATA USED IN EXPERIMENTS

Dataset	Size	Features	Classes
Ionosphere	351	34	2
Glass	214	9	6
Breast Cancer	699	9	2
Iris	150	4	4
Wine	178	13	3
Diabetes	768	8	2
Heart Statlog	270	13	2
Sonar	208	60	2
Cleveland	303	13	2

The runtime complexity of AISLFS is determined by N, T, l, n, and C. T is not fixed, dependent on several factors: n, l, t and optimization problem complexity. The apoptosis process is the most costly of these three procedures. The most costly, frequently used operation in each of these phases is the distance calculations between ABs and AGs.

V. Experimental Results

The proposed AIS was verified on several test problems of data classification. Benchmark datasets, described in Table II, were taken from the UCI repository [59]. The features

TABLE III

Classification Results of AISLFS with Euclidean Metric

Dataset	$Acc\ A_c^1$	$Acc\ A_c^2$	$Acc\ A_c^3$	$\sigma_{Acc}\ A_c^1$	$\sigma_{Acc}\ A_c^2$	$\sigma_{Acc}\ A_c^3$	$AccAp\ A_c^3$	$NUAGAp\,\%$	NMC	$CR_f\,\%$	$CR_{AB}\ \%$	$CR_t\,\%$
Ionosphere	93.78	93.89	93.98	0.40	0.55	0.60	92.60	1.43	16.20	23.90	5.13	1.23
Glass	70.79	74.51	73.38	1.38	1.92	2.21	71.08	10.24	39.40	38.07	20.41	7.77
Breast Cancer	96.35	96.42	96.11	0.21	0.26	0.31	94.28	2.86	30.00	45.51	4.76	2.17
Iris	94.67	95.44	94.80	0.58	0.53	0.75	94.67	0.67	8.90	44.92	6.59	2.96
Wine	97.50	97.39	97.33	0.46	0.50	0.49	92.75	2.78	5.40	39.07	3.35	1.31
Diabetes	73.14	74.01	72.84	0.86	0.90	0.89	70.18	6.26	123.50	44.06	17.85	7.86
Heart Statlog	81.67	81.49	80.74	0.86	1.08	1.21	79.63	2.59	32.70	41.23	13.46	5.55
Sonar	82.62	87.79	88.10	1.51	1.16	1.18	80.79	2.40	13.20	25.52	7.02	1.79
Cleveland	81.77	81.92	81.18	0.79	0.95	1.17	77.47	6.30	35.90	40.86	13.15	5.37

where *Acc* is mean accuracy of the AISLFS classifier with avidity A_c^1, A_c^2, or A_c^3 in 30 training sessions, σ_{Acc} standard deviation of accuracies in 30 training sessions, *AccAp* accuracy of the AISLFS classifier with apoptosis, *NUAGAp* percentage of AGs unrecognized by the AISLFS classifier with apoptosis, *NMC* mean number of memory cells after apoptosis, CR_f mean compression ratio of the feature number, CR_{AB} mean compression ratio of AB number after apoptosis, CR_t mean total data compression ratio $CR_f{\cdot}CR_{AB}$ after apoptosis.

TABLE IV

Classification Results of AISLFS with Manhattan Metric

Dataset	$Acc\ A_c^1$	$Acc\ A_c^2$	$Acc\ A_c^3$	$\sigma_{Acc}\ A_c^1$	$\sigma_{Acc}\ A_c^2$	$\sigma_{Acc}\ A_c^3$	$AccAp\ A_c^3$	$NUAGAp\,\%$	NMC	$CR_f\,\%$	$CR_{AB}\,\%$	$CR_t\,\%$
Ionosphere	94.19	94.16	94.37	0.44	0.45	0.49	92.32	3.42	14.10	28.79	4.46	1.28
Glass	70.56	75.42	74.76	1.72	1.61	1.56	65.41	13.12	37.00	40.45	19.17	7.75
Breast Cancer	96.37	96.17	96.37	0.24	0.23	0.32	94.56	2.58	29.30	47.50	4.65	2.21
Iris	94.58	95.71	94.58	0.45	0.38	1.00	94.67	0.67	9.20	45.85	6.81	3.12
Wine	97.54	97.71	97.76	0.56	0.53	0.58	95.00	1.67	4.60	41.52	2.86	1.19
Diabetes	73.69	74.21	73.20	0.86	0.90	0.94	69.01	7.94	123.00	44.75	17.77	7.95
Heart Statlog	81.57	81.77	80.59	0.92	1.03	1.63	74.81	4.44	32.10	42.74	13.21	5.65
Sonar	83.83	87.76	88.03	1.19	1.09	1.10	78.43	7.19	12.90	29.20	6.86	2.00
Cleveland	81.76	81.97	81.47	0.71	0.96	0.95	77.17	4.00	37.90	42.58	13.88	5.91

in the datasets were standardized to zero-mean and unit-variance.

The distances between ABs and AGs were measured using the Euclidean and the Manhattan metrics. 10-fold cross-validation was used as a validation procedure for performance estimation. The tests were carried out in two variants: 1) applying the AISLFS classifier without apoptosis and with three methods of avidity calculation (5), (6), and (7) (in this variant 30 training sessions were performed), and 2) applying the AISLFS classifier with apoptosis and avidity calculation according to (7). The number l of clones generated in loop 3.1 was equal to round$(n/3)$ and the number of cycles in this loop without the result improvement t was equal to 10. These values of the AISLFS parameters were adjusted in the preliminary tests.

The results of classification are presented in Tables III and IV, where the compression ratio of the feature number CR_f, the compression ratio of the AB number after apoptosis CR_{AB} and the total compression ratio CR_t are also shown. The compression ratio is defined as follows:

$$CR = \frac{SN}{TN}100\% \tag{9}$$

where SN is the number of selected features or ABs and TN is the total number of features or ABs, respectively.

The accuracy of the AISLFS classifier without apoptosis depends on the method of the individual AB decision aggregation, which in turn depends on the avidity definition, but it is hard to identify one, best method for the avidity calculation. In many cases the difference between classifier accuracies for different avidity calculations is small (1–2 percentage points), but in the case of the Glass and Sonar datasets the difference reaches 2.59–5.48 percentage points in favor of the avidity definitions (6) and (7). For some datasets there were some, but not many, unrecognized test datapoints (Glass: 1%, Breast Cancer: 0.006%, Iris: 0.02%, Diabetes: 0.18%, Heart Statlog: 0.04%, Cleveland: 0.02%).

When only maximization of function (4) without minimization of the feature number was used as a winner selection criterion in Step 3.1.5 of Algorithm 1, the similar accuracies are observed but the compression ratios of the feature number are higher: from 2.06 to 13.78 percentage points.

An important feature of the AISLFS is the data reduction capability. As shown in Tables III and IV, AISLFS is able to reduce the amount of data needed to classify a given data set by 52–76% without apoptosis and by 92–99% with apoptosis.

When apoptosis is used to reduce the final number of memory cells, one can observe a decrease in the classifier accuracy on the test sets up to 9.60 percentage points (on average 3.60 percentage points) of the accuracy without apoptosis, depending on the dataset. The number of unrecognized AGs increases in these cases and varies from 0.67 to 13.12%. This is because the training set is not always representative and after apoptosis the small number of memory cells does not accurately cover the regions of the input space in which the test points lie (the decision surface of the classifier is modeled roughly).

The memory cells after apoptosis for the Iris dataset are presented in Tables V and VI. In this experiment, the classifier

TABLE V
MEMORY CELLS AFTER APOPTOSIS FOR IRIS DATASET USING AISLFS CLASSIFIER WITH EUCLIDEAN DISTANCE

ABNumber	*F1*	*F2*	*F3*	*F4*	*Class*	*r*	*NAG*
1	0.1892	–	0.7602	–	2	0.0567	1
2	–	−0.5858	1.0436	1.3121	3	0.8670	29
3	–	0.3367	0.4202	0.3948	2	0.6294	15
4	–	–	0.0801	−0.1293	2	0.8145	47
5	–	−1.2777	–	0.6569	3	0.2621	3
6	–	–	0.7035	0.3948	3	0.0567	2
7	–	–	1.2136	1.1811	3	0.7314	44
8	–	–	1.6670	–	3	0.9068	34
9	–	–	–	−1.1776	1	0.9172	50

where *F1–F4* are feature values, *NAG* is the number of AGs detected by AB.

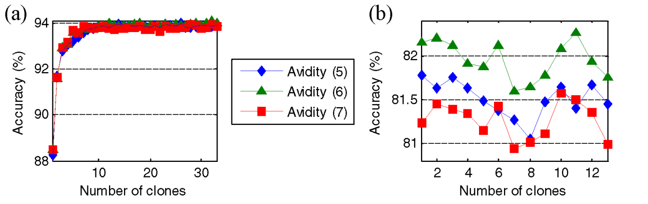

Fig. 5. Accuracy of the AISLFS classifier depending on the number of clones for (a) ionosphere and (b) Cleveland data.

learnt on the whole dataset, so the accuracy was 100%. As Tables V and VI show, each point of class 1 is recognized by only one AB using only one, the fourth feature. Classes 2 and 3 are not separated as easily, so more ABs are needed to recognize them.

A. Choice of Parameter Values

Theoretically, the higher the value of t, the deeper exploration of the searching space and, in consequence, the better results. A lower value of the number of clones l makes the searching process more stochastic, resistant to local maximum traps but slower. In practice, the values of the parameters should be a compromise between computation time and accuracy of classification. Accurate calculation of the optimal parameter values, ensuring the maximum accuracy, is not always a good solution (or even possible) due to the rough surface of the goal function $Acc = f(parameters)$ (see Figs. 5, 6 where mean accuracies over 30 training sessions depending on the parameter values are shown). This is caused by the inner optimization method, tournament searching here, which does not ensure best results when the goal function is multimodal. However the stochastic nature of the tournament searching increases the probability of finding a global optimum, there is no certainty that it will be found. This problem affects many machine learning algorithms, e.g., in the case of the multilayer perceptron where the gradient based, suboptimal inner optimization methods are applied, the learning results are sensitive on the starting weight values. Another reason not to tune the parameters in order to achieve the highest performance is the fact that optimization on the training set often does not ensure best results on the test set. This is due to an insufficient number of learning points in relation to the number of features.

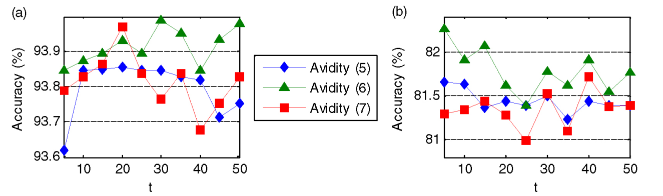

Fig. 6. Accuracy of the AISLFS classifier depending on the parameter t (number of cyclical cloning and hypermuations without result improvement) for (a) ionosphere and (b) Cleveland data.

TABLE VI
MEMORY CELLS AFTER APOPTOSIS FOR IRIS DATASET USING AISLFS CLASSIFIER WITH MANHATTAN DISTANCE

AB number	*F1*	*F2*	*F3*	*F4*	*Class*	*r*	*NAG*
1	0.1892	–	0.7602	–	2	0.0567	1
2	0.6722	–	1.0436	1.3121	3	1.3575	30
3	–	0.1061	0.3635	0.2638	2	0.9815	23
4	–	−1.2777	1.1569	0.7880	3	0.9032	6
5	–	–	0.7035	0.3948	3	0.0567	2
6	0.3100	−0.1245	–	0.7880	3	0.5241	5
7	–	–	−0.0332	−0.2603	2	1.3706	47
8	0.0684	–	0.2501	–	2	0.4608	16
9	–	–	1.0436	1.5742	3	1.2396	44
10	–	–	–	−1.1776	1	0.9172	50

TABLE VII
SENSITIVITY INDICES SI AND DETERIORATION OF RESULT INDICES DRI FOR MODEL WITH AVIDITY (6) AND EUCLIDEAN DISTANCE

Dataset	SI_t %	SI_l %	DRI_t %	DRI_l %
Ionosphere	0.15	5.91	0.12	0.19
Glass	0.81	2.08	0.73	0.89
Breast Cancer	0.10	0.22	0.08	0.08
Iris	0.53	0.28	0.28	0.28
Wine	0.50	0.37	0.14	<0.01
Diabetes	0.83	0.95	0.00	0.29
Heart Statlog	1.46	1.28	0.72	0.65
Sonar	0.65	0.74	0.46	0.64
Cleveland	1.06	0.81	0.42	0.43

The influence of the parameter values (l and t) on the classifier accuracy is limited. This is shown in Table VII where the values of sensitivity indices are presented. These indices are calculating using

$$SI = \frac{D_{\max} - D_{\min}}{D_{\max}} 100\% \qquad (10)$$

where $D_{\min}$ and $D_{\max}$ represent the minimum and maximum output values (accuracies), respectively, resulting from varying the input (parameter value) over its entire range.

The parameters were changed in this analysis as follows: $t = 5, 10, \ldots, 50$, at the fixed value of $l = \text{round}(n/3)$ (sensitivity index in this case is denoted by SI_t), and $l = 1, 2, \ldots, n$, at the fixed value of $t = 10$ (sensitivity index in this case is denoted by SI_l).

When in (10) we replace $D_{\min}$ with the result for $t = 10$ and $l = \text{round}(n/3)$, we can evaluate our choice of parameter values used in experiments. These new indices (denoted by DRI_t and DRI_l in Table VII) inform about the deterioration

TABLE VIII

Accuracies of Classification Using Different AIS Based Classifiers

Dataset	AISLFS	1	Immunos-2	99	AIRS1	AIRS1	AIRS1	AIRS2	AIRS2	Parallel	Parallel	M-NSA	MINSA
		[40]	[40]	[40]	[41]	[60]	61	[42]	[61]	AIRS [43]	AIRS[61]	[44]	[44]
Ionosphere	94.37				94.90	87.04	84.44	**95.60**	85.13		84.44		
Glass	**75.42**												
Breast Cancer	96.42	86.14	68.74	82.16		**96.84**	96.28		96.17		96.47	96.37	96.51
Iris	95.71	**97.13**	97.00	97.33	96.70	95.26	94.67	96.00	95.00	94.67–96.13	95.60	95.33	96.00
Wine	97.76												
Diabetes	74.21				74.10	69.70	71.60	74.20	70.87	72.98–**74.49**	70.96	70.44	72.00
Heart Statlog	**81.77**						78.15						
Sonar	**88.10**	67.93	61.64	68.51	84.00	69.81	67.03	84.90	66.01	83.65–85.63	64.71		
Cleveland	**81.97**	80.99	80.53	81.32		80.79			79.14		79.57	<70	<70

Best results are in bold.

TABLE IX

Classification Results of AISLFS, *k*-NN, SVM, and RF Classifiers

Dataset	AISLFS	*k*-NN	SVM	RF
Ionosphere	94.37 (0.49)	86.16 (0.85)	**95.22** (0.39)	93.45 (0.42)
Glass	75.42 (1.61)	69.63 (1.38)	68.22 (2.03)	**78.81** (1.40)
Breast Cancer	96.42 (0.26)	96.74 (0.31)	**97.59** (0.27)	96.15 (0.27)
Iris	95.71 (0.38)	95.40 (0.83)	95.07 (0.51)	95.09 (0.71)
Wine	97.76 (0.58)	96.42 (0.61)	98.54 (0.54)	98.10 (0.31)
Diabetes	74.21 (0.90)	75.15 (0.65)	76.70 (0.48)	76.68 (0.53)
Heart Statlog	81.77 (1.03)	82.96 (1.29)	**84.26** (0.73)	82.74 (0.88)
Sonar	**88.10** (1.18)	85.95 (1.47)	84.44 (1.23)	83.43 (1.16)
Cleveland	81.97 (0.96)	82.56 (0.94)	83.69 (0.56)	82.71 (0.79)

of result caused by the heuristic choice of parameter values. However the deterioration of results is observed but the *DRI* values do not exceed 1%.

Taking this all into account, the heuristic way of setting the parameter values, based on trial and error method, is suggested.

B. Comparative Study

In Table VIII the results for other AIS based classifiers, described in Section III, are shown. Referring to other AIS classifiers, the results obtained by the AISLFS place it among the best methods.

The results (accuracies and their standard deviations in 30 training sessions) for other popular classifiers: the *k*-NN, SVM, and RF in Table IX are presented. The experiments were performed in MATLAB 7.11 using functions from the Maltab toolbox for SVM and the MATLAB implementation of Breiman and Cutler's RF from http://code.google.com/p/randomforest-matlab/. The Gaussian radial basis function kernels with the optimized values of their widths σ were used in SVM, and the quadratic programming was applied to find the separating hyperplanes. The number of trees grown was 500 in RF and the number of features sampled for splitting at each tree node was equal to the square root of the total number of features. The default function settings were used except those described above. The algorithm parameters k in *k*-NN and σ in SVM were determined using 10-fold cross-validation procedure. 30 training sessions were performed.

In order to indicate the best classifier we check if the difference between the classifier accuracies is statistically significant using the *t*-test for equality of the means, Wilcoxon rank sum test for equality of the medians and two-sample Kolmogorov–Smirnov test for equality of distribution functions. The 5% significance level is applied. The null hypotheses are that the most accurate classifier has the same mean accuracy (median or distribution) as the second most accurate classifier. Rejection of all null hypotheses indicates the best classifier for a given dataset. The best results which are significantly better than others are in bold in Table IX.

In most cases the best classifier is the SVM but the AISLFS is not much worse (in the worst case its accuracy is lower than the SVM accuracy by 2.49 percentage points). However for the Glass dataset the AISLFS outperform SVM in accuracy by 7.20 percentage points and for the Sonar dataset by 3.66 percentage points.

VI. Conclusion and Further Work

A new immune inspired general purpose classifier based on the concept of energetic residues was introduced. The classifier is composed of memory cells (ABs) working as recognition units. Each AB recognizes an input point (AG) to be classified by selected features, different for different ABs. Consequently, the feature selection is local, i.e., only the features which contain most information about membership to the class represented by the memory cell, in the region where this memory cell is located, form its paratope. The final class is determined by combining answers of stimulated ABs (an ensemble of ABs). In order to reduce the number of memory cells, the apoptosis is introduced, but this reduces the classifier efficiency and increases the number of unrecognized test points. Both local feature selection and apoptosis mean that the classifier is capable of performing data reduction by up to 99%.

Even though ABs are evaluated independently of each other, and the evaluation function is based not on the individual AB discriminatory abilities but on the number of AGs of the same class in the AB recognition regions, this recognition mechanism proved to be quite effective in data classification.

The AISLFS algorithm has only two user defined parameters: the number of clones and the number of cyclic cloning and hypermuations without improvement in results (as the stop criterion). Such a small number of parameters is unusual in AIS and makes AISLFS easy to use. The AISLFS classifier is rather stable: no drastic changes in classifier accuracy over a wide range of parameter values were observed. The heuristic trial and error method is recommended to set the parameter values.

The AISLFS classifier does quite well in comparison with both other immune inspired classifiers and other classifiers in general. The power of AISLFS is in its unique embedded approach to the local feature selection. The local dimensionality reduction property distinguishes it from other classifier solutions. The data reduction capability of AISLFS is its other important feature.

The AISLFS algorithm has a parallel structure. The ABs can be processed in parallel, independently, as well as the population of clones. The parallel implementation of the algorithm will speed up the learning process considerably. The maximum acceleration through the parallel implementation is the product of the number of learning points and the tournament size.

Possibly some kind of interaction between ABs (like in idiotypic networks) should be introduced in further work on AISLFS development, so as not to generate a new AB in a region covered by another AB of the same class. This can prevent the creation a big set of memory cells, especially in "easy" regions of the feature space with AG of the same class, and can replace the time consuming apoptosis process. The strength of the natural immune system lies in generating a huge number of cells with different receptors, but our goal is not to replicate all the mechanisms of the natural immune system but to create an accurate, simple and easy to use artificial recognition system.

In the presented approach the antibody positions in the feature space are fixed: one antibody lies on one antigen. The release of antibodies, i.e., allowing the antibodies to change their positions across the feature space and locate themselves in optimal regions, could improve the results and lead to better data compression. This can be achieved by introducing an additional mutation that operates on feature values, or by combining the feature values of points located in the same region of space. In the latter case, the cluster of points (AGs) would be represented by their centroid (AB) in a certain subspace.

The weighted feature selection is the next step in AISLFS development. In this case, the feature selection is not binary but weighted: vector $\mathbf{v}_k$ (1) has real components, which express the levels of importance of the features (the strength of individual feature binding).

It is planned to apply the AISLFS to regression problems, especially to short-term electrical load forecasting. In this multiinput problem, a regression function can be modeled not globally but locally using different sets of input variables (mainly historic loads, but also weather factors) for each day type, seasons of the year and hour of the forecasted load.

Another idea is to apply AISLFS to unsupervised learning, where data clusters can be formed based on locally selected features. In this case one point can belong to many different clusters, represented by ABs, taking into account different feature subsets encoded in AB receptors.

Acknowledgment

The author would like to thank Prof. R. Duquesnoy from the University of Pittsburgh Medical Center, Pittsburgh, PA, M. Sankowska and L. Kauc from Medigen Molecular Diagnostics, Warsaw, Poland, and Prof. A. Jóźwik from the Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Warsaw, for their inspiration and creative cooperation.

References

[1] E. Hart and J. Timmis, "Application areas of AIS: The past, the present and the future," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 191–201, Jan. 2008.

[2] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proc. IEEE Symp. Res. Security Privacy*, May 1994, pp. 202–212.

[3] S. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evol. Comput.*, vol. 7, no. 1, pp. 1289–1296, 2000.

[4] D. Dasgupta and S. Forrest, "Tool breakage detection in milling operations using a negative selection algorithm," Dept. Comput. Sci., Univ. New Mexico, Albuquerque, NM, Tech. Rep. CS95-5, 1995.

[5] Y. Ishida, "An immune network model and its applications to process diagnosis," *Syst. Comput. Japan*, vol. 24, no. 6, pp. 646–651, 1993.

[6] A. Ishiguru, Y. Watanabe, and Y. Uchikawa, "Fault diagnosis of plant systems using immune networks," in *Proc. IEEE Int. Conf. Multisensor Fus. Integr. Intell. Syst.*, Oct. 1994, pp. 34–42.

[7] K. P. Anchor, J. B. Zydallis, G. H. Gunsch, and G. B. Lamont, "Extending the computer defense immune system: Network intrusion detection with a multiobjective evolutionary programming approach," in *Proc. 1st Int. Conf. Artif. Immune Syst.*, 2002, pp. 12–21.

[8] R. Deaton, M. Garzon, J. A. Rose, R. C. Murphy, S. E. Stevens, Jr., and D. R. Franceschetti, "DNA based artificial immune system for self-nonself discrimination," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 1997, pp. 862–866.

[9] K. Mori, M. Tsukiyama, and T. Fukada, "Immune algorithm with searching diversity and its application to resource allocation problem," *Trans. Instit. Electr. Engineers Japan*, vol. 113-C, no. 10, pp. 872–878, 1993.

[10] T. Fukuda, K. Mori, and M. Tsukiyama, "Parallel search for multimodal function optimization with diversity and learning of immune algorithm," in *Artificial Immune Systems and Their Applications*, D. Dasgupta, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 210–220.

[11] L. N. de Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proc. Workshop Artif. Immune Syst. Their Applicat.*, 2000, pp. 36–37.

[12] A. Watkins and L. Boggess, "A new classifier based on resource limited artificial immune systems," in *Proc. Congr. Evol. Comput.*, May 2002, pp. 1546–1551.

[13] T. Knight and J. Timmis, "AINE: An immunological approach to data mining," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2001, pp. 297–304.

[14] S. Garrett, "How do we evaluate artificial immune systems?" *Evol. Comput.*, vol. 13, no. 2, pp. 145–178, 2005.

[15] S. T. Wierzchoñ, "Function optimization by the immune metaphor," *Task Quart.*, vol. 6, no. 3, pp. 1–16, 2002.

[16] V. Cutello, G. Nicosia, M. Pavone, and G. Stracquadanio, "An information theoretic approach for clonal selection algorithms," in *Proc. 9th Int. Conf. Artif. Immune Syst.*, 2010, pp. 144–157.

[17] L. N. de Castro and F. J. Von Zuben, "An immunological approach to initialize centers of radial basis function neural networks," in *Proc. 5th Brazilian Conf. Neural Netw.*, 2001, pp. 79–84.

[18] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239–251, Jun. 2002.

[19] G. Nicosia, V. Cutello, and M. Pavone, "A hybrid immune algorithm with information gain for the graph coloring problem," in *Proc. Genet. Evol. Comput. Conf.*, LNCS 2723. 2003, pp. 171–182.

[20] E. Hart, P. Ross, and J. Nelson, "Producing robust schedules via an artificial immune system," in *Proc. IEEE World Congr. Comput. Intell. Int. Conf. Evol. Comput.*, May 1998, pp. 464–469.

[21] J. Greensmith and S. Cayzer, "An artificial immune system approach to semantic document classification," in *Proc. 2nd ICARIS*, 2003, pp. 136–146.

[22] G. Dudek, "Artificial immune system for short-term electric load forecasting," in *Proc. 9th Int. Conf. Artif. Intell. Soft Comput.*, LNAI 5097. 2008, pp. 1007–1017.

[23] G. Dudek, "Artificial immune clustering algorithm to forecasting seasonal time series," in *Proc. 3rd Int. Conf. Comput. Collective Intell. Technol. Applicat.*, LNCS 6922. 2011, pp. 468–477.

[24] N. K. Jerne, "Toward a network theory of the immune system," *Ann Immunol (Inst Pasteur)*, vol. 125C, nos. 1–2, pp. 373–389, Jan. 1974.

[25] J. Farmer, N. Packard, and A. Perelson, "The immune system, adaptation and machine learning," *Physica D*, vol. 22, nos. 1–3, pp. 187–204, Oct.–Nov. 1986.

[26] T. Fukuda, K. Mori, and M. Tsukiyama, "Immune networks using genetic algorithm for adaptive production scheduling," in *Proc. 15th IFAC World Congr.*, vol. 3. 1993, pp. 57–60.

[27] D. Cooke and J. Hunt, "Recognizing promoter sequences using an artificial immune system," in *Proc. 3rd Int. Conf. Intell. Syst. Mol. Biol.*, 1995, pp. 89–97.

[28] J. Timmis, M. Neal, and J. Hunt, "An artificial immune system for data analysis," *Biosystems*, vol. 55, nos. 1–3, pp. 143–150, 2000.

[29] L. N. de Castro and F. J. Von Zuben, "aiNet: An artificial immune network for data analysis," in *Data Mining: A Heuristic Approach*, H. A. Abbass, R. A. Sarker, and C. S. Newton, Eds. Hershey, PA: Idea Group Publishing, 2001, ch. XII, pp. 231–259.

[30] S. T. Wierzchoń and U. Kużelewska, "Stable clusters formation in an artificial immune system," in *Proc. 1st ICARIS*, 2002, pp. 68–75.

[31] J. E. Hunt and A. Fellows, "Introducing an immune response into a CBR system for data mining," in *Proc. BCS ESG: Res. Dev. Expert Syst. XIII*, 1996, pp. 35–42.

[32] R. E. Langman and M. Cohn, "The complete idiotype network is an absurd immune system," *Immunol. Today*, vol. 7, no. 4, pp. 100–101, Apr. 1986.

[33] P. Matzinger, "The danger model: A renewed sense of self," *Science*, vol. 296, no. 5566, pp. 301–305, 2002.

[34] J. Greensmith and U. Aickelin, "The deterministic dendritic cell algorithm," in *Proc. 7th ICARIS*, LNCS 5132. 2008, pp. 291–302.

[35] D. K. Male, J. Brostoff, I. M. Roitt, and D. B. Roth, *Immunology*. Maryland Heights, MO: Mosby Elsevier, 2006.

[36] R. J. Duquesnoy, "A structurally based approach to determine HLA compatibility at the humoral immune level," *Hum. Immunol.*, vol. 67, no. 11, pp. 847–862, Nov. 2006.

[37] R. J. Duquesnoy and M. Askar, "HLAMatchmaker: A molecularly based algorithm for histocompatibility determination: V. Eplet matching for HLA-DR, HLA-DQ and HLA-DP," *Hum. Immunol.*, vol. 68, no. 1, pp. 12–25, Jan. 2007.

[38] J. H. Carter, "The immune system as a model for classification and pattern recognition," *J. Am. Med. Inform. Assoc.*, vol. 7, no. 1, pp. 28–41, 2000.

[39] D. Goodman, L. Boggess, and A. Watkins, "Artificial immune system classification of multiple-class problems," in *Proc. Intell. Eng. Syst. Through Artif. Neural Netw. Smart Eng. Syst. Des. Neural Netw.*, 2002, pp. 179–184.

[40] J. Brownlee, "Immunos-81: The misunderstood artificial immune system," Center Intell. Syst. Complex Processes, Faculty Inform. Commun. Technol., Swinburne Univ. Technol., Melbourne, Australia, Tech. Rep. 3-01, 2005.

[41] A. Watkins and L. Boggess, "A new classifier based on resource limited artificial immune systems," in *Proc. Congr. Evol. Comput.: Part World Congr. Computat. Intell.*, 2002, pp. 1546–1551.

[42] A. Watkins and J. Timmis, "Artificial immune recognition system (AIRS): Revisions and refinements," in *Proc. 1st ICARIS*, Sep. 2002, pp. 173–181.

[43] A. Watkins and J. Timmis, "Exploiting parallelism inherent in AIRS, an artificial immune classifier," in *Proc. 3rd ICARIS*, 2004, pp. 427–438.

[44] U. Markowska-Kaczmar and B. Kardas, "Multiclass iteratively refined negative selection classifier," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 972–984, Mar. 2008.

[45] C. McEwan and E. Hart, "Representation in the (artificial) immune system," *J. Math. Model. Algorithms*, vol. 8, no. 2, pp. 125–149, 2009.

[46] A. S. Perelson and G. Weisbuch, "Immunology for physicists," *Rev. Mod. Phys.*, vol. 69, no. 4, pp. 1219–1268, 1997.

[47] J. Carneiro and J. Stewart, "Rethinking shape space: Evidence from simulated docking suggests that steric shape complementarity is not limiting for antibody-antigen recognition and idiotypic interactions," *J. Theor. Biol.*, vol. 169, no. 4, pp. 391–402, Aug. 1994.

[48] C. McEwan, E. Hart, and B. Paechter, "Boosting the immune system," in *Proc. ICARIS*, LNCS 5132. 2008, pp. 316–327.

[49] T. Stibor, J. Timmis, and C. Eckert, "On the use of hyperspheres in artificial immune systems as antibody recognition regions," in *Proc. ICARIS*, LNCS 4163. 2006, pp. 215–228.

[50] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[51] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Berlin, Germany: Springer, 2001.

[52] I. R. Cohen, U. Hershberg, and S. Solomon, "Antigen receptor degeneracy and immunological paradigms," *Molec. Immunol.*, vol. 40, nos. 14–15, pp. 993–996, Feb. 2004.

[53] P. Andrews and J. Timmis, "Alternative inspiration for artificial immune systems: Exploiting Cohen's cognitive immune model," in *Silico Immunology*, D. Flower and J. Timmis, Eds. Berlin, Germany: Springer, 2007, pp. 119–137.

[54] P. Andrews and J. Timmis, "A computational model of degeneracy in a lymph node," in *Proc. 5th ICARIS*, LNCS 4163. 2006, pp. 164–177.

[55] M. Mendao, J. Timmis, P. S. Andrews, and M. Davies, "The immune system in pieces: Computational lessons from degeneracy in the immune system," in *Proc. IEEE Symp. Found. Comput. Intell.*, Apr. 2007, pp. 394–400.

[56] G. Dudek, "Tournament searching method to feature selection problem," in *Proc. 10th ICAISC*, LNAI 6114. 2010, pp. 437–444.

[57] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London, U.K.: Prentice-Hall, 1982.

[58] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[59] A. Asuncion and D. J. Newman. (2007). "UCI machine learning repository," School Inform. Comput. Sci., Univ. California, Irvine, CA [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[60] J. Brownlee, "Artificial immune recognition system (AIRS): A review and analysis," Center Intell. Syst. Complex Processes (CISCP), Faculty Inform. Commun. Technol., Swinburne Univ. Technol., Melbourne, Australia, Tech. Rep. 1-02, 2005.

[61] C. McEwan and E. Hart, "On AIRS and clonal selection for machine learning," in *Proc. 8th ICARIS*, 2009, pp. 67–79.

Grzegorz Dudek received the Ph.D. degree in electrical engineering from the Czestochowa University of Technology, Czestochowa, Poland, in 2003.

Currently, he is an Assistant Professor (Adjunct) with the Department of Electrical Engineering, Czestochowa University of Technology. He is the author and coauthor of over 50 scientific papers. Since 1999, he has been the Head Contractor of four grants financed by the Polish State Committee for Scientific Research and the Polish Ministry of Science and Higher Education. His current research interests include pattern recognition, machine learning, artificial intelligence, and their applications in classification, regression, and optimization problems.