# An Overlap-aware Positive Selection Algorithm using Variable-size Detectors

Azra Aryania[1], Ahmad Akbari[1], Mahdi Mohammadi[2], Bijan Raahemi[2], Elnaz Bigdeli[2]
[1]Computer Engineering Department
Iran University of Science and Technology
University Road, Hengam Street
Resalat Square
Tehran 16846-13114, Iran
[2]Knowledge Discovery and Data mining Lab
University of Ottawa
55 Laurier Ave, E., Ottawa
ON Canada K1N 6N5
ariania@comp.iust.ac.ir, {mmohamm6, braahemi, ebigd008}@uottawa.ca, akbari@iust.ac.ir

***ABSTRACT:*** *Classification of the test samples using positive selection is computationally expensive, as it requires comparisons with large number of detectors. In this paper, we propose an enhanced positive selection algorithm with variable-size detectors to reduce the number of detectors required to cover the training sample space while resolving the issues of ambiguity of the test samples (test samples covered by more than one detector from different classes) and exclusion (test samples not covered by any detectors). We apply clustering in the preprocessing phase to reduce the number of detectors. We then perform overlap checking to adjust the radiuses of the variable-size detectors aiming at reducing overlap while covering the sample space. Furthermore, a weighted voting scheme is employed to resolve the ambiguity, and a distance-based method is devised to resolve the issue of exclusion. We evaluate the performance of our proposed algorithm on five benchmark datasets. The experimental results confirm the superiority of the proposed method in terms of number of detectors and accuracy rate.*

**Keywords:** Artificial Immune System, Positive Selection algorithm, Variable-size Detectors, Classification

**Received:** 1 February 2014, Revised 9 March 2014, Accepted 14 March 2014

## 1. Introduction

Artificial Immune Systems are soft computing techniques based on metaphor of the biological immune system. The natural immune system exhibits many interesting characteristics like learning, pattern matching, feature extraction (Mohammadi et al.2012) and distributed processing. The AIS algorithms mostly imitate one of the following mechanisms of the immune system: negative selection, positive selection and immune network. (Dasgupta, Ji and Gonzalez 2003; Castro and Timmis 2003).

In an artificial immune system (AIS), principles and processes of the natural immune system are abstracted and applied to solve information-processing problems, such as anomaly detection. An early (and popular) immune-inspired algorithm for anomaly detection is negative selection. Forrest (Forrest et al. 1994) developed this algorithm. Forrest described a general method for distinguishing self from other in the context of computational systems, and he has illustrated its feasibility as a change-detection method on the problem of computer virus detection. The main idea of negative selection algorithm is generating detectors in the data space, and then classifying the unseen samples as *self* or *non-self*.

Ebner et al (Ebner et al. 2002) present the main idea of generating detectors for continuous data. They considered the detectors and antigens as *n*-dimensional vectors. Followed by Ebner, Gonzalez et al (Gonzalez and Dasgupta 2003) used real-value representation (instead of binary values) to specify *self* and *non-self* spaces named as real-valued negative selection (RVNS). The input of RVNS algorithm is *n*-dimensional vectors of *self*-samples. More specifically, *T* lymphocytes are abstracted by means of hyper spheres. In the training phase, the hyper spheres (also called detectors) are (randomly) distributed in an unitary hypercube $H = [0, 1]_n$ of dimension *n*, such that each self-element — also represented as a hyper sphere is not covered by any detector.

In the testing phase, an element $p \in H$ is classified as a self-element, if *any detector does not cover p*. This type of hyper sphere detection is known as instance-based learning.

Gonzalez et al in (Gonzalez 2003) described a real-valued representation for the negative selection algorithm and its applications to anomaly detection. Gonzalez in (Gonzalez, Dasgupta and Nino 2003) proposed randomized real-value negative selection algorithm based on Monte Carlo methods. This algorithm is based on solid mathematical foundation that solves some of the drawbacks of the RVNS algorithm. Specifically, it can produce a good estimate of the optimal number of detectors needed to cover the *non-self* space, and the maximization of the non-self coverage is done through an optimization algorithm with proved convergence properties.

Ji and Dasgupta (Ji and Dasgupta 2004) proposed an extension of real-valued negative selection algorithm with a variable coverage detector generation scheme. They called it V-Detector. It can cover more sample space than what the RVNS does with constant detectors. Moreover, experimental results demonstrated that V-Detector scheme is more effective in using smaller number of detectors because of their variable sizes. Wu and Zheng (Wu and Zheng 2012) proposed an improved Variable-Radius Real-valued Negative Selection Algorithm that includes two tolerance processes to generate mature detectors. In first tolerance process, each candidate detector tolerates with mature-detector set and is considered as semi-mature detector when it does not match any existing mature detector. In second process, each semi-mature detector tolerates with self-set and accepted as mature detector when it does not match any self sample. Theoretical analysis and simulation shown that proposed algorithm has better detector set generation efficiency and quality in comparison with RNSA and V-Detector.

RNSA and V-Detector use negative selection algorithm to conduct self-tolerance of training set in n-dimensions real value space that leads to high false alarm rate of classification algorithm. Zheng and et all (Zheng, Zhou and Fang 2013) proposed an algorithm named (PRR-2NSA) to solve the problems of RNSA and V-Detector. The proposed algorithm employs Antigen Presenting Cells (APC) classifiers co-stimulation for T-Cell classifier to reduce false classification rate and time cost. The experimental results show that the PRR-2NSA is better than V-detector and NSA in terms of efficiency and false alarm rate.

Ataser (Ataser 2013) proposed a new version of negative selection algorithm called V-shaped detector to obtain maximum nonself coverage. The proposed algorithm focuses on two issues, self-space determination and non-self coverage. This algorithm employs local outlier factor (LOF) and *k*-nearest neighbor (KNN) to determine self-boundary. In addition, the new algorithm, allows detectors to include self-samples and the shapes of detectors with self-samples are changes using cubic spline. The experimental results reveal that the accuracy of V-shaped detector is better than V-detector. However, the number of generated detectors to cover non-self space in V-shaped detector is more than the number of detectors in V-detector.

Instead of covering the hypercube *H* with detectors completely, Ebner et al. (Ebner et al. 2002), and subsequently, Stibor et al. (Stibor, Mohr and Timmis 2005) discussed a different hyper sphere detection form where there exists no *T* Lymphocyte detector. Instead, only the given self-elements are abstracted by hyper spheres. Therefore, no training phase is required. In the testing phase**,** *p* is classified as a self-element if *p* is covered by a hyper sphere, and otherwise, as an anomalous element. This type of hyper sphere detection is a simple instance-based learning method, and is called real-valued positive selection (RVPS).

Positive selection has been successfully applied to solve various problems from anomaly detection to NP complete optimization. Positive selection is also applied to multi-class classification tasks. Its detection rate and false alarm rate are superior to that of negative selection algorithm (Stibor, Timmis and Eckert 2005; Stibor and Timmis 2007). However, the number of generated detectors (in the training phase) is high and it can be a challenging disadvantage of RVPS. A positive selection classification algorithm (PSCA) based on positive selection algorithm is proposed in (Zhang and QI 2012). PSCA turns the multi-class classification problem into a two-class classification problem: self and non-self, therefore classifiers are generated only for selfclass samples. Also, K-nearest neighbor algorithm is used to solve hole problem (is called exclusion problem in this paper). The experimental results demonstrated that PSCA outperforms AIRS, ANSC and some classification algorithm.

In this paper, we introduce an enhanced version of positive selection to reduce the number of detectors while maintaining its high detection rate. For this, after applying clustering in the preprocessing phase, we check for overlaps in the training phase and adjust the radius of the variable-size detectors to minimize overlaps among them. In the test phase, we propose techniques to resolve the issues of ambiguity (where more than one detector from different classes covers test samples) and exclusion (where any detectors do not cover test samples).

The rest of the paper is organized as follows. In Section 2, we briefly describe the real-value positive selection algorithm. We then describe our proposed algorithm in details in Section 3. The experimental results on the IRIS, Wine, Segment, P2P and a synthetic 2D dataset are presented in Section 4. The comparison between our proposed method and other classification algorithms in terms of accuracy, and number of detectors are explained in this section, as well. Finally, we conclude the paper in the final section.

## 2. Real - value Positive Selection

Real-valued positive selection (RVPS) algorithm was informally described by Ebner et al. (Ebner et al. 2002) and formally by Stibor et al (Stibor, Mohr and Timmis 2005). In positive selection algorithm, each self-sample is covered by a detector. The self detector classification operates on a unitary hypercube $[0,1]^n$. The Self-elements are considered as *self-detectors*. A self detector $d = (c, r_s)$ has a center $c \in [0,1]^n$ and a self-radius $r_s$. An element $e$ lies within a detector $d_s = (c, r_s)$, if the Euclidean distance *dist* $(c, e) = ( \Sigma_{i=1}^{n} (c_i - e_i)^2 )^{1/2} < r_s$ where $r_s$ is determined in training phase by means of the ROC analysis. An element is classified as self if it lies within a *self-detector*. Otherwise, it is *non-self*. Geometrical interpretation of the definition is represented in Figure 1.

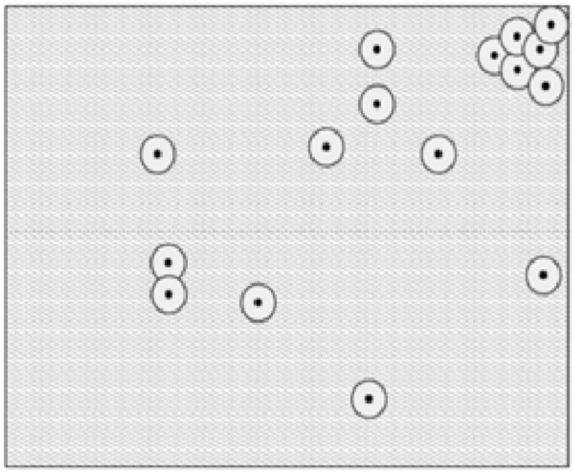
Figure 1. Self - detectors in a unitary hypercube

The positive selection algorithm includes 3 steps:

1. For training examples $s \in s$, generate *self-detectors* $d_s$ with $r_s = 0$.

2. Perform $i$ training classification runs and find the radius $r_s$, which yields the minimum error.

3. Classify new examples as *self* if Euclidean dist $(d_s, x) < r_s$. Otherwise, consider it as *non-self*.

In this algorithm, no detector generation phase is necessary. The entire training set is considered as the detector set, i.e. the number of detectors is equal to the number of samples in the training set. However, the classification of the new sample is computationally expensive, compared to the randomized real-valued negative selection. Stibor showed that the real-valued positive selection outperformed the other classification methods on a large data set. However, real-valued positive selection suffers from high computational complexity (Stibor, Timmis and Eckert 2005). To solve this problem, Stibor et al. applied kmeans clustering algorithms to reduce the number of detectors in the RVPS (Stibor and Timmis 2007).

Since positive selection, unlike other immune system algorithms, can be used for multi-class classification, and its detection and false alarm rates are superior to that of negative selection algorithm, in this research, we focus on positive selection, and introduce some enhancements to overcome its shortcomings, and improve its performance. We introduce methods in the training and test phases to improve the detection rate of the proposed algorithm. With the proposed approaches, we achieve high detection rate and low incorrect rate with less number of detectors compared to the original RVPS. The proposed algorithm is explained in more details in Section 3.

## 3. The Proposed Enhanced Positive Selection Algorithm

In this section, we introduce an improved version of positive selection algorithm, called Enhanced Positive Selection (EPS),which is illustrated in Figure 2. As shown in Figure 2, at first, the data set is normalized, then divided into two sets, training and test sets. The training instances are clustered; the number of clusters is an input parameter for the proposed method. Although the number of clusters is obtained by trial, its value is dependent on the size of dataset and the number of classes. In this way, we assume double or triple of number of classes as number of clusters for small data sets. The center of each cluster is considered as a detector. Since detectors might overlap, we apply overlap resolving to remove overlap between detectors. Then, new detectors are generated to include uncovered training instances. In this step, we apply overlap checking to tune the radius of the

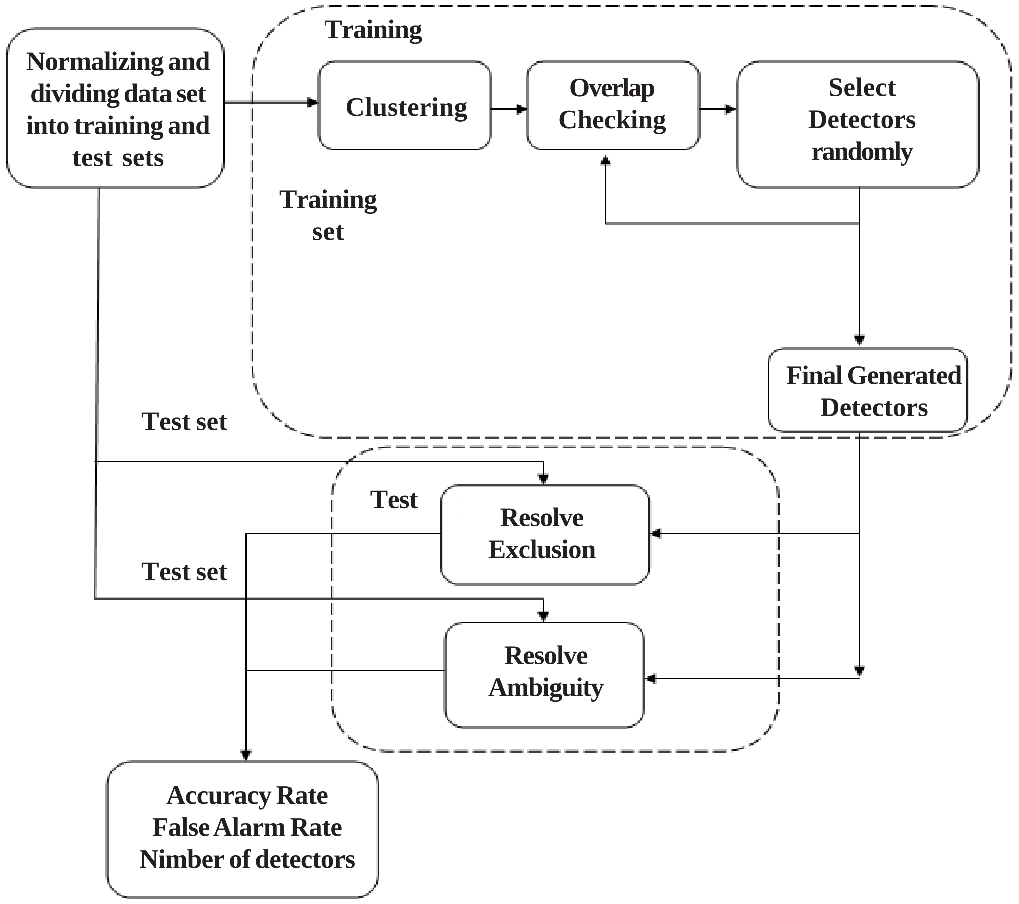

Figure 2. Block diagram of the proposed Enhanced Positive Selection algorithm

detectors.

In the test step, we resolve the problems of Exclusion (test samples not covered by any detectors) and *Ambiguity* (ambiguity in test samples assignments to detectors) applying novel approaches that are explained in the following sub-sections in details.

**3.1 Normalizing Data Set**

Normalization scales numeric variables in the range [0, 1]. After normalization, the data set is divided into training and test sets with the portion of 9 to 1, i.e. we use 10-fold cross validation for evaluating the proposed method.

**3.2 Clustering the Training Data Set**

Whereas the distribution of the data set is not specific, the training data set is divided into *k* clusters by applying K-means algorithm on the training samples. Clustering algorithm is applied into training samples to reduce the number of generated detectors. Each detector is represented based on the four components: the covered samples by the detector, detector's center, detector's label and detector's radius. Each detector is then represented by a cluster covering all samples in own self. This is to say, instead of considering each sample as a *self detector* (similar to what Positive Selection does) we consider each cluster as a *self detector* which covers the entire samples within that cluster, considering the center of the detector the same as the center of the cluster. We then label each detector based on the label of the sample that represents the detector's center. If the cluster center is not a training sample, the label of the most samples of cluster with the same label is considered as the detector's label. In the other words, the majority voting approach on the label of detector's samples is used to determine the detector's label. The detector's radius is the distance between the cluster's center (detector's center) and the farthest sample in that cluster.

**3.3 Overlap Checking**

Having established detectors in the previous step, we check the overlap between the detectors to reduce the number of clusters. Needless to say, overlap is sometimes necessary to cover all instances of the training set (Figure 3 (Gonzalez 2003)) based on which we just reduce the overlap.

Therefore, detectors with the same class label might overlap to cover the entire space, but the amount of overlap should not be more than a given *threshold value*. The threshold value is primarily dependent on the detectors' radiuses. The overlap threshold for small detector is smaller than the overlap threshold for large detector. In addition, the threshold should be calculated such that the entire space is covered by a minimum number of detectors. Based on the mentioned criteria we have proposed Equation (1) based on which the value of threshold for each cluster is calculated

$$\sigma_{overlap} = (R_i + R_j) - \alpha (R_i + R_j) \tag{1}$$

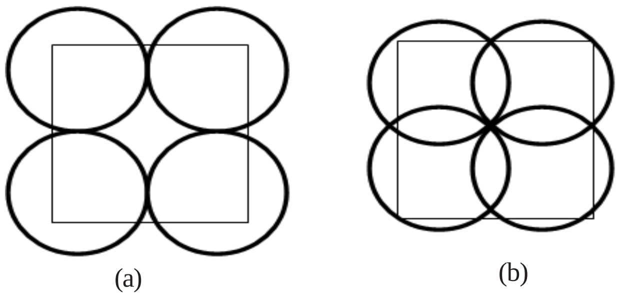

Figure 3. Complete coverage of rectangular area by circular detectors, (a) without overlap; and (b) with overlap

where $R_i$ and $R_j$ are the radiuses of the detectors, and $\alpha$ is a constant parameter used to adjust the amount of overlap between detectors *i* and *j*.

Having calculated the overlap threshold based on Equation (1), we resolve overlap between the detectors. The decision for resolving overlap is made locally with the goal of increasing accuracy so then the proposed method works in a greedy approach. As greedy methods are usually fast enough to solve optimization problems, we devise a greedy method by which we reduce amount of overlap between the local detectors belonging to the same class, with the hope that the number of detectors will lead to a desired number at the end of run. As we generate detectors one-by-one, the new generated detector is compared to the two steps to resolve overlap for the two cases:

previous ones. Detectors, which overlap the new generated detector, are considered as its neighbors. The radius of the new generated detector is reduced based on the amount of overlap among its neighbors in Equation (1). More detailed explanation about the reduction process is proposed in session 3.3.1 and 3.3.2.

Overlap can be among either detectors with the same label or detectors with different labels. Therefore, we propose the following two steps to resolve overlap for the two cases:

**a) Overlap among detectors with the same labels:** If two detectors with the same label overlap and the overlap value is larger than the threshold, then the detector's radius is reduced to the point to which the overlap value is smaller than the threshold. This approach is explained in sub-section 3.3.1 in details.

**b) Overlap among detectors with different labels:** If detectors with different labels overlap, then the overlap is totally removed. The detector's radius is reduced to the point that the boundaries of the detectors are approximately tangent. This approach is explained in sub-section 3.3.2 in details.

### 3.3.1 Resolving Overlap among Detectors with the Same Class Labels

If the distance between centers of detectors (with the same class label) is smaller than the overlap *threshold*, the radius of the variable-size detector should be reduced (variable size detector is the one that was added to the pool of detectors recently). That is to say, if $d_{i,j} < \sigma_{overlap}$ where $d_{i,j}$ the distance between the centers of is overlapped detectors, and $\sigma_{overlap}$ is the overlap threshold, then the radius of the variable-size detector $R_j$ is adjusted according to Equation (2):

$$R_j = (d_{i,j} + R_i * (\alpha - 1)) / (1 - \alpha) \tag{2}$$

where, $\alpha$ is the overlap constant. In the next section is explained resolving overlap among detectors with different labels in details.

### 3.3.2 Resolving Overlap among Detectors with Different Class Labels

In the case of overlap among detectors with different class labels, to achieve higher accuracy rate, the radius of the variable-size detector should be reduced to almost zero (reduced to $\varepsilon > 0$). Suppose detector $D_i$ covers instances $\{s_1, s_2, ..., s_n\}$ of the other class. The following steps are taken to remove the overlap between the detectors with different labels.

**Step 1.** Calculate distance $\{d_{1,i}, d_{2,i}, ..., d_{n,i}\}$ between the center of detector $D_i$ and the instances $\{s_1, s_2, ..., s_n\}$ of a different class covered by detector $D_i$

**Step 2.** Find the minimum of $\{d_{1,i}, d_{2,i}, ..., d_{n,i}\}$, and reduce the radius of the variable-size detector according to Equation (3):

$$R_j = min\,(d_{1,i}, d_{2,i}, ..., d_{n,i}) + \varepsilon \tag{3}$$

where, $\varepsilon$ is the acceptable error margin for the overlap between the detectors with different class labels.

### 3.4 Assigning Detectors to Uncovered Samples

Having resolved overlap among detectors, we might be left with samples which do not belong to any detectors. This is because reducing the radius of detectors may leave out some samples. This is conceptually shown in Figure 4 where resolving overlaps may leave samples not covered by any detectors.

We propose the following five steps to generate new detectors. This algorithm is repeated until there are no uncovered samples left:

**Step 1.** Randomly select one sample from the uncovered training instances.

**Step 2**. Consider the sample as the center of a new detector with the initial radius *r*. Note that we initiate the radius of detectors to an initial value that can be reduced during resolve overlap phase. The initial radius can be valued among 0.9, 0.8, 0.7, 06, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05 or 0.01 at the first of each run.

**Step 3:** All the uncovered samples which are surrounded in the new generated detector, will be removed from uncovered sample set.

**Step 4**: Assign a label to the new detector based on the label of center.

**Step 5**: Resolve overlap between new detector and its neighbor detectors, as explained in sub-section 3.3.1 and sub-section 3.3.2.

## 3.5 Test Phase

The detectors generated in the training phase will now be used to predict the label of instances in the test set. In this phase, three cases could happen:

1. One or more detectors with the same class label cover the test sample. In this case, the label of the sample is the same as the label of detectors.

2. Two or more detectors with different labels cover the test sample. This is the case of ambiguity, which is resolved by the approach presented in sub-section 3.5.1.

3. Any detectors do not cover the test sample. This case of exclusion (or holes) is resolved by the approach proposed in sub-section 3.5.2.

### 3.5.1 Resolving Ambiguity

As shown in Figure 5, a test sample may be covered by more than one detector with different labels. In this case, there is an ambiguity about the label should be assigned to the sample. Here, we present an approach to assign a proper label to the *ambiguous sample*.

The main idea of this approach is that the closer a detector is to an *ambiguous* sample, the more influence it has to determine the label of the sample. Therefore, the distance between the *ambiguous* sample and the detectors is considered as the main determinant. For this, we proposed a membership score based on which the *ambiguous* sample can be classified. For this to happen, we calculate the membership score for the ambiguous sample and each detector, which covers the sample. When the distance between the test sample and the detector is small, the membership score should be high; Moreover, the detectors with larger radius are more general and less accurate than the detectors with smaller radius, which are more specific. To consider the inverse relationship between the membership score and the size of the detector, as well as the distance between the sample and detector, we calculate the membership score for the *ambiguous* sample belonging to the detector $D_i$ as:

$$MS\,(x \mid D_i) = \frac{1}{d_i * R_i} \tag{4}$$

Where $x$ is the test sample, $D_i$ is ith detector, $d_i$ is distance between the *ambiguous* sample and the detector $D_i$ , and $R_i$ is radius of $i^{th}$ detector. This score is calculated for each detector that covers the *ambiguous* sample. Then, the membership score for the ambiguous sample belonging to the class with label $\omega_j$ is the maximum of the scores for all detectors with class label $\omega_j$ as expressed in Equation (5):

$$MS\,(x \mid \omega_j) = \max_i MS\,(x \mid D_i) = \max_i \frac{1}{d_i * R_i} \quad i = 1....n \tag{5}$$

where, $\omega_j$ is the class label of the detector $D_i$ and $n$ is the number of detectors with class label $\omega_j$. The score are calculated for each different class 1…$m$. Then, the score that the *ambiguous* sample belongs to a class $\omega$ $MS\,(x \mid \omega)$ is the maximum scores for all detectors covering that sample:

$$\begin{aligned} MS\,(x \mid \omega_x) &= \max\,[MS\,(x \mid \omega_i), MS\,(x \mid \omega_j), ..., MS\,(x \mid \omega_p)] \\ &= Max\,[\max_i MS\,(x \mid D_i), \max_j MS\,(x \mid D_j), ..., \max_p MS\,(x \mid D_p)] \\ & i = 1....n_1 \; j = 1...n_2 \; p = 1...n_m \end{aligned} \tag{6}$$

where, $n_1$, $n_2$ and $n_m$ are the number of detectors with class label $\omega_i$ , $\omega_j$ and $\omega_p$, respectively.

### 3.5.2 Resolving Exclusion

If any detector does not cover a test sample, we apply a weighted K-Nearest Neighbor (KNN) algorithm to determine the label of the sample. In KNN, in its basic form, K-nearest neighbors of the test sample are specified. Then, the label for the sample is determined by simple majority voting. However, in cases such as the one illustrated in Figure 6, when there is a tie in voting, it will be difficult to conclude on the class label.

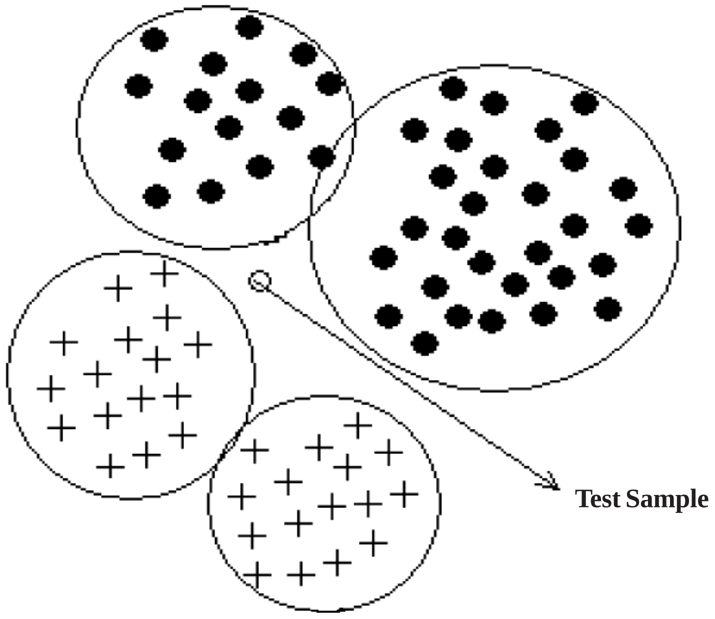

Figure 6. When there is a tie in simple voting, it is difficult to determine the class label of the test sample using KNN

To address this issue, detectors' votes are weighted based on the distance between the test sample and the center of the detectors. The closer the detector is to the test sample, the higher its voting weight is. Specifically, the following steps are proposed to calculate the proper weights for the KNN algorithm:

1. Calculate the distance between the uncovered sample and its $k$ neighboring detectors.

$$d\,(x, D_i),\ \ i = 1...k$$

2. Normalize the distances using:

$$d_{norm}\,(x, D_i) = \frac{d\,(x, D_i)}{D}\ \ i = 1...k \qquad (7)$$

where, $d\,(x, D_i)$ is the distance between the uncovered sample x and the neighbor detector $D_i$ and D is the distance between the test sample and its farthest neighbor that is $D = Max\,[d\,(x, D_i),\ i = 1\ldots k]$.

3. Sort and rank the neighbor instances based on their distances from the test sample such that the rank of the closest neighbor is $k$, and that of the farthest is 1.

*if $D_i$ is Closest neighbor detector then rank $_{Di}$ is minimum and rank $_{Di} = k$*

*if $D_i$ is farthest neighbor detector then rank $_{Di}$ is minimum and rank $_{Di} = 1$*

4. Weight neighboring instances using Equation 8.

$$W_{Di} = rank_{\,Di} / k \qquad (8)$$

where, *rank* $_{Di}$ and $k$ are neighbor sample rank and the number of neighbor instances, respectively. This relationship gives higher weight to closer detectors. Equation (9) calculates the membership score that the test sample belongs to detector $D_i$.

$$MS\,(x \mid D_i) = d_{norm}\,(x, D_i) * W_{Di} \tag{9}$$

The score is calculated for K nearest detectors to the uncovered sample. Then, the score that the uncovered sample belongs to class with label $\omega_j$,will be the maximum of the membership scores for all detectors with label $\omega_j$. Equation (10) states this expression:

$$MS\,(x \mid \omega_j) = \max_i MS\,(x|D_i) = \max_i d_{norm}(x, D_i) * W_{Di} \quad i = 1...n \tag{10}$$

where, $\omega_j$ is the class label and $n$ is the number of detectors with class label $\omega_j$. The scores are calculated for all classes. Then, the score that a test sample belongs to a class is the maximum score as in Equation (11):

$$\begin{aligned} MS\,(x \mid \omega_x) &= max\,[MS\,(x \mid \omega_i), MS\,(x \mid \omega_j), ..., MS\,(x \mid \omega_p)] \\ &= Max\,[\max_i MS\,(x \mid D_i), \max_j MS\,(x \mid D_j), ..., \max_p MS\,(x \mid D_p)] \\ & i = 1.... n_1\ j = 1... n_2\ p = 1... n_m \end{aligned} \tag{11}$$

where, $n_1$, $n_2$ and $n_m$ are the number of detectors with class label $\omega_i$ , $\omega_j$ and $\omega_p$, respectively.

## 4. Experiments and Results

In this section, the proposed methods are evaluated on five datasets. In the following, we describe the data sets, evaluation parameters and the experimental results on each data set. The evaluated methods are described in Table 1.

| Method | Description |
|---|---|
| RVPS | Real value positive selection algorithm with constant radius where each training sample is considered as a detector. |
| Randomized Positive Selection –RPS | The version of real value positive selection, in which detectors are selected randomly from uncovered instances.Clustering, overlap checking, resolving exclusion and ambiguity are not included in this method. |
| EPSL | The light version of the proposed algorithm, which performs clustering, overlap checking and assignment of uncovered samples, but does not resolve Exclusion and Ambiguity. |
| EPS | The full version of the proposed algorithm (Enhanced Positive Selection) |

Table 1. The evaluated methods

| Data set | Total Number of Samples | Number of Training Samples | Number of Test Samples | Number of Features | Number of Classes |
|---|---|---|---|---|---|
| IRIS | 150 | 135 | 15 | 4 | 3 |
| 2D data set | 1800 | 1620 | 180 | 2 | 2 |
| Wine | 178 | 160 | 18 | 14 | 3 |
| Segment (Image Segmentation) | 2310 | 2079 | 231 | 19 | 7 |
| P2P | 32767 | 29490 | 3277 | 5 | 2 |

Table 2. Characteristics of benchmark data sets

The Proposed algorithm (EPS) is evaluated on five data sets including IRIS, Wine, image segmentation (Segment), P2P and 2D dataset as listed in Table 2. The proposed methods are evaluated based on 10-fold cross validation. Therefore, the number of training samples is equal to 0.9 * *total number of samples* for each validation and the remains are considered as test samples. One of the benchmark datasets is a synthetic data set, called 2D dataset, with the scatter plot as shown in Figure 7.

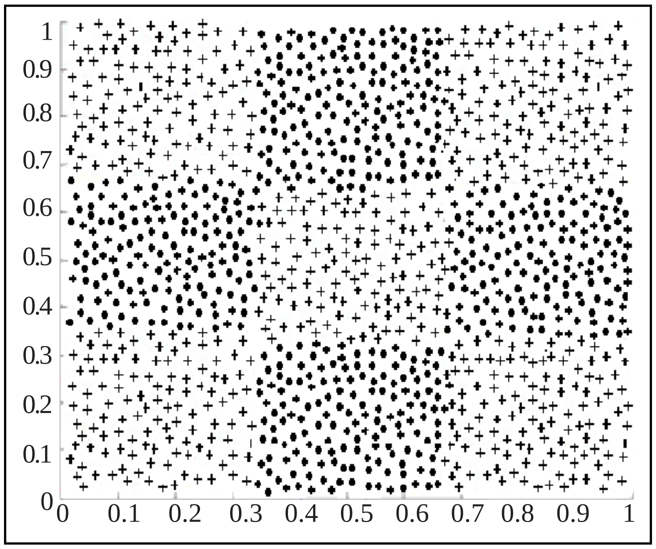

Figure 7. The distribution of the two dimensional synthetic dataset

As shown in Figure 7, 2D dataset consists of two classes. The samples are scattered throughout nine areas with the label of each area different from its neighbors. Therefore, none of the neighbor areas is from the same class. Iris, Wine and Segment datasets can be downloaded from UCI Machine Learning Repository. The details of P2P dataset is explained in (Mohammadi et al. 2011).

## 4.1 Evaluation Metrics

Performance of the proposed method is evaluated based on five metrics as follows:

- **N**umber of **D**etectors (*ND*)
- Rate of test instances classified correctly (*AR*)

$$AR\ (\textbf{A}\text{ccuracy }\textbf{R}\text{ate}) = \frac{\text{Number of correct classified instances}}{\text{Total number of instances in the test set}}$$

- Rate of test instances classified incorrectly (*FR*)

$$FR\ (\textbf{F}\text{alse }\textbf{R}\text{ate}) = \frac{\text{Number of incorrect classified instances}}{\text{Total number of instances in the test set}}$$

- Rate of test instances placed in different detectors (*AmR*)

$$AmR\ (\textbf{A}\text{mbiguity }\textbf{R}\text{ate}) = \frac{\text{Number of Ambiguious instances}}{\text{Total number of instances in the test set}}$$

- Rate of instances which are not placed in any detectors (*ExR*)

$$ExR\ (\textbf{E}\text{xclusion }\textbf{R}\text{ate}) = \frac{\text{Number of excluded instances}}{\text{Total number of instances in the test set}}$$

## 4.2 Experimental Results

In this section, we present experimental results on benchmark data sets, Iris, Wine, Segment P2P and 2D data set. The schema of experiment results on 2D data set is shown in Figure 8. As shown in this figure, without overlap checking there are many detectors with overlaps in the same classes or different ones. As shown in Figure 8(b), by applying overlap checking, the number of samples that are placed in different detectors is reduced significantly, which leads the final classifier to have lower incorrect detection (false alarm) rate. Hereafter, the proposed method is compared with the other methods, mentioned in Table 1, on different datasets separately.

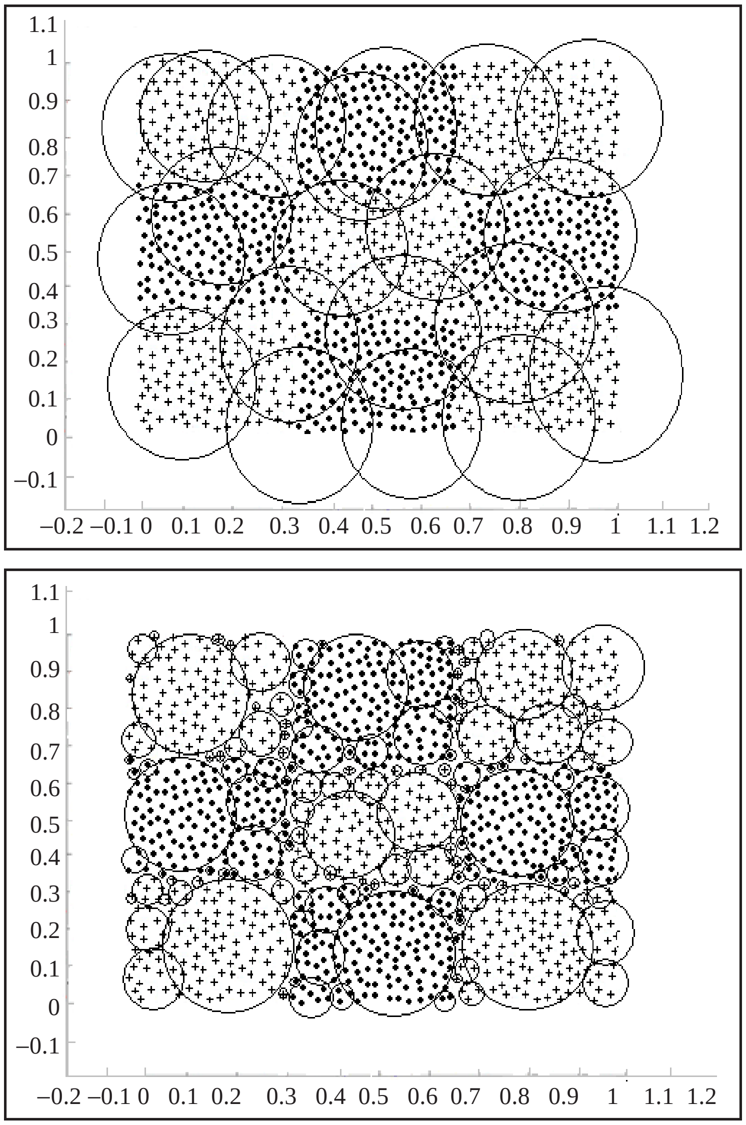

Figure 8. The results of RPS and EPS algorithms on 2D dataset

### 4.2.1 Analysis of the Results on the IRIS Data Set

The experimental results on the IRIS dataset are presented in Table 3. Each value in this Table is the average of10 different runs for each method. Each run is evaluated with the mean of 10-fold cross validation that makes the ND value is a float point number, instead of integer number.

In this table, in each column the best result among the different methods is highlighted in **bold and underline,** and the second best value is marked by underline. This notation will be used in the upcoming tables as well. As EPS and EPSL are the same algorithm (apart from dealing with ambiguous and uncovered samples), their results in the ND column (Number of Detectors) are the same. ND is an important measure, which has a direct impact on memory complexity. As we devised two methods to deal with ambiguity and uncovered samples, number of ambiguous and uncovered samples is zero when the EPS is applied; the false alarm classification rate for this method is about 4.5%. False alarm value for the EPSL is 2% with 4% ambiguous and 4% uncovered samples, which collectively are 10% of the whole samples of the test set. By applying the EPS method, we could label all the test samples while the false alarm rate is increased only around 2.5%.In term of accuracy rate (AR), the EPS outperforms other methods significantly. Not only the EPS is a method with the minimum number of detectors among the others, but also it performs the best in terms of accuracy rate.

| **Method** | **ND** | **AR** | **FR** | **AmR** | **ExR** |
|---|---|---|---|---|---|
| **RVPS** | 135 | 77.33 | **0.67** | 1.33 | 2.06 |
| **RPS** | 36.3 | 76.67 | 2.67 | 14 | 6.67 |
| **EPSL** | **31.7** | 90.00 | 2.00 | 4.00 | 4.00 |
| **EPS** | **31.7** | **95.33** | 4.66 | **0** | **0** |

Table 3. Experiment Results on the IRIS Dataset

### 4.2.2 Analysis of the Results on the Wine Data Set

The experimental results on the Wine dataset are presented in Table 4. The experimental results on the Wine dataset are similar to what we observed on the Iris dataset. Considering number of detectors, the EPS and EPSL show the best results in comparison with the other methods. In terms of accuracy rate, the EPS performs the best. This demonstrates that our proposed method, while employing fewer numbers of detectors, achieves the best accuracy rate among the others. The EPSL cannot label around 15% of the test samples (7.81% ambiguous samples and 6.69% uncovered samples). This value is around 22% for the RSP, and around 20% for the RVPS. Since the EPS can label ambiguous and uncovered samples, the false alarm value for this method is only 2.81%.

| **Method** | **ND** | **AR** | **FR** | **AmR** | **ExR** |
|---|---|---|---|---|---|
| **RVPS** | 160.2 | 78.18 | 1.14 | 10.13 | 10.53 |
| **RPS** | 58.9 | 76.94 | **0.55** | 3.95 | 18.54 |
| **EPSL** | **45.9** | 83.87 | 1.70 | 7.81 | 6.69 |
| **EPS** | **45.9** | **97.18** | 2.81 | **0** | **0** |

Table 4. Experiment Results on the Wine Dataset

### 4.2.3 Analysis of the Results on the Segment Data Set

The experimental results on the Segment dataset are presented in Table 5. The number of detectors for the EPS and EPSL is 149.8, followed by the RSP with 454.4 detectors (4 times higher than that of the EPS). The difference between the number of detectors for the EPS and RPS is significant. In terms of uncovered and ambiguous samples, the RVPS cannot label around 24% of the test dataset samples. These values are about 23% and 8% for the RSP and EPSL methods, respectively. Whereas, the EPS, while resolving the ambiguity and exclusion problems, exhibits only 4.5% false alarm value. This demonstrates that the EPS can label 95% of the test dataset samples correctly, while most of the ambiguous and uncovered samples are labeled correctly.

| **Method** | **ND** | **AR** | **FR** | **AmR** | **ExR** |
|---|---|---|---|---|---|
| **RVPS** | 2079 | 75.23 | **0.47** | 18.05 | 6.23 |
| **RPS** | 454.4 | 75.88 | 0.82 | 13.76 | 9.52 |
| **EPSL** | **149.8** | 89.08 | 1.29 | 5.49 | 2.12 |
| **EPS** | **149.8** | **95.45** | 4.54 | **0** | **0** |

Table 5. Results on the Segment Dataset

### 4.2.4 Analysis of the Results on the P2P Data Set

The experimental results on the P2P dataset are presented in Table 6. Similar to the previous experiments, the EPSL and EPS algorithms employ the lowest number of detectors among the others, while their accuracy rates are the highest. In addition, the ESP addresses the ambiguity and exclusion problems (i.e. zero ambiguous and uncovered samples). In overall, similar to results on the previous evaluated datasets, the EPS is placed first or second in terms of ND, AR, AmR and ExR criteria among the others.

| Method | ND | AR | FR | AmR | ExR |
|---|---|---|---|---|---|
| **RVPS** | 29491 | 91.72 | **0.03** | 4.85 | 3.38 |
| **RPS** | 2333 | 92.09 | 0.09 | 4.94 | 2.86 |
| **EPSL** | **811** | 95.23 | 0.45 | 1.12 | 0.18 |
| **EPS** | **811** | **98.83** | 1.16 | **0** | **0** |

Table 6. Results on the P2P Dataset

In the next experiments, we compare the proposed method with some well-known classification algorithms in terms of accuracy rate.

## 4.3 Comparison of the EPS and Other Classification Algorithms

In this section, the comparison of the EPS and other classification algorithms on the benchmark datasets is presented. Table 7 shows the accuracy rate of the various algorithms. The best result in each row is highlighted in **bold and underline**, and the second best result is marked as underline.

A variety of classifiers with different attributes is considered in this experiment. For instance, Knn is a classifier, which works based on measuring distances, and C4.5 is a decision tree. On the IRIS dataset, the EPS is the best method with 97.33% accuracy rate. In the case of Wine dataset, the best result belongs to Naïve Bayes classifier with 98.31% followed by the EPS with 97.22% detection rate. Similarly, for the P2P dataset, the EPS is the second best method among the others.

Knowing that the EPS algorithm performs well considering the "*accuracy rate*" criteria, we would also like to evaluate its computational complexity in terms of number of detectors generated. Since the number of detectors is only relevant for the family of the artificial immune systems (AIS) algorithms (i.e. not in MLP or Knn), we compare the EPS with the AIRS in the following sub-section.

## 4.4 Comparison of the EPS and AIRS In Terms of Number of Detectors

Watkins and et al (Watkins 2001; Watkins and Boggess 2002; Watkins, Timmis and Boggess 2004) presented a new supervised learning paradigm, resource limited artificial immune classifiers, inspired by mechanisms exhibited in biological and artificial immune systems. They called it AIRS (Artificial Immune Recognition System). Its performance has been investigated for UCI datasets and results were remarkable. AIRS1 is the first version of the algorithm but its algorithmic complexity is high. AIRS2 (Watkins and Timmis 2002) has been developed and it has lower complexity, higher data reduction percentage and a few decreases in accuracy.

In this section, the comparison between AIRS1, AIRS2 and EPS is presented, all of which are in the family of artificial immune based classifiers. We compared these methods according to the number of detectors they need. This criterion is a measure of the computational complexity such that the lower the number of detectors, the less memory space is needed.

In this experiment, the results are reported using 10-fold cross-validation. The first numbers in each cell is the number of detectors and the second number after the '/' sign is the percentage of reduction in the size of the training set.

The reduction value for the EPS is 76.5% on the IRIS dataset, while this value is 64.4%for the AIRS2. The average number of

| | Naive Bayes | MLP | LibSVM | Decision Tree (C4.5) | Rule Induction | Knn (k = 5) | AIRS | EPS |
|---|---|---|---|---|---|---|---|---|
| **IRIS** | 95.33 | 96.67 | 93.67 | 93.33 | 94.67 | 96.00 | 95.33 | **97.33** |
| **Wine** | **98.31** | 97.19 | 69.10 | 91.01 | 88.76 | 69.10 | 95.42 | 97.22 |
| **Segment** | 79.52 | **96.23** | 95.58 | 95.41 | 94.42 | 93.72 | 86.96 | 95.45 |
| **P2P** | 93.98 | 97.03 | 98.07 | 93.08 | 93.08 | **99.52** | 93.05 | 98.83 |

Table 7. Correct detection rate of the EPS and other classification algorithms on the benchmark datasets

| Data Set | Training Set Size | AIRS1 | AIRS2 | EPS |
|---|---|---|---|---|
| **IRIS** | 135 | 54/ 60% | 48/ 64.4% | **31.7/ 76.5%** |
| **Wine** | 160 | 92/ 42.5% | 101/ 36.9% | **45.9/ 71.3%** |
| **Segment** | 2079 | 375/ 81.9% | 224/ 89.2% | **149.8/ 92.8%** |
| **P2P** | 29490 | 673/ 97.7% | **585/ 98.0%** | 811/ 97.2% |

Table 8. Number of detectors employed in AIRS1, AIRS2, and EPS algorithms on the benchmark dataset

detectors for the EPS is 31.7, while it is 48 for the AIRS2. The number of detectors the EPS requires is about two times less than that of the AIRS2. The results achieved on the Wine dataset are similar to those on the IRIS dataset indicating the EPS as the bestmethod followed by AIRS1. On the Wine dataset, the EPS shows the highest reduction value (71.3%) which is two times more than that of the AIRS2. The number of detectors in the EPS is two times less than AIRS2.

The same results are observed on the Segment dataset. Finally, on the P2P dataset, all the three methods show the same results and the reduction value is around 98%.

## 5. Conclusions

In this paper, we presented an enhanced version of positive selection algorithm in which the number of detectors is significantly reduced in comparison with the original positive selection. Not only does the proposed method produce less number of detectors, it also outperforms other evaluated methods according to the accuracy. We applied clustering in the preprocess phase to reduce the number of detectors, and performed overlap checking in the training phase in order to increase the accuracy of the classifier. We also proposed two methods to deal with exclusion (samples not covered by any detectors) and ambiguity (samples covered by more than two detectors from different classes). The experimental results confirm that our proposed method outperforms other artificial immune-based algorithms (AIRS1 and AIRS2) in terms of number of detectors and accuracy rate. We also compared the proposed method with well-known non-artificial immune-based algorithms (C4.5 decision tree and Knn), and the results demonstrated that our proposed method is at par with them in terms of accuracy.

## References

[1] Ataser, Z. (2013). *Variable Shaped Detector: A Negative Selection Algorithm*. PhD Thesis. The Graduated School of Natural and Sciences of Middle East Technical University.

[2] Castro, L. N., Timmis, J. I. (2003). Artificial immune systems as novel soft computing paradigm. *Soft Computing-A Fusion of Foundations, Methodologies and Applications,* 7 (8) 526-544. Springer-Verlag.

[3] Dasgupta, D., Ji, Z., Gonzalez, F. (2003). Artificial Immune System (AIS) Research in the Last Five Years. *In*: Proceedings of Congress on Evolutionary Computation (CEC '03), 1, 123-130.

[4] Ebner, M., Breunig, H-G., Albert, J. (2002). On the Use of Negative Selection in an Artificial Immune System. *In*: Proceedings of Genetic and Evolutionary Computation Conference (GECOO '02), p. 957-964.

[5] Forrest, S., A., Perelson, S., Allen, L., Cherukuri, R. (1994). Self Non-self Discrimination in a Computer. *In*: Proceedings of 1994 IEEE Symposium on Research Security and Privacy.

[6] Gonzalez, F., Dasgupta, D. (2003). Anomaly Detection Using Real-Valued Negative Selection. *Journal of Genetic Programming and Evolvable Machines,* 4 (4) 383– 403.

[7] Gonzalez, F. (2003). A Study of Artificial Immune Systems Applied to Anomaly Detection. PhD Thesis. University of Memphis.

[8] Gonzalez, F., Dasgupta, D., Nino, L. F. ( 2003). A Randomized Real-Valued Negative Selection Algorithm. *In*: Proceedings of the 2nd International Conference on Artificial Immune Systems ICARIS. LNCS 2787: 261-272.

[9] Ji, Zh., Dasgupta, D. (2004). Real-Valued Negative Selection Algorithm with Variable-Sized Detector. *In*: Proceedings of Genetic and Evolutionary Computation Conference (GECOO '04). LNCS 3102: 287-298.

[10] Mohammadi, M., Raahemi, B., Akbari, A., Moeinzadeh, H., Nassersharif, B. (2011). Genetic-based minimum classification error mapping for accurate identifying peer-to-peer applications in the internet traffic. *Expert Systems with applications*, 38 (6) 6417-6423.

[11] Mohammadi, M., Raahemi, B., Akbari, A., Nassersharif, B. (2012). New Class-Dependent Feature Transformation for Intrusion Detection Systems. *Journal of Security and Communication Networks*, 5 (12) 1296-1311. John Wiley.

[12] Stibor, T., Mohr, P.H. , Timmis, J., Eckert, C (2005). Is negative selection appropriate for anomaly detection? *In*: Proceedings of Genetic and Evolutionary Computation Conference (GECCO '05), p. 321–328. ACM Press.

[13] Stibor, T., Timmis, J., Eckert, C (2005). A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques. *In*: Proceedings of the 4th International Conference on Artificial Immune Systems. LNCS 3627: 262-275. Springer-Verlag.

[14] Stibor, T., Timmis, J. (2007). Comments on Real-Valued Negative Selection vs. Real-Valued Positive Selection and One-Class SVM. *In*: Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007), p. 3727-3734.

[15] Watkins, A. (2001). *AIRS: A Resource Limited Artificial Immune Classifier*. Master of Science Thesis. Mississippi State University.

[16] Watkins, A., Boggess, L. C (2002). A Resource Limited Artificial Immune Classifier. *In*: Proceedings of Congress on Evolutionary Computation (CEC 2002), 1, 926-931.

[17] Watkins, A., Timmis, J., Boggess, L. C. (2004). Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. *Journal Genetic Programming and Evolvable Machines*, 5 (3) 291-317.

[18] Watkins, A., Timmis, J. (2002). Artificial Immune Recognition System (AIRS): Revisions and Refinements. *In*: Proceedings of the 1st International Conference on Artificial Immune Systems ICARIS, p. 173-181.

[19] Wu, P., Zheng, X. (2012). An Improved Variable-radius Real-valued Negative Selection Algorithm. *Journal of Information & Computational Science*, 9 (16) 4713-4720. Available at http://www.joics.com.

[20] Zhang, F., QI, D. (2012). A Positive Selection Algorithm for Classification. *Journal of Computational Information Systems*, 8 (1) 207-215. Available at http://www.Jofcis.com.

[21] Zheng, X., Zhou, Y., Fang, Y. (2013). The dual Negative Selection Algorithm Based on Pattern Recognition Receptor Theory and Its Application in Two-class Data Classification. *Journal of Computers*, 8 (8) 1951-1959.