



# Negative Selection Based Data Classification with Flexible Boundaries

Lena Nemati <sup>a</sup>, Mojtaba Shakeri <sup>b,\*</sup>

<sup>a</sup> Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

<sup>b</sup> Assistant professor, Computer Group - Faculty of Engineering, Guilan University, Rasht, Iran

Received 09 May 2018;

Revised 22 July 2018;

Accepted 11 August 2018;

Available online 18 September 2018

---

## Abstract

One of the most important artificial immune algorithms is negative selection algorithm, which is an anomaly detection and pattern recognition technique; however, recent research has shown the successful application of this algorithm in data classification. Most of the negative selection methods consider deterministic boundaries to distinguish between self and non-self-spaces. In this paper, two negative selection based algorithms are proposed for two-class and multi-class classification problems; using a Gaussian mixture model which is fitted on normal space to create a flexible boundary between self and non-self-spaces, by determining the dynamic subsets of effective detectors to solve the problem of data classification. Initialization of effective parameters such as the detection threshold, the maximum number of detectors etc. for each dataset, is one of the challenges in negative selection based classification algorithms, which affects the precision and accuracy of the classification; therefore, an adaptive and optimal calculation of these parameters is necessary. To overcome this problem, the particle swarm optimization algorithm has been used to properly set the parameters of the proposed methods. The experimental results showed that using a Gaussian mixture model and dynamic adjustment of parameters such as optimum number of Gaussian components according to the shape of the boundaries, creation of appropriate number of detectors, and also automatic adjustment of the training and testing thresholds, using particle swarm optimization algorithm as well as utilization of a combinatorial objective function has led to a better classification with fewer detectors. The proposed algorithms showed competitive performance compared with some of the existing classification algorithms, including several immune-inspired models, especially negative selection ones, and other traditional classification methods.

**Keywords:** Classification, Negative Selection Algorithm, Gaussian Mixture Model, Particle Swarm Optimization, Flexible Boundaries.

---

## 1. Introduction

Along with the advancement of science, many methods and models for solving complex engineering and technical problems have been proposed. Each of these methods presents a special field of science for providing optimal solutions. Methods that are currently popular can be divided into traditional symbolic methods, statistical methods, artificial intelligence and computational

intelligence. The artificial immune system (AIS), as a branch of computational intelligence, has gained significant success since it emerged in the 1990s and was quickly spotlighted by researchers aiming to design models and techniques and presenting solutions for complex engineering problems. The inherent properties of the natural immune system, such as self-organizing, self-adaptiveness, classification, and high distribution, with other useful features such as learning, memory,

---

\* Corresponding author. Email: Mojtaba.shakeri@gmail.com

robustness, fault tolerance, generalization, feature extraction, pattern recognition have made it more powerful compared with similar systems and enable researchers to perform system computations called artificial immune system [1-4].

The immune system has evolved to allow robust responses against pathogens while avoiding autoimmunity. This is notably enabled by stimulatory and inhibitory signals which contribute to the regulation of immune responses. In the presence of a pathogen, a specific and effective immune response must be induced and this leads to antigen-specific T-cell proliferation, cytokines production, and induction of T-cell differentiation toward an effector phenotype. [5]. The AIS application areas can be categorized into machine learning, anomaly detection, and optimization [6, 7]. The most important and effective mechanisms of artificial immune systems are: negative selection, clonal selection, immune network theory and danger theory [8]. The negative selection algorithm is one of the important branches of artificial immune systems. This algorithm, presented by Forrest et al. in 1994, to identify the virus intrusion in computer systems by detecting data manipulation. The negative selection algorithm is defined in training and testing phases, in which the process of identifying and detecting anomalies is made by detectors that simulate antibodies in the natural immune process.

The negative selection algorithm has many applications. Due to its uniqueness ability and high performance in self and non-self discrimination, some of the diagnostic applications of this algorithm are anomaly detection, computer virus detection and pattern recognition, intrusion detection in network, network security and fault detection. Data mining, classification, computer security and adaptive control are also other applications of negative algorithms [9].

Classification is an important method in data mining. Classification algorithms can permanently distinguish between the elements of each class and classify the new samples automatically using the resulting rules by learning from a large set of pre-classified data. Due to the importance of Classification in various fields such as

machine learning, image retrieval, machine vision and medical diagnosis, many studies have been done on classifiers [10]. Multi-class classification is also considered as an important issue in machine learning. Each instance in the learning set belongs to a set of pre-defined labels in multi-class classification. Multi-class classification problem refers to assigning each observation to one of the  $m$  classes, and the purpose is to create a function that correctly predicts the new data point class [11].

In most researches conducted in the classification with negative selection algorithm, rarely used the flexible boundary for self and non-self-space, as well as the variable and optimum number of detectors. In this research, the combination of the negative selection algorithm with Particle Swarm Optimization (PSO) algorithm and the Gaussian Mixture Model (GMM) are used. The optimum values of the effective parameters in the algorithms are determined by the PSO algorithm; their adjustment will improve performance and reduce the number of detectors. The proposed classifiers can be automatically fitted to each dataset and provide desirable precision in classification of two-class and multi-class datasets.

In this section, an introduction to the research was presented. In the next section, we will investigate the background of study and then the methods used in the proposed algorithms include the Expectation Maximization (EM) algorithm, particle swarm optimization and also the Distribution Estimation Negative Selection Algorithm (DENSA) as the base algorithm. Afterwards the proposed methods are presented in more detail. After providing the methods, a range of evaluations is performed on different datasets, to illustrate the capability of the methods. In the end, the research findings are also discussed.

## **2. Literature Review**

Regarding the capability of negative selection algorithms in the classification, in this section some of the classification methods and algorithms based on negative

selection that are presented by researchers in different articles, are investigated.

In 2004, Lee and Sim used a negative selection algorithm for DNA patterns classification at the nucleotide and amino acid levels. Empirical experiments have shown that, this method is effective when a size of pattern is big and not fixed like DNA sequences and the performance of the pattern detection depends on the parameters such as the number and length of detectors and the threshold [12].

Markowska Koczmar and Kordas introduced the M-NSA algorithm in 2006 for multi-class data classification. The algorithm uses a fixed detection threshold distance for not identifying the self data and another detection threshold distance to recognize non-self data. The LED-24 Digit and Fisher Iris datasets have been used to evaluate this method [13].

In 2007, Igawa and Ohashi provided a data classification algorithm called DAIS based on the principle of distinguish between the self and non-self cells by the T cell, or the negative selection. In this method, the classification is performed by artificial lymphocyte cells and a detection distance threshold. This algorithm can reduce the number of memory cells by generating detectors with different detection radius, but it's not very precise compared with other popular classifiers [14].

In 2008, Markowska Koczmar and Kordas, modified the M-NSA receptor set, and an additional algorithm was proposed to improve the results, in which two refinement functions were used to modify the receptors. The new model was named MINSA. The results showed that MINSA has a better accuracy and performance than the M-NSA, but its accuracy is less than AIRS. The classifier was tested on Fisher Iris, Cleveland Heart Disease, and Wisconsin Breast Cancer Dataset [15].

Igawa and Ohashi, in 2009, improved the DAIS algorithm and presented a new algorithm called ANSC (Artificial Negative Selection Classifier). This algorithm uses a cutting method to reduce noise effects and

classification costs. Pima Diabetes, Sonar, Fisher Iris, and other Datasets have been used to test this method [16].

An algorithm called MCA by Lian and Yong-Kang was introduced in 2010. The algorithm is designed for multi-class classification and follows a structure similar to the M-NSA. The mutation algorithm is used to generating new receptors. Experimental results have shown the high accuracy compared to other methods. This algorithm was tested on Sonar, Fisher Iris and Ionosphere datasets [17]. Oliveira and Drummond presented a modified real-valued negative selection (RNS) for classification. The purpose of this method is to optimize the number of detectors and detectors radius per class. The detection rates obtained in experiments have been analysed by the data complexity criterion. Implementation of the proposed algorithm has been done on the available datasets in the UCI database [18].

In 2011, Seyed Fakhari and Eftekhari Moghadam presented a negative selection based algorithm called NSSCA to classify data. The proposed algorithm used an adaptive technique to calculate predefined parameters such as detection threshold for each problem in an automated and adaptive manner regardless of setting any parameters. In this study, Iris, Diabetes and Cancer datasets have been used to test the algorithm [19]. An updated version of this article was presented in 2014, which explains the proposed idea broadly. The experimental results showed that the NSSCA is useful for solving problems with complex structure [20]. Elberfeld and Textor sought to resolve the question of whether the negative selection algorithm could be used for real-world problems provided two algorithms for classification. Proposed algorithms have used prefix trees instead of patterns to speed up the implementation of the NSA [21].

Soliman and Adly, in 2012, provided the Q-NSA algorithm for associative classification. This algorithm combines the concepts of quantum computing and negative selection algorithm to build an efficient classifier, which leads to the production of rule detectors to find the best subset of associative rules [22].

Zheng et al., in 2013, introduced a hybrid algorithm called PRR-2NSA to solve real-valued negative selection algorithms drawbacks such as low efficiency and size of classifier set. The datasets used in this method were Fisher Iris, Wisconsin Breast Cancer and Chess [23].

In 2014, Lytvynenko offered a supervised hybrid classification algorithm called HSNA based on the combination of negative selection algorithm and the PSO algorithm, which is useful for solving multi-class classification problems. The main idea is to modify the learning phase in the negative selection algorithm to change size and location, to prevent redundancy of detectors, and to provide better coverage of the self-space. Comparison of the results of the experiments has clearly shown that the proposed approach has desirable accuracy and less computational complexity, is more efficient compared with other classification algorithms [24].

Jantan et al., in 2015, used the negative selection algorithm as a bio-inspired classification method for classifying academic management. Experimental results have shown that the negative selection algorithm has the potential to classify academic management and employ the right person to the proper work [9].

Mishra and Bhusry, in 2016, used a combination of negative selection and artificial bee colony algorithms to improve the convergence behaviour of the negative selection algorithm. Before entering the steps of the negative selection algorithm, optimization of data using artificial bee optimization has been used in the proposed method to overcome some limitations of the negative selection algorithm, such as local minimum and computational complexity. Empirical results have shown that the proposed method has a high degree of accuracy in classifying the Fisher Iris dataset and is a very effective method compared to other approaches for random searches [25].

In 2017, Zhu and Chen presented a single-class classification model for performing on big data based on Voronoi diagrams called VorNSA. In this research, the authors claimed that the classification based on the negative selection algorithm has many limitations. The experiments were conducted on two sets of synthetic

datasets called SDS and skin segmentation set [7]. Abreu et al., presented a new method for the detection and classification of noise in speech based on the negative selection algorithm and the dual-tree complex wavelet transform. The proposed algorithm recognizes the failure of a speech sentence. In this research, an artificial dataset has been used to test the method [26].

Rashid et al., in 2018 proposed a novel classification approach for the classification of complex EEG brain signals. In this study, dimensionally reduction of data was carried out by applying two hidden layered stacked out-encoder. GA optimized detectors were trained using NSA for detection and classification. The experimental results achieved high accuracy for multi-class data classification [27]. Researchers introduced a new classification algorithm based on Voronoi diagrams (VorNSA) and an immune detection process of VorNSA to cope with the challenge of big data. The scheme of the detector generation process changed from the traditional "Random-Discard" model to the "Computing-Designated" model by VorNSA. The results have shown that the time spent by VorNSA was averagely decreased by 87.5% compared to the traditional NSA methods in UCI skin dataset [7].

### **3. Methods**

As described in the introduction, the classification process is performed to produce efficient detectors and self-space modelling using the Gaussian mixture model, the adjustment of the main parameters of the algorithm and their initialization through the particle swarm optimization method in the proposed algorithms. In this section, the expectation maximization and PSO algorithms that are used in the proposed methods are explained. Subsequently, DENSA algorithm is discussed as the basis for the proposed algorithms.

#### *3.1. Expectation Maximization Algorithm*

One of the methods for estimating the model parameters, when the observations are fully available, is the expectation maximization method. In this method, the

parameters are estimated to maximize the likelihood function of a model; however, in most cases, the logarithm likelihood maximization is performed [28]. EM method is a repetitive method for calculating the maximum estimation for models with hidden variables. When the vector of observations is not complete and some data are not available due to noise or missing values and it is not possible to find the model parameters, the concept of the hidden variable is used.

Each repetition of the EM method involves two steps of expectation and maximization, and they are as follows:

- Step 1: Initialize the model parameters.
- Step 2: Expectation (Step E): At this stage, estimating the hidden variable vectors are performed using the training data, the current parameters of the model and calculating the probability of subsequent variables.
- Step 3: Maximization (step M): parameters values are selected to maximize the likelihood function and update the model parameters based on the hidden variables computed by the MLE method, having hidden variables and training data.
- Step 4: Investigate ML convergence or parameter values and repeat steps 2 and 3.

As the likelihood value increases in each repetition, finally the EM algorithm is converged; therefore, estimates obtained from this result tends to the maximal value of their likelihood. The purpose of using the EM algorithm for a given Gaussian mixture is to maximize the likelihood function by obtaining parameters such as means and covariance of components and mixing coefficients over the following steps [28]:

- Initialization of GMM parameters: The EM algorithm requires initializing some initial assumptions for model parameters such as mean ( $\mu_j$ ) and ( $\Sigma_j$ ) covariance, and mixing coefficients ( $\pi_j$ ) that are performed at this stage and the initial value of the likelihood logarithm is computed.

- Step E: Calculate the value of the posterior probability of parameter  $\gamma(z)$  according to the current values for the parameters, taking into account the number of effective points assigned to the k-cluster and the total number of data points (N) as well as the amount and is measurable using formulas (1) and (2):

$$\pi_j = \frac{N_k}{N} \quad (1)$$

$$\gamma(Z) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^k \pi_j N(x_n | \mu_j, \Sigma_j)} \quad (2)$$

- Step M: The re-estimation of the model parameters takes place using the calculated values in the previous step and is calculated by relation (3).

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^k \pi_k N(x_n | \mu_k, \Sigma_k) \right\} \quad (3)$$

- Likelihood logarithm is calculated, and in the case of non-convergence, it returns to step E.

### 3.2. Particle Swarm Optimization Algorithm

The particle swarm optimization algorithm (PSO) is a population-based optimization method of swarm intelligence algorithms. In PSO, the population is called swarm, congestion, or group, and the word particle refers to each individual of the population. Particles seek to achieve optimum response based on an objective function, thus they move and there is a velocity. Velocity and position parameters are considered for each particle. Each particle represents a response or a solution to the problem space, and it also has memory, and can store the best position that it has achieved or the local best particle (personal best). It has the ability to communicate with the particles around it. Hence, on the one hand, it is inclined to return to its own optimum position and, on the other hand, to move towards the best neighbourhood points and the global best in the population. The velocity and position of the particles in the search space are influenced by the direction of motion, the best position observed by

the particle itself, and the best position of the particles. As time passes, the particles accelerate to particles that have a higher degree of fitness. Updating velocity and position in a repetitive process continues until reaching the goal condition of the algorithm that is to achieve an optimum solution. The particle swarm optimization algorithm is as follows [29]:

- Step 1: A population of particles with random positions and velocities are initialized throughout the d-dimensional search space.
- Step 2: Evaluating the position of each particle is performed by the objective function, which means that the magnitude of the fitness value of each particle is calculated.
- Step 3: If each particle's new fitness value is higher than its predecessor's best fitness, then the substitution will take place, thus pbest will be replaced with the new position. The fitness of the current particle is also compared with the fitness of the global best particle; if it is better gbest will be replaced either.
- Step 4: The particle velocity update is carried out by formula (4):

$$V_i [t+1] = V_i [t] + C_1 \times \text{rand}_1 \times (\text{pbest} - X_i [t]) + C_2 \times \text{rand}_2 \times (\text{gbest} - X_i [t]) \quad (4)$$

Where  $i$  is the particle index.  $C_1$  and  $C_2$  are constant coefficients of acceleration; they are considered a number between zero and two, and respectively represent the amount of influence and contribution of each personal best and the global best of the population to determine the new position of each particle, based on what has been observed so far.  $\text{rand}_1$  and  $\text{rand}_2$  are random values between zero and one.  $V_i [t]$  And  $X_i [t]$  indicate the velocity and position of the particle in time  $t$ , respectively.  $\text{pbest}$  is the best position that the particle ever had, and  $\text{gbest}$  is the best particle in the population. To prevent excessive movement of a particle, such as the divergence of the velocity vector, speed variations are limited to the interval  $(V_{\min}, V_{\max})$ .

The upper and lower limits are determined according to the type of problem.

- Step 5: Movement of particles to new position is accomplished by formula (5):

$$X_i [t+1] = X_i [t] + V_i [t+1] \quad (5)$$

- Step 6: Go back to step 2 to meet the conditions for stopping the algorithm.

### 3.3. Distribution Estimation Negative Selection Algorithm

The DENSA algorithm, an improved version of the NSA, was presented by Fouladvand et al., in 2015, and uses the Gaussian mixture model that tries to create an efficient and flexible boundary for self samples. With strong statistical backgrounds, in this method GMMs can cover the self-space efficiently and also flexibly choose components distribution, especially when the full covariance matrixes are used. Hence, the generating detectors through this sophisticated model can help in design to efficient NSA, even in real-world contexts. Incorporating GMMs into a negative selection algorithm provides good opportunities to develop an efficient, and robust algorithm overcoming its limitations [30]. In 2017, an extended and revised version of DENSA was presented in order to make it more efficient in real world applications [31]. The probabilities of Gaussian mixture model are used to distribute detectors on the non-self-space more efficiently, which together with a suitably defined objective function that attempts to cover non-self-space, enables DENSA to be more efficient in real-world applications. Similar to the NSA, the goal of DNESA is to generate a set of detectors that covers the non-self-space effectively.

In the training phase of DENSA, first maps the data in the interval between zero and one; then it compares the randomly generated samples with a GMM, which is suitably fitted on the self-space to generate a predefined number of detectors. In the nutshell, instead of comparing a randomly generated pattern to all self- samples, DENSA compares these samples with a flexible Gaussian

components which are representing the self-space. As DENSA generates the detectors, it calculates an objective function for the future decision on the optimum number of detectors. This objective function selects the optimum number of detectors based on estimating effect of the detector set. In DENSA, an objective function is considered based on three main terms: a first term for evaluating the detection rate; a second one for estimating the false positive rate; and a last one for keeping the size of the detector set as small as possible. Thus, the objective function used in order to satisfy these constraints is defined as relation (6):

$$\text{Objective}(i) = \left( \begin{array}{l} W_1 \times \text{DetectionRate}(S, G, D) + \\ W_2 \times \frac{1}{1 + \text{FalsePositiveRate}(S, G, D)} + \\ W_3 \times \frac{1}{|D|} \end{array} \right) \quad (6)$$

Where  $S$  is the self-space,  $G$  is the Gaussian mixture model of normal space and  $D$  is the detector set. The Parameters  $W_1$ ,  $W_2$  and  $W_3$  represent instead weighted values of the single sub-objectives, whose increase the flexibility of the objective function and cope to different applications and datasets.

Detection Rate ( $S, G, D$ ) and False Positive Rate ( $S, G, D$ ) denote the estimated detection rate and the false positive rate of the anomaly detection system respectively using a validation set and the current detector set  $D$ . It is easy to investigate how the optimum number of detectors is obtained by maximize the objective function in relation (6). This means maximize the estimated detection rate, and minimize the estimated false positive rate, as well as the size of the detector set. In fact, as the number of efficient detectors increases, the first two terms also increase. In other words, by increasing in the number of detectors, the last term of the relation (6) decreases and, as a result, the total function value decreases. The DENSA test phase is performed in two parts:

- Calculating the Euclidean distance of an input sample from the nearest detector.

- Classifying the input sample as normal or abnormal, based on the calculated distance in the previous step and a threshold called the Threshold\_2, which is defined in formula (7):

$$\text{Output} = \begin{cases} 1 & \text{if Dis(sample, Detector) > Threshold\_2} \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

Threshold\_1 and Threshold\_2 are important parameters which have a significant impact on the accuracy of DENSA. Threshold\_1 is a parameter used in the training phase for evaluating if randomly generated sample is far enough from the Gaussian components in the normal space in order to be considered as a detector. While the Threshold\_2 is another control parameter used in the testing phase and evaluates the proximity of a sample to a detector, to be detected as abnormal or non-abnormal. As Threshold\_2 increases, more test samples are assumed anomalies, therefore the rate of false positive and true positive rates increase too.

### 3.4. DENSA Limitations

There are some limitations in DENSA which we are trying to resolve them in the proposed method:

- Selecting the number of Gaussian components ( $k$ ) corresponding to the first local minimum of Bayesian Information Criterion (BIC) values; which may not be optimal and reduce the efficiency of the algorithm.
- The need to set the Threshold\_1, which is obtained in DENSA by dividing data into three sets of training, testing, and validation, as well as calculating the objective function on the validation set to obtain its optimum value.
- The requirement to set the Threshold\_2, which is achieved with trial and error.
- Having no information about the maximum number of detectors ( $|D|$ ), which by default has a predefined value.

- The main application of the DENSA method is anomaly detection and has not been developed for multi-class classification.

#### 4. The Proposed Methods

In this section, in order to eliminate the weaknesses and limitations of the DENSA and to classify through the development of this technique and also to overcome the challenges in some of the negative selection algorithms; the proposed methods which are designed for classification based on negative selection will be presented in details. These methods focus on the production of effective detectors through a flexible boundary for self patterns. These detectors utilize a Gaussian mixture model fitted on a normal space. The proposed algorithms have the ability to determine the effective subset of detectors dynamically; using the capabilities of GMM. In these algorithms, the combination of proposed algorithms and particle swarm optimization is used to set certain parameters such as thresholds and number of detectors. The training process of the proposed classifiers is generally composed of the four steps; the last three steps are repeat until the stop condition is reached:

- Normalization and mapping input data into zero and one interval.
- Initializing the parameters and fitting the Gaussian mixture model.
- Generating the detector sets.
- Calculate the objective function and selecting the best detector set.

The purpose of the proposed methods is to adequately and effectively cover non-self-space by detectors, like the other NSA-based methods. In order to train the methods, first, the feature values of input data are normalized and mapped into zero and one. Then the Gaussian mixture model fits on the normalized data using the EM algorithm. Estimating of threshold values and other model parameters such as the number of

detectors for each dataset can be done using an optimization algorithm. In these methods, the combination of DENSA with particle swarm optimization algorithm is used, which is the reason we call the proposed methods PSO-DENSA in two-class mode and MPSO-DENSA in multi-class mode. Moreover, the high convergence capability of the particle swarm optimization, the ability to work with all types of data whether continuous or discrete are the important motivations for choosing PSO.

##### 4.1. The Proposed Two-Class Classifier (PSO-DENSA)

As mentioned, using the PSO algorithm in this method, a number of particles are generated randomly which are representative of the parameter values such as the number of detectors ( $D$ ),  $Threshold\_1$  and  $Threshold\_2$  and the number of Gaussian components ( $K$ ) as Fig 1. Then the negative selection processing is performed according to the values of the existing parameters for every particle in each PSO iteration, until a classifier model can be created by generating the detector set. Instead of comparing the random generated pattern with all self-samples, it can be compared to a number of flexible Gaussian components representing the self-space in the detector generation process. In the following, the classifier model produced by each particle is evaluated using an efficient objective function which is based on three criteria: detection rate, false positive rate, and the size of the detector set, according to Formula (6), so the best particle is identified and the location of each particle is updated. The above steps are repeated until the PSO termination conditions are met. Lastly, the classifier model corresponding to the best particle is chosen as the conclusive model. The revised version of DENSA is the basis of design the proposed algorithms. Fig 2 shows the PSO-DENSA detector generation. In the proposed algorithm, several steps of DENSA method are eliminated because the parameters such as the number of Gaussian components, both thresholds and the maximum number of detectors are automatically determined according to the Fig 3.



1	2	3	4
Number of detectors	Number of Gaussian components	Threshold_1	Threshold_2

Fig. 1. The structure of each particle in PSO-DENSA.

Upon completion of the algorithm at the training phase, the classifier produced by the best particle, which is the same as global best and generates the highest value of the objective function, is selected as the trained classifier. The testing phase is completely similar to DENSA algorithm. The proposed algorithm calculates the Euclidean distance of a test sample from the nearest detector, and then decide about the sample class using the calculated distance and also the comparison with the threshold\_2 for the testing process. If the calculated distance is greater than threshold\_2, which is determined by the optimal particle, then it is a self-sample and otherwise non-self.

1. Particle decoding and extracting the parameters (Gaussian components, both thresholds and the maximum number of detectors).
2. Fitting a GMM on training data using the EM algorithm.
3. Repeat
  - 3.1 Generating a random pattern  $x_i$  in the interval between zero and one.
  - 3.2 Adding  $x_i$  to Detector set D if  $p(x_i) < \text{Threshold}_1$
4. Until  $|D|$  is smaller than the maximum number of detectors.

Fig. 2. Detector generation process of PSO-DENSA.

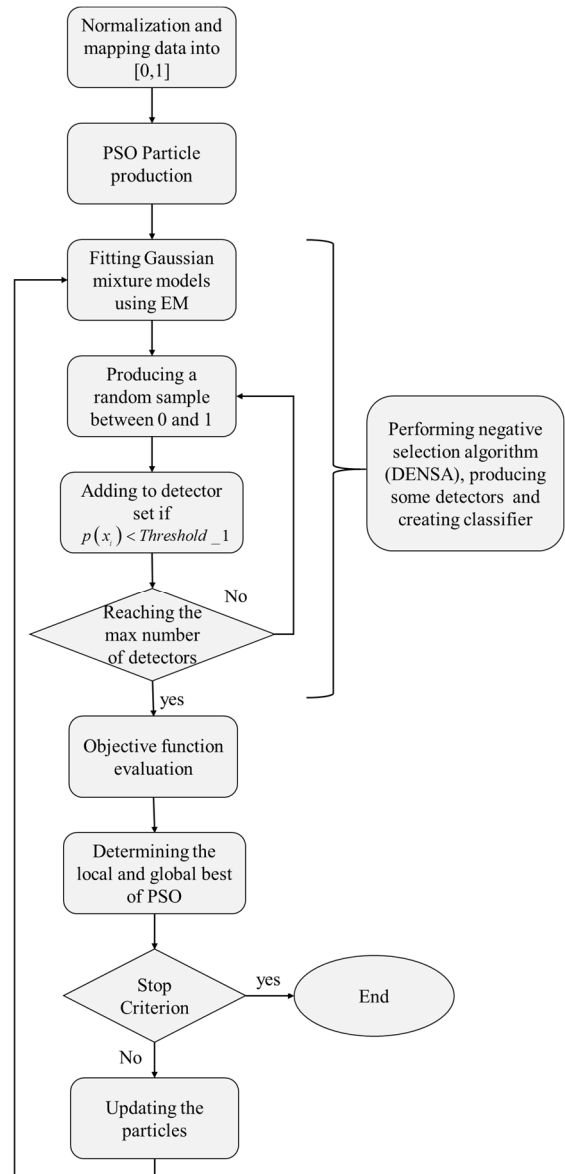


Fig. 3. The training process for the proposed two-class classifier model.

#### 4.2. The Proposed Multi-Class Classifier (MPSO-DENSA)

The Negative selection based classifiers consist of  $m$  sets of detector for assigning pattern to corresponding  $m$  classes. Thanks to these sets it is possible to determine the classification result of each testing pattern. The structure of these classifiers is shown in Fig 4. To apply the negative selection mechanism and generate each of detector set, it is necessary to divide the dataset into two classes; self and non-self classes. The *one vs all* classification method is used to classifying each class.

One class is considered as self each time, and non-self classes consists of all the remaining classes; this process is repeated in the number of classes of the problem. Data belonging to self class are named self data and the others are named non-self data.

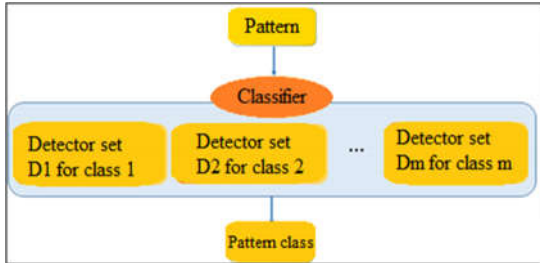


Fig. 4. The general structure of multi-class negative selection based classifiers.

Each detector set, which is generated in training phase, contains the structures, which get more stimulated by patterns belonging to classes not corresponding to the set and do get less stimulated by patterns from the corresponding class. Thus, the output of a detector set that is less stimulated is the pattern class. The stimulation can be determined by taking into account the number of activated detectors.

The proposed multi-class classification algorithm, MPSO-DENSA, is an extended version of PSO-DENSA method. As a result, the previous algorithm should be modified; in order to this purpose, the structure of each particle is changed. In PSO-DENSA, the length of each particle is four, because each particle contains four parameters. Therefore, in n-class mode, the length of particle is (4×n). By generalizing the PSO-DENSA, the structure of each particle in MPSO-DENSA is obtained.

Fig 5 shows this structure, where  $P_{i,j}$  is the i-th parameter for j-th class.

The MPSO-DENSA training process is shown in Fig 6. The global objective function of MPSO-DENSA is calculated by averaging the objective functions of different classes using relation (8):

$$\text{Objective} = \frac{\sum_{i=1}^m \text{Objective}(i)}{m} \quad (8)$$

1	2	3	4	5	6	7	8	...	4×(n-3)	4×(n-2)	4×(n-1)	4×n
$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	$P_{1,4}$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	$P_{2,4}$	...	$P_{n,1}$	$P_{n,2}$	$P_{n,3}$	$P_{n,4}$

Fig. 5. The structure of each particle in MPSO-DENSA.

Like other negative selection based multi-class classifiers, each element contains a detector set corresponding a class, if only one detector set ( $D_i$ ) is not able to identify a pattern in the testing phase, this pattern belongs to i-th class.

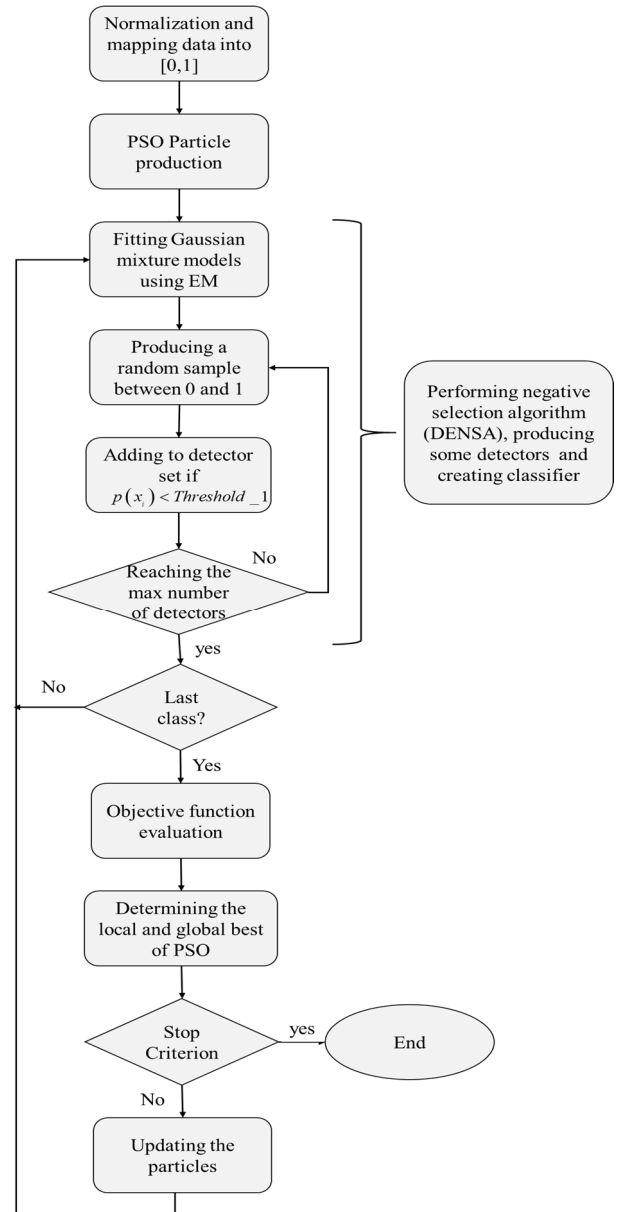


Fig. 6. The training process for the proposed multi-class classifier model.

There are two challenges in the classification to the proposed multi-class method:

- First: More than one detector set fail to identify the pattern (more than one class consider for the pattern). For resolve this challenge, the distance between the patterns with the closest detector in every set, that identified the pattern, is calculated. Finally, the pattern class is the one which has the greatest  $\beta_i$ .  $\beta_i$  can be calculated using formula (9), in which  $Det_i$  is the closest detector to the  $i$ -th class.

$$\beta_i = \text{Dist}(Det_i, \text{Pattern}) - \text{Thresh}_i \quad (9)$$

- Second: All detector sets identify the pattern (no class is recognized for the pattern). In this case, the class of the set whose the closest detector has the highest  $\beta_i$  value is presented as the selected class.

## 5. Evaluation of the Proposed Classifiers

In this section, we will evaluate and analyse the results of the proposed algorithms. The results of the experiments are presented in two general sections, including the evaluation of the proposed two-class classifier (PSO-DENSA) and multi-class classifier (MPSO-DENSA). In each section, we will first introduce the testbed, the dataset, and then continue to present and analyse the results of proposed algorithms and compare them with other methods on some interesting datasets.

### 5.1. Evaluation Criteria

Five classification criteria including detection rate, false positive rate, accuracy, number of detectors and objective function are used for evaluation and comparison of the methods. Formula (10) is used to calculate the detection rate, or the correct positive rate of the proposed PSO-DENSA algorithm, and formula (11) can be used to calculate the false positive rate. The accuracy of the classification is also obtained from equation (12).

$$\text{TruePositiveRate(TPR)} = \text{DetectionRate(DR)} = \frac{TP}{TP+FN} \quad (10)$$

$$\text{FalsePositiveRate(FPR)} = \text{FalseAlarmRate(FAR)} = \frac{FP}{FP+TN} \quad (11)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

In these formulas, true positives represent the number of abnormal samples that are correctly classified as abnormal. True negatives are the number of normal samples that are correctly classified as normal. False positives are the number of normal samples that are classified as abnormal and false negatives are the number of abnormal samples classified as normal incorrectly [32].

### 5.2. Evaluation of the Proposed Two-Class Classifier (PSO-DENSA)

Five synthetic datasets are used to evaluate this classifier in these experiments. Each dataset is partitioned into two parts; one of them is used for training, and the rest of for testing; only self-samples are used as training data. The test data is also divided into two parts; the first part is used to calculate the objective function (to evaluate the particles), and the rest are used to test the detectors generated after the training. Five synthetic datasets; Pentagonam, Cross, Triangle, Stripe and Ring, are used in evaluations. The proposed PSO-DENSA approach should have the ability to create a boundary between the self-space (red spots) and the non-self-space (blue spots). Each dataset is a geometric shape and has 2000 samples (1000 self and 1000 non-self samples) with the actual values as shown in Fig 7.

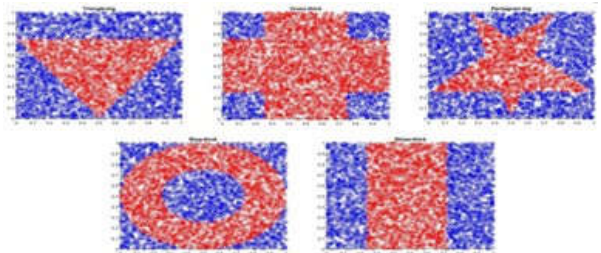


Fig. 7. Synthetic geometric datasets used by the PSO-DENSA method. (Red spots: self, blue spots: non-self).

### 5.2.1. Parameters Initialization

The proposed two-class PSO-DENSA algorithm has several adjustable parameters that should be set at the beginning of the algorithm. The first group of these parameters includes the parameters used in the objective function according to equation (6), which represent the weighted values of single sub-objectives associated with the detection rate, the false positive rate, and the number of detectors. In these evaluations, the weighted values have been set to ( $W_1 = 1, W_2 = 1, W_3 = 0.5$ ) in order to maximize the detection rate and minimize the false positive rate with similar importance in the objective function and both more important than minimizing the number of detectors. In the second group, the PSO control parameters include the number of particles, constant acceleration coefficients ( $C_1$  and  $C_2$ ), and inertia weight. In order to set these parameters, the trial-and-error method was used. In this process, number of particles 20, coefficient values of  $C_1$  and  $C_2$  two and inertia weight 0.9 have been selected.

### 5.2.2. Evaluation Results of the PSO-DENSA Algorithm

Table 1 shows the values of detection rate, false positive rate and number of detectors, the value of the objective function, and finally the classification accuracy of the PSO-DENSA algorithm on the Stripe dataset for several iterations which the objective function has improved. We can conclude that the PSO-DENSA method, in spite of decreasing detection rates in some execution, has a general upward trend in DR, according to Table 1. The false positive rate has also a decreasing trend on average and the algorithm utilizes the optimum number of detectors. The objective function values are continuously increasing, and despite the fluctuation in some of the repetitions, the algorithm has a high classification accuracy because the PSO-DENSA algorithm is capable of generating detectors in the non-self-space and has a robust detection rate on the synthetic dataset.

Table 1. The progress of PSO-DENSA on the Stripe Dataset.

rep	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)
2	63.79	85.04	13	7506.1	52.85
3	75.90	17.08	14	8326.1	26.91
27	36.98	06.17	25	8382.1	43.89
29	83.90	28.07	14	8411.1	72.91
31	100	54.15	8	8668.1	03.91
38	74.92	99.02	16	8990.1	71.94
44	84.95	61.05	21	9021.1	94.94
45	03.99	30.08	24	9141.1	17.95
49	70.97	05.05	26	9293.1	32.96
54	17.98	15.04	28	9422.1	01.97

In the following, we compare the PSO-DENSA algorithm with the DENSA algorithm [31]; it is necessary to mention that the DENSA method receives the different number of detectors as one of the input parameters, while in the proposed PSO-DENSA method, a combination of input parameters, including dynamic number of detectors are considered to optimize the performance of the algorithm. The results of this comparison can be seen in Tables 2 to 6 after the completion of all iterations and the final test:

Table 2. Comparison of PSO-DENSA and DENSA on Pentagon Dataset.

Method	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)	Time (S)	
						Train	Test
DENSA	92.21	15.12	499	1.7911	91.53	3265	13
PSO-DENSA	95.74	08.06	39	1.9689	93.48	12365	11

Table 3. Comparison of PSO-DENSA and DENSA on Cross Dataset.

Method	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)	Time (S)	
						Train	Test
DENSA	87.52	09.92	384	1.7854	89.62	2456	12
PSO-DENSA	100	00.04	27	2.00	96.32	10325	13

Table 4. Comparison of PSO-DENSA and DENSA on Stripe Dataset.

Method	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)	Time (S)	
						Train	Test
DENSA	97.42	12.29	599	1.8651	93.68	5469	13
PSO-DENSA	98.78	08.68	45	1.9082	97.93	11245	13

Table 5. Comparison of PSO-DENSA and DENSA on Triangle Dataset.

Method	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)	Time (S)	
						Train	Test
DENSA	98.21	18.11	524	1.8290	92.37	6542	12
PSO-DENSA	99.12	11.89	44	2.12	94.62	10296	12

Table 6. Comparison of PSO-DENSA and DENSA on Ring Dataset.

Method	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)	Time (S)	
						Train	Test
DENSA	91.42	09.13	540	1.8310	90.31	3569	13
PSO-DENSA	94.21	08.07	32	1.9585	94.35	12125	13

The results in Table 2 to Table 6 show the overall superiority of the PSO-DENSA algorithm compared to DENSA, while DENSA uses more detectors than the proposed PSO-DENSA algorithm, because DENSA has no knowledge about the optimum number of detectors and other parameters, and get the results using trial and error. In addition, the results show that the proposed algorithm has achieved higher percentages in terms of detection rate on datasets that have linear boundaries such as Cross, Stripe and Triangle (100%, 98.7% and 99.2% respectively), It seems also reasonable theoretically, because flexible boundaries are more easily separable. The proposed PSO-DENSA algorithm has achieved a higher percentage in terms of accuracy on Stripe dataset due to the simplicity of the self and non-self structure.

### 5.3. Evaluation of the Proposed Multi-Class Classifier (MPSO-DENSA)

In this section, the proposed multi-class classification algorithm is evaluated. The UC Irvine Machine Learning Repository datasets [33] which are used to perform this

assessment are presented in table 7. Each of these datasets has different number of classes (2-10), features (4-18), and samples, which will cause diversity in testing and better evaluation. The datasets used for comparison are Iris, Diabetes and Cancer, because all the compared methods especially artificial immune-based models used them as test beds. Each dataset is randomly divided into three parts; half of the data are used for training, the remaining data are utilized for validation (objective function calculation) and the final test. The implementation of the proposed multi-class algorithm is repeated 30 times, and also the obtained average results is considered as the final result.

Table 7. Datasets for MPSO-DENSA evaluation.

Dataset	Number of samples	Number of Features	Number of classes
Ecoli	336	7	8
Car evaluation	1728	6	4
Balance	625	4	3
Glass	214	9	6
Iris	150	4	3
Diabetes	150	8	2
Cancer	768	9	2
Seed	210	7	3
Vehicle	846	18	4
Wine	178	13	3
Yeast	1484	8	10
zoo	101	16	7

#### 5.3.1. Parameters Initialization

There are some adjustable parameters in this algorithm, which should be set at the beginning of the algorithm, alike the proposed two-class classifications algorithm. The first group is the parameters of the objective function, which are those weights multiplied in the sub-objectives associated with the detection rate, the false positive rate, and the number of detectors. Their values are  $w_1 = 1$ ,  $w_2 = 1$ , and  $w_3 = 0.1$ , then in the objective function, the importance of maximizing the detection rate and minimization false positive rate is equal to each other, and each one is 10 times more important

than minimizing the number of detectors. The second group is the PSO parameters. In the algorithm implementation process the number of particles 20, the coefficient values  $C_1$  and  $C_2$  two and the inertia weight 0.9 have been considered.

### 5.3.2 Evaluation Results of the MPSO-DENSA Algorithm

In this section, the behaviour of the proposed multi-class method has been investigated on several datasets, and it has shown a fairly good ability in data classification. The performance of this method in terms of the accuracy and the training and testing time is shown in table 8. It is necessary to mention that the training time is not the critical time for the proposed algorithm.

Table 8. The performance of MPSO-DENSA

Dataset	Accuracy (%)	Time (S)	
		Train	Test
Ecoli	78.52	16167	5
Car evaluation	83.22	48500	15
Balance	97.23	13256	14
Glass	89.35	7275	6
Iris	98.38	3637	5
Diabetes	98.24	2425	4
Cancer	80.52	12021	14
Seed	92.41	4546	8
Vehicle	96.35	84875	16
Wine	98.71	3789	4
Yeast	96.45	80833	15
zoo	95.99	4467	1

As it can be seen in table 8, the proposed method has obtained the acceptable results and high accuracy in many datasets, but the low accuracy of this algorithm in some of them such as Cancer, Car evaluation, and Ecoli can be due to non-linear correlation between features and classes as well as noise in data. MPSO-DENSA is challenged in some of dataset with high number of classes and small number of samples, which is observed in Ecoli dataset, because of only utilizing self-class data in training phase. In other situations, the proposed method will not encounter any challenges, due to using the dynamic number of detectors in the identification of samples which are individually in a same class, even with nonlinear

boundaries between classes, because it can assign a detector to each sample. Since MPSO-DENSA uses a one vs all approach, one class is considered as a self-class and other classes as non-self classes each time, therefore the classification results can be shown separately for each class of dataset. The results for some datasets which have three classes are shown in Tables 9 to 12.

Table 9. The performance of MPSO-DENSA for each class separately on Iris dataset.

Class	Detection rate	False positive rate	Number of detectors	Objective function	Accuracy
1	98.21	07.21	12	1.9232	97.99
2	99.98	04.16	9	1.9710	99.01
3	98.88	03.68	11	1.9724	98.23

Table 10. The performance of MPSO-DENSA for each class separately on Balance dataset.

Class	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)
1	97.11	08.15	11	1.9048	96.52
2	95.02	11.02	19	1.8562	95.12
3	99.89	02.36	17	1.9817	99.10

Table 11. The performance of MPSO-DENSA for each class separately on Seed dataset.

Class	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)
1	89.29	06.67	8	1.8429	88.85
2	96.81	14.01	13	1.8529	93.37
3	97.54	08.21	13	1.9072	94.01

Table 12. The performance of MPSO-DENSA for each class separately on Wine dataset.

Class	Detection rate (%)	False positive rate (%)	Number of detectors	Objective function	Accuracy (%)
1	98.11	06.11	10	1.9335	97.74
2	100	01.11	8	2.0015	99.95
3	99.88	00.87	11	1.9993	99.67

According to the results obtained, it can be seen that the proposed multi-class algorithm utilizes the detectors as needed. For example, in the implementation on the Balance dataset, more detectors are used than Iris, due to the more samples in Balance more detectors are required to cover non-self-space. Experimental results have shown

that the MPSO-DENSA algorithm has achieved the sufficient number of detectors using the PSO objective function. Because of the variable radius of the detectors, which has obtained by adaptive parameter adjustment in the proposed algorithm, the minimum number of detectors has reached.

### 5.3.3 Comparison of MPSO-DENSA with Other Methods

In this section, we compare the MPSO-DENSA accuracy with other immune system classifiers and some of the most popular classifiers. Immune system classifiers selected for comparison are M-NSA [15], MINSA [14], ANSC [16], DAIS [15], and NSSAC [20] which are negative selection based and also AIRS [34] which is based on immune network theory. Some other robust classifiers named k-NN [35], C4.5 [36], and Radial Basis Function (RBF) based on neural networks [37] are also used. Table 13 shows the results obtained from MPSO-DENSA compared to other classifiers in terms of accuracy.

The experimental results demonstrate that the proposed multi-class algorithm is competitive with artificial immune-based classifiers as well as other powerful classifiers. There are two important capabilities for achieving the high accuracy in MPSO-DENSA algorithm: using the one vs all approach; which converts the multi-class classification problem into several binary classification problems and finally combining the results. For example, in the implementation on Iris dataset which has three classes of data, the problem is divided into three binary classification problems and are combined the results eventually. This issue is clearly also evident in datasets such as Diabetes and Cancer with two classes show this ability of the algorithm. Therefore, the proposed method can overcome some limitations of negative selection methods like black holes. The second ability of proposed algorithm is automatic and adaptive parameters adjustment using PSO algorithm; as regards in most of the methods compared, the input parameters of the algorithm are determined by trial and error and manually, hence this potential capability can be considered as the reason for high classification accuracy.

Table. 13. Comparison of accuracy percentage between MPSO-DENSA and other classifiers.

Dataset / Method	Iris	Diabetes	Cancer
NSSAC	97.00	97.35	73.40
M-NSA	95.33	96.37	70.44
DAIS	95.80	96.67	73.40
MINSA	96.00	96.51	72.00
ANSC	95.80	96.55	75.60
k-NN	96.00	96.60	67.60
C 4.5	94.00	86.70	67.00
RBF	94.00	94.40	66.36
MPSO-DENSA	98.38	98.24	80.52

## 6. Conclusion

One of the important challenges for negative selection based classification is developing an explicit and deterministic boundary between self and non-self-spaces. A flexible boundary between self and non-self-spaces was defined in the proposed methods inspired by negative selection algorithm based on the distribution estimation (DENSA) and using self-space modelling utilizing Gaussian mixture model. Absence of a specific process for effective parameter initialization such as the detection threshold, the maximum number of detectors, and etc. in each dataset is another limitation of negative selection classification algorithms which prevents achieving adequate classification accuracy and precision. Therefore, adaptive and optimal calculation of these parameters is necessary which has been realized in the proposed methods. New classifiers named PSO-DENSA and MPSO-DENSA were presented in this paper. Combination of the negative selection with the PSO algorithm is used to optimize and adapt the initial parameters and achieve the best results and high accuracy of classification in the proposed algorithms, because the combination of negative selection algorithms with metaheuristic methods increases the strength of classification.

The proposed two-class PSO-DENSA method was evaluated on five synthetic binary datasets, and all comparisons have shown improvement in detection rate, the value of objective function, accuracy and at the same time the significant decrease in the false positive rate and the number of detectors. For example DENSA achieved 91.53% accuracy with 499 detectors on the Pentagram dataset, while PSO-DENSA was successfully reached the accuracy of 93.48% only using 39 detectors.

Our proposed multi-class method was tested on UC Irvine machine learning repository. 12 datasets with different number of samples, features, and classes have been used, for these evaluations. A wide range of algorithms, including artificial immune based methods (especially negative selection) and other classifiers were used in comparisons. According to the experimental results the proposed multi-class method is comparable to other popular multi-class classifiers using suitable and adaptive parameter initialization. Evaluating on Iris, Diabetes and Cancer dataset, it was found that the proposed classifier has acceptable performance in terms of accuracy compared to other classification algorithms.

Finally, it can be concluded that the proposed adaptive classification approaches have been able to achieve excellent results in both two-class and multi-class classification problems using the least possible resources (including the minimum number of detectors) with the ability to automatically adjust the main parameters of negative selection algorithm.

## References

- [1] Kephart, J. O., "A biologically inspired immune system for computers", in *Artificial Life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*, pp. 130-139 (1994).
- [2] Dasgupta, D.; Yu, S.; Nino, F., "Recent advances in artificial immune systems: models and applications", *Applied Soft Computing*, vol. 11, pp. 1574-1587 (2011).
- [3] Timmis, J.; Hone, A.; Stibor, T.; Clark, E., "Theoretical advances in artificial immune systems", *Theoretical Computer Science*, vol. 403, pp. 11-32 (2008).
- [4] Dasgupta, D., "Using immunological principles in anomaly detection", *Proceedings of the Artificial Neural Networks in Engineering (ANNIE'96)*, St. Louis, USA (1996).
- [5] Vigano, S.; Perreau, M.; Pantaleo, G.; Harari, A., "Positive and negative regulation of cellular immune responses in physiologic conditions and diseases", *Clinical and Developmental Immunology*, vol. 2012 (2012).
- [6] Hart, E.; Timmis, J., "Application areas of AIS: The past, the present and the future", *Applied soft computing*, vol. 8, pp. 191-201 (2008).
- [7] Zhu, F.; Chen, W.; Yang, H.; Li, T.; Yang, T.; Zhang, F., "A Quick Negative Selection Algorithm for One-Class Classification in Big Data Era", *Mathematical Problems in Engineering*, vol. 2017 (2017).
- [8] Aydin, I.; Karakose, M.; Akin, E., "Artificial immune classifier with swarm learning", *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 1291-1302 (2010).
- [9] Jantan, H.; Sa'dan, S. A.; Azemi, C.; Syafiqah, N. H., "Academic leadership bio-inspired classification model using negative selection algorithm", the 5<sup>th</sup> International Conference on Computing and Informatics, ICOCI 2015 (2015).
- [10] Bendiab, E.; Kholadi, M. K., "The negative selection algorithm: a supervised learning approach for skin detection and classification", *International Journal of Computer Science and Network Security*, vol. 10, pp. 86-92 (2010).
- [11] Sahare, M.; Gupta, H., "A review of multi-class classification for imbalanced data", *International Journal of Advanced Computer Research*, vol. 2, pp. 160-164 (2012).
- [12] Lee, D. W.; Sim, K.-B., "Negative Selection Algorithm for DNA Sequence Classification", *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 4, pp. 231-235 (2004).
- [13] Markowska-Kaczmarska, U.; Kordas, B., "Negative Selection based method for Multi-Class problem classification", in *Intelligent Systems Design and Applications, 2006. Sixth International Conference on*, 2006, pp. 1165-1170 (2006).
- [14] Igawa, K.; Ohashi, H., "Discrimination-based artificial immune system: Modeling the learning mechanism of self and non-self discrimination for classification", *Journal of Computer Science*, vol. 3, pp. 204-211 (2007).
- [15] Markowska-Kaczmarska, U.; Kordas, B., "Multi-class iteratively refined negative selection classifier", *Applied Soft Computing*, vol. 8, pp. 972-984 (2008).
- [16] Igawa, K.; Ohashi, H., "A negative selection algorithm for classification and reduction of the noise effect," *Applied Soft Computing*, vol. 9, pp. 431-438 (2009).
- [17] Lian, Y.; Yong-kang, X., "A New Classifier for Multi-Class Problems Based on Negative Selection Algorithm", in *Education Technology and Computer Science (ETCS), Second International Workshop on*, 2010, pp. 105-108 (2010).
- [18] Oliveira, L.; Drummond, I., "Real-valued Negative Selection (rms) for classification task", *Recognizing Patterns in Signals, Speech, Images and Videos*, pp. 66-74 (2010).
- [19] Fakhari, S. N. S.; Moghadam, A. M. E., "NSSAC: Negative selection-based self adaptive classifier", in *Innovations in Intelligent Systems and Applications (INISTA), International Symposium on*, 2011, pp. 29-33 (2011).
- [20] Fakhari, S. N. S.; Ziabari, M. T., "A Self Adaptive Algorithm for Classification Based on Negative Selection Technique", *Artificial Intelligence*, vol. 1, (2014).
- [21] Elberfeld, M.; Textor, J., "Negative selection algorithms on strings with efficient training and linear-time classification", *Theoretical Computer Science*, vol. 412, pp. 534-542 (2011).
- [22] Soliman, O. S.; Adly, A., "Quantum-negative selection algorithm for associative classification", in *Granular Computing (GrC), 2012 IEEE International Conference*, pp. 418-423 (2012).
- [23] Zheng, X.; Zhou, Y.; Fang, Y., "The Dual Negative Selection Algorithm Based on Pattern Recognition Receptor Theory and Its Application in Two-class Data Classification", *JCP*, vol. 8, pp. 1951-1959 (2013).
- [24] Lytvynenko, V., "Hybrid swarm negative selection algorithm for dna-microarray data classification", *Bulletin of the National University of Lviv Polytechnic. Computer Science and Information Technology*, pp. 134-148 (2014).
- [25] Mishra, P. K.; Bhusray, M., "Artificial Bee Colony Optimization based Negative Selection Algorithms to Classify Iris Plant Dataset", *International Journal of Computer Applications*, vol. 133, no. 10, pp. 40-43 (2016).



- [26] de Abreu, C. C. E.; Duarte, M. A. Q.; Villarreal, F., "An immunological approach based on the negative selection algorithm for real noise classification in speech signals", *AEU-International Journal of Electronics and Communications*, vol. 72, pp. 125-133 (2017).
- [27] Rashid, N.; Iqbal, J.; Mahmood, F.; Abid, A.; Khan, U. S.; Tiwana, M. I., "Artificial Immune System-Negative Selection Classification Algorithm (NSCA) for Four Class Electroencephalogram (EEG) Signals", *Frontiers in human neuroscience*, vol. 12, pp. 439-453 (2018).
- [28] Bishop, C. M., "Pattern Recognition and Machine Learning": Springer, (2006).
- [29] Kennedy, J., "Particle swarm optimization", in *Encyclopedia of machine learning*, ed: Springer, pp. 760-766 (2011).
- [30] Fouladvand, S.; Osareh, A.; Shadgar, B., "Distribution estimation based negative selection algorithm (DENSA)", in *Artificial Immune Systems (AIS), 2015 International Workshop*, pp. 1-7 (2015).
- [31] Fouladvand, S.; Osareh, A.; Shadgar, B.; Pavone, M.; Sharafi, S., "DENSA: An effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors", *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 359-372 (2017).
- [32] Lasisi, A.; Ghazali, R.; Herawan, T., "Comparative performance analysis of negative selection algorithm with immune and classification algorithms", in *Recent Advances on Soft Computing and Data Mining*, ed: Springer, pp. 441-452 (2014).
- [33] Merz, C. J.; Murphy, P. M., "{UCI} Repository of machine learning databases", (1998).
- [34] Watkins, A.; Timmis, J.; Boggess, L., "Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm" *Genetic Programming and Evolvable Machines*, vol. 5, pp. 291-317 (2004).
- [35] Altman, N. S., "An introduction to kernel and nearest-neighbor nonparametric regression", *The American Statistician*, vol. 46, pp. 175-185 (1992).
- [36] Quinlan, J. R., *C4. 5: programs for machine learning*: Elsevier, (2014).
- [37] Lowe, D., "Adaptive radial basis function nonlinearities, and the problem of generalisation", in *1989 First IEE International Conference on Artificial Neural Networks*, (Conf. Publ. No. 313), pp. 171-175 (1989).