

1682. An intelligent fault diagnosis method using variable weight artificial immune recognizers (V-AIR)

Hongli Zhang¹, Jicheng Liu², Erping Zhou³, Dong Li⁴, Bo Wang⁵, Kunju Shi⁶

^{1,2,4,5,6}School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200072, China

¹Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai 20072, China

³University of Bolton, Bolton, UK

²Corresponding author

E-mail: ¹zhang40941@126.com, ²liujichenghit@163.com, ³e.zhou@bolton.ac.uk, ⁴monkey_ld@163.com, ⁵nianqingzaixian@sohu.com, ⁶shikunjv@sina.cn

(Received 4 December 2014; received in revised form 20 April 2015; accepted 5 May 2015)

Abstract. The Artificial Immune Recognition System (AIRS), which has been proved to be a successful classification method in the field of Artificial Immune Systems, has been used in many classification problems and gained good classification effect. However, the network inhibition mechanisms used in these methods are based on the threshold inhibition and the cells with low affinity will be deleted directly from the network, which will misrepresent the key features of the data set for not considering the density information within the data. In this paper, we utilize the concept of data potential field and propose a new weight optimizing network inhibition algorithm called variable weight artificial immune recognizer (V-AIR) where we replace the network inhibiting mechanism based on affinity with the inhibiting mechanism based on weight optimizing. The concept of data potential field was also used to describe the data distribution around training samples and the pattern of a training data belongs to the class with the largest potential field. At last, we used this algorithm to rolling bearing analog fault diagnosis and reciprocating compressor valves fault diagnosis, which get a good classification effect.

Keywords: V-AIR, data potential field, weight optimizing, network inhibition, combining mutation.

1. Introduction

Immune system has been noticed by many computer sciences and engineering researchers, for it has many advantages that many artificial-intelligence systems can't match such as the distinction of self-nonsel self molecules, early warning of danger signals, distributed disposition of hazardous situations and incremental learning. Thus, the immune system characteristics has been developed and applied in many engineering field [1]. In the last decade, several interdisciplinary research scientists have produced a prolific amount of immune inspired algorithms, mathematical models and hybrid intelligent systems by extracting one or more properties, function and mechanism of immune system.

Based on the principles of self-nonsel self discrimination of immune system, one of the major algorithms developed is the Negative Selection Algorithm (NSA) which is analogous to the censoring process of T cell maturation in immune system [2, 3]. The NSA generates detectors randomly and eliminates the ones that detect self so that the remaining detectors can detect any non-self [4]. Since the matching criterion is not specific, we can cover the non-self space with limited number of detectors. However, the original NSA represents detectors in binary, which may not capture the structure of some problem spaces or cause a low level of computing efficiency [5]. The real-valued negative selection (RNS) extends the previous binary anomaly detectors to real space anomaly detectors [5, 6] and a lot of further researches concentrating on detector generation schemes also have sprung up such as the hyper-rectangular detectors [5], the variable-sized detectors (V-detector) [7, 8], the hyper-ellipsoid detector [9]. Nevertheless, almost all of the methods don't consider the distribution density of self samples and use constant-size self hyperspheres to express the profile of self space, which may increase false alarm rates with a

smaller self radius and decreases detection rates on the contrary. The ANSA disposes the issue with variable-sized self radius and acquires a great of achievement [10].

Another method of imitating the immune system is to use the specific immunity of B cells. Compared with NSA, it is a straightforward artificial immune algorithm and called as “positive selection algorithm”. Jerne [11] and Farmer [12] firstly proposed the immune network theory to illustrate the B cell concentration changes. Inspired by the immune network theory, a clustering algorithms-ainet model was firstly invented to find some memory points called as memory antibodies representing the data distribution. As an information compression algorithm, the algorithm can largely reduce the number of memory cells, however, it also may misrepresent the key features of the data for not considering the density information within the data [13, 14]. So the Adaptive Radius Immune Algorithm (ARIA) [15] introduces an antibody adaptive suppression radius that varies inversely with the local density for each antibody’s neighborhood to preserve the density information of the data set and obtains a superior performance.

Artificial immune network model (AINE) is another algorithm obtained by the proposing of B cell stimulation and network affinity threshold [16]. The most successful unsupervised learning algorithm based on AINE is the resource limited artificial immune system (RLAIS) which not only uses the immune network dynamics and introduces the concept of shape-space but also puts forward the concept of artificial recognition ball (ARB) [17]. Influenced by the concept of ARB in RLAIS, Watkins et al. [18] put forward a supervised learning algorithm named as Artificial Immune Recognition System (AIRS) using the immune mechanism of high frequency variation, clonal mutation, resource allocation mechanism and network suppression. Since then, AIRS is applied to a serial of classification problems and a lot of improved algorithms continuously emerge. A hybrid classifier that uses the fuzzy weighted pre-processing to weight the input data and the AIRS classifier shows an effective performance on several problems such as machine learning benchmark problems and medical classification problems [19, 20]. In the study of Polat et al. [21], a new resource allocation mechanism was done with fuzzy-logic in the Fuzzy-AIRS. As a further research, another nonlinear recognition system involving AIS and ANN (artificial neural network (ANN)-aided AIS-response, AaA-response) was proposed, which used multiple antibodies to form an output for a presented input [22]. In the research of MAIRS2, a modified AIRS2 was used to replace the k -nearest neighbor algorithm with the fuzzy k -nearest neighbor to improve the diagnostic accuracy of diabetes diseases. However, the network inhibition mechanisms used in these methods are all based on threshold inhibition and the cells with low affinity will be deleted directly from the network, so as an information compression algorithm it may misrepresent the key features of the data set for the relative distances among the trained memory cells did not adequately represent the original data [13-15].

In this paper, we utilize the concept of data potential field and propose a new classification method called variable weight artificial immune recognizers (V-AIR). In this algorithm, we use weight optimizing inhibition mechanism to realize the network inhibition mechanism, which is realized by quadratic programming. In this way, for high density portion of training samples, memory cells are allowed to have larger weights; for sparse region of samples, the memory cells are allowed to have smaller weights. Therefore, this method reflects the density distribution information of data through increasing the memory cell’s weight in dense region and decreasing the weight of memory cells in sparse region. In testing phase, the pattern of a training data belongs to the class with the largest potential field.

This paper is structured as follows: In the next section, we give our motivation to the definition of data potential field. In Section 3, we introduce the V-AIR to solve the problem of data classification. Related works are also discussed in Section 3. In Section 4, 4 kinds UCI data sets are used to compare with AIRS. Beside, rolling bearing fault diagnosis is also researched to prove the effectiveness of V-AIR.

2. Definition of data potential field

In 1837, British physicist Faraday firstly proposed the concept of physical to describe the non-contact effect among objects. He thought the occurrence of non-contact interaction such as the universal gravitation, electrostatic force and magnetic force must be achieved with some intermediate medium and the medium is field. With the development of field theory, people abstracted this as a mathematical concept [23]:

Definition 1: If every point in the space corresponds to a certain physical or mathematical value, a field can be determined in the space and this field can be expressed with different univalent functions according to different objects such as:

- The gravitational field:

$$\varphi(r) = G \times \sum_{i=1}^n \frac{m_i}{\|r - r_i\|} \tag{1}$$

with m_i is the weight of a particle, $\|r - r_i\|$ is distance between site r and particle r_i ;

- The nuclear field:

$$\varphi(r) = V_0 \times e^{-\left(\frac{\|r-r_i\|}{R}\right)^2}, \tag{2}$$

with V_i is the weight of a nucleus, $\|r - r_i\|$ is the distance between site r and nucleus r_i , R is the scope of nuclear force.

Referencing the thought of field in physics, Li et al. [23] also introduced the concept of potential field into the number field space and created the data potential field:

Definition 2: Given a data set with N objects $D = \{x_1, x_2, \dots, x_n\}$ in space $\Omega \subseteq R^p$. A field can be determined in the space and this field can be expressed with different univalent functions according to different objects such as:

$$\varphi(x) = \sum_{i=1}^N \varphi_i(x) = \sum_{i=1}^N m_i \times e^{-\left(\frac{\|x-x_i\|}{\delta}\right)^2}, \tag{3}$$

with $\|x - x_i\|$ is the distance between site x and nucleus x_i , m_i is the weight of nucleus x_i , δ is the impact factor of nuclear field function. Assuming that the normalization condition meet, we have:

$$\sum_{i=1}^n m_i = 1. \tag{4}$$

Similar to the contour in physics, we also represent the characteristics of physical field distribution with the potential function.

3. The variable weight artificial immune recognizers (V-AIR)

In this research, an innovative classification model called variable weight artificial immune recognizers (V-AIR) is proposed. The motivation for developing this algorithm comes from the fact that deleting an antibody from the immune system directly may misrepresent the key features of data set for not considering the density information within the data. In this paper, we utilize the concept of potential field data and propose a new weight optimizing network inhibition algorithm called variable weight artificial immune recognizers (V-AIR) where we replace the network

inhibiting mechanism based on affinity with the inhibiting mechanism based on weight optimizing. In this algorithm, the weights of memory cells are optimized by the quadratic programming and the cells with weights below the setting threshold will be deleted directly from the memory network. In this way, for high density portion of training samples, memory cells are allowed to have a larger weight; for sparse region of samples, memory cells are allowed to have a smaller weight. Therefore, this method reflects the density distribution information of data through increasing the memory cell's weight in dense region and decreasing the weight of memory cells in sparse region. To describe conveniently, V-AIR also introduces the concept of Artificial Recognition Ball (ARB) and variable weight memory cells (*V-mc*). Like the memory cells used in RLAIIS and AIRS [17, 18], a *V-mc* also represents a number of identical B cells but has a variable weight changing according to the result of optimizing.

3.1. The algorithm description

The V-AIR can be described specifically as follows:

AG_h : the antigen set of h th class, $AG_h \in R^{M \times L}$. L is the dimension of attributive character, M is the number of antigens belonging to a class.

$MC-P_h$: the memory cell pool used for the h class antigens, which represents the collection of memory cells, $MC \in R^{N \times L}$. L is the dimension of attributive character, N is the number of memory cells.

Ag_{hi} : the i th antigen belonging to h class, $Ag_{hi} \in AG_h$.

$ARB-P_h$: the artificial recognition ball pool used for h th antigens. It is used for storing the cells produced in the process of cloning.

ARB_{hj} : stored in $ARB-P_h$ and corresponds to the j th ARB of class h , $ARB_{hj} \in ARB-P_h$.

$C-mc_{hi}$: the i th candidate memory cells stored in $MC-P_h$, $C-mc_{hi} \in MC-P_h$.

$V-mc_{hi}$: the retained memory cells after quadratic programming, which is also stored in $MC-P_h$ with a variable weight.

Af : the measure of similarity degree or affinity between antigen and antibody cells, the value in V-AIR is expressed with kernel function, such as:

$$Af(V - ARB_{hj}, Ag_{hi}) = e^{-\left(\frac{\|V - ARB_{hj}(j,:) - Ag_{hi}(i,:)\|^2}{\delta}\right)}, \quad (5)$$

where Re_h : the $V-ARB_h$'s total resources, N_{hj} : the clone scales of ARB_{hj} , T : the cloning coefficient of ARB_h , M_r : the mutation rate of ARB_h , s_h : the average affinity threshold between antigen and ARB, C_h : the number of class h 's ARB before resource allocation, C_h^* : the number of ARB_h 's after resource allocation, R_s : the total number of resource that all ARB_h are allowed to have, δ : the impact factor of nuclear field function, r : the variable coefficient to adjust the merging criteria between antibodies, D : the number of antigen categories, E_t : the setting threshold to delete or reserve an *V-mc*. If the weight of an *V-mc* is greater than E_t , the *V-mc* will be reserved. If the weight of an *V-mc* is smaller than E_t , the *V-mc* will be deleted.

During the training process, all kinds of antigens are independent of each other, so we can freely take the antigens of class h as the instance to illustrate the generative process of *V-mc*. The following of this part will instruct the steps of the algorithm in detail:

1. Initialization: In this part, we firstly normalize the training samples in the unit square $[0, 1]^n$. At the same time, the required input: AG_h , s_h , R_s , δ , r , T , E_t , k are also set. The initial ARB_h 's in $ARB-P_h$ are generated randomly on this step.

2. The produce of candidate memory cells: For every Ag_{hi} ($i = 1, 2, \dots, M$), do the following steps:

a) Calculate the affinity Af between Ag_{hi} and the initialized ARB_h 's.

b) Clone and mutate ARB_{hj} with $N_{hj} = T^* Af(ARB_{hj}, Ag_{hi})$ and mutation rate M_r . Then get

the number of ARB_{h_s} , which are labeled as C_h .

c) Calculate the total ARB_{h_s} ' resources $R_{eh} = \sum_{j=1}^{C_h} N_{hj}$ and distribute the resource in ARB_{h_s} , until the $R_{eh} \leq R_s$. This will result in some ARB_{h_s} with resource approaching 0 to die and ultimately control the population. The specific process of resource allocation is shown in Fig. 1.

d) Calculate the average affinity Af between ARB_{h_s} and Ag_{hi} , if the average affinity Af larger than s_h , select the candidate memory cells ($C-mc_{h_s}$); if the average affinity Af less than s_h , return to step (b) until meeting the affinity conditions.

e) Choose the ARB_{h_j} with highest affinity and take it as a $C-mc_{hi}$. The $C-mc_{hi}$ will be put into $MC-P_h$.

The step (a)-(e) will be looped until all of the class h antigens are processed.

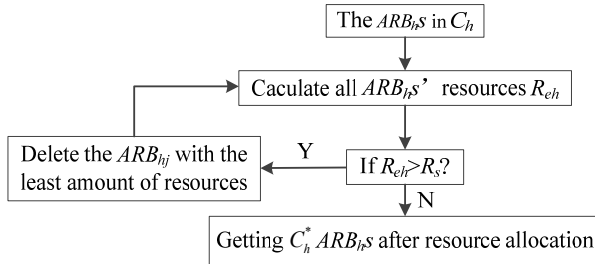


Fig. 1. The process of resource allocation

3. The network inhibition of candidate memory cells: The weights of $C-mc_{h_s}$ in $MC-P_h$ (or the concentration of each $C-mc_{hi}$) are optimized by quadratic programming function and the $C-mc_{h_s}$ with weight below the setting threshold E_r will be deleted directly from the memory network. At last, the retained $C-mc_{h_s}$ change into $V-mc_{h_s}$ and constitute the $MC-P_h$.

After all the D types of training antigens are recognized, the $V-mc_{h_s}$ will be used for data classification. The classification process is conducted by the principle of potential field dominant namely each kind of $V-mc_{h_s}$ ' potential field is superposed at the point of the test sample Y_k , which can get the potential field value of Y_k belonging to each kind of $V-mc_h$: $\varphi_1, \varphi_2, \dots, \varphi_D$ and Y_k belongs to the class with the largest superimposed potential field φ_i .

As a supervised learning classification system, each antigen Ag_{hi} has its class and produces many ARB_{h_s} responding to this antigen Ag_{hi} . After clone mutation and resource allocation process, the ARB_{h_s} with resource equal to 0 will be deleted from the $ARB-P_h$ directly. The ARB_{h_s} with the strongest response will be seen as the $C-mc_{h_s}$ and put into the $MC-P_h$. However, the number of these $C-mc_{h_s}$ still exceeds the people's cognitive needs, so we need to use the quadratic programming function to estimate the weights of objects and get a few non-zero core objects to reduce the number of memory cells. At last, the several core objects are used to represent the data classification model. The quadratic simplified estimation process is as follows: In $\Omega \in R^d$, the set of $H = \{C - mc_{h1}, C - mc_{h2}, \dots, C - mc_{hE}\}$ includes E candidate memory cells. Besides, in order to describe conveniently, we describe them as $H = \{X_1, X_2, \dots, X_E\}$, so according to the concept of data potential field, the data potential field of any test sample in the space can be decided as:

$$\varphi(Y) = \sum_{i=1}^E \varphi_i(Y) = \sum_{i=1}^E \left(m_i \times e^{-\left(\frac{\|Y-X_i\|}{\delta}\right)^2} \right). \tag{6}$$

When the weights of objects don't require equal, the weight of m_1, m_2, \dots, m_E can be seen as a group of function according to spatial position X_1, X_2, \dots, X_E . If the population distribution is known, the weights of all objects can be estimated through minimizing the error criterion between potential function $\varphi(Y)$ and distribution density function. Assuming that the overall density of all

objects is $p(Y)$, so when the value of δ is certain, we can minimize the following integral square error criterion [24]:

$$\min J = \min_{\{m_i\}} \int_{\Omega} \left[\frac{\varphi(Y)}{(\sqrt{\pi}\sigma)^d} - p(Y) \right]^2 dx. \quad (7)$$

Accessibility:

$$\min J = \min_{\{m_i\}} \int_{\Omega} \left[\frac{\varphi^2(Y)}{(\sqrt{\pi}\sigma)^{2d}} - \frac{2p(Y) \cdot \varphi(Y)}{(\sqrt{\pi}\sigma)^d} + p^2(Y) \right] dx. \quad (8)$$

Obviously, $\int_{\Omega} p^2(Y) dx$ has nothing to do with m_1, m_2, \dots, m_E , so the objective function can be simplified as:

$$\min J = \min_{\{m_i\}} \left[\int_{\Omega} \frac{\varphi^2(Y)}{2(\sqrt{\pi}\sigma)^d} dx - \int_{\Omega} p(Y)\varphi(Y) dx \right]. \quad (9)$$

Analysis the following function, $\int_{\Omega} p(Y)\varphi(Y) dx$ is the mathematical expectation of $\varphi(Y)$ and we can use E independent extraction samples to approximate:

$$\min J = \min_{\{m_i\}} \left[\frac{1}{2(\sqrt{\pi}\sigma)^d} \int_{\Omega} \varphi^2(Y) dx - \frac{1}{E} \sum_j^E \varphi(X_j) \right], \quad (10)$$

with Eq. (5), can get:

$$\min J = \min_{\{m_i\}} \left[\frac{1}{2(\sqrt{2})^d} \sum_{i=1}^E \sum_{j=1}^E m_i \times m_j \times e^{-\left(\frac{\|Y_i - Y_j\|}{\sqrt{2}\delta}\right)^2} - \frac{1}{E} \sum_{i=1}^E \sum_{j=1}^E m_i \times e^{-\left(\frac{\|Y_i - Y_j\|}{\delta}\right)^2} \right]. \quad (11)$$

Obviously, this is a typical constrained quadratic programming problem and satisfies the following constraint conditions:

$$\sum_{i=1}^E m_i = 1, \quad (12)$$

with $m_i \geq 0$.

Optimize the Eq. (11) and get a set of optimal weights. The optimization result is a few objects in the candidate memory cell concentration areas having great weights and most objects with much smaller weights or a weight of zero.

4. Combining mutation: After each presentation of the training antigen, the $V-mc_{hi}$ s are generated with the variable weights calculated by Eq. (11). However, these cells still have large coincidence with each other for each training antigen being assigned a $V-mc_{hi}$, which increases the computational complexity. Thus, the redundant $V-mc_{hi}$ must be cleared away. However, if an antibody is deleted from the memory antibody network directly, the rest memory antibody would be far away from the antigen and cause the relative distances distortion phenomenon between memory antibodies. So, in this paper we decrease the coincidence and reduce the number of them through merge the similar two memory cells into a new one. We call this merger as combining

mutation for the position and weight of the memory cells changed.

When the distance between two memory cells satisfies the inequality, the two memory cells can be merged into a new one:

$$\|Y_i - Y_j\| \leq r \times \delta \times (M_i + M_j), \tag{13}$$

where r is a variable coefficient to adjust the merging criteria between the antibodies, Y_i and Y_j are the spatial position of the original two memory cells. Here we use the momentum conservation law [23] as the merge mutation rule and the fusion memory cell's new position and weight can be calculated by the following equation:

$$Y_{new} = \frac{M_i \times Y_i + M_j \times Y_j}{M_i + M_j}, \quad (i \neq j), \tag{14}$$

$$M_{new} = M_i + M_j \quad (i \neq j), \tag{15}$$

where Y_{new} is the new spatial position of fusion memory cell, M_i and M_j are the weights of memory cells to be merged, Y_i and Y_j are the spatial position of the original two memory cells and M_{new} is the new weight of fusion memory cell. The merging mutation is recursively implemented, until all the memory cells that meet the merging condition are merged. Through multiple combining mutations, some fusion memory cell would claim a larger and larger weight calculated by Eq. (15). From Eq. (14), we also find that the fusion memory cell would get more close to the center of larger weight memory cell and the larger weight the memory cell have, the closer the fusion memory cell will be. Thus, through many times combining mutation, some fusion memory cells would have much larger weights than other memory cells needed to be merged. In the extreme cases, if the other memory cell's weight is too small to be ignored, the weight and spatial position of the fusion memory cell will have no change in the combining mutation.

5. Recognition phase: After training has completed, each $V-mc_{hi}$ in MC is endowed with a weight. In view of the unbalance factor of the training samples, a $V-mc_{hi}$'s weight should be divided by the number of the training antigens of its class j , so the weight of $V-mc_{hi}$ is changed into:

$$h_i = \frac{m_i}{N_j}, \tag{16}$$

where h_i is the weight of $V-mc_{hi}$ after transformation, m_i is the weight of $V-mc_{hi}$ before transformation and N_j is the number of training samples of class j . The affinity between a test antigen and a $V-mc_{hi}$ changes from Eq. (3) into:

$$\text{affinity}(Y, X_t) = h_i \times e^{-\left(\frac{\|Y-X_t\|}{\delta}\right)^2}. \tag{17}$$

In the next, the recognition phase is performed in a new nearest neighbor approach: k - weight nearest neighbor approach. Like the k -nearest neighbor approach, this method firstly presents each test sample to all of the memory cells and then picks out the k memory cells with the largest affinities in MC . However, the system's classification of a data item is not determined directly by using a majority vote of the outputs of the k most nearest memory cells. In our k -weight nearest neighbor approach, the minimum affinity memory cell is endowed with a vote of 1 and the votes for the k most nearest memory cells are decided by the following equation:

$$V_i = \frac{\text{affinity}(Y, X_i)}{\min(\text{affinity}(Y, X_1), \text{affinity}(Y, X_2), \dots, \text{affinity}(Y, X_k))}, \tag{18}$$

where V_i is the vote for test antigen Y . So the memory cell's vote for a test antigen is converted into decimals and the classification of the test antigen belongs to the class with the largest total value of votes, e.g. in the 3-weight nearest neighbor approach, for a test antigen if there are two nearest memory cells in class 1 with the votes $V_1 = 1, V_2 = 1.1$ and one nearest memory cell in class 2 with the vote $V_3 = 2.2$, then the test sample belongs to the class 2 not the class 1.

3.2. The flow of V-AIR algorithm

V-AIR uses the quadratic to imitate the network inhibition mechanism among $V-mcs$ and distribute the number of B cells included in a $V-mc$. Like the the method used in AIRS, the antigens used in V-AIR are firstly initialized as eigenvectors $Ag_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and proposed to the system during training and learning process. The B cells are also expressed as eigenvectors $Ab_i = (y_{i1}, y_{i2}, \dots, y_{in})$ with x_{ik} and y_{ik} ($k = 1, 2, \dots, n$) expressing the i th attributive character of Ag_i or Ab_j respectively. To reduce the repetition of B cells, the same B cells are expressed as a ARB and the memory cell are expressed as a $V-mc$. The key of the algorithm is to use the mechanism of quadratic programming optimizing the weight of $V-mc$ and use the data field to classify the testing data.

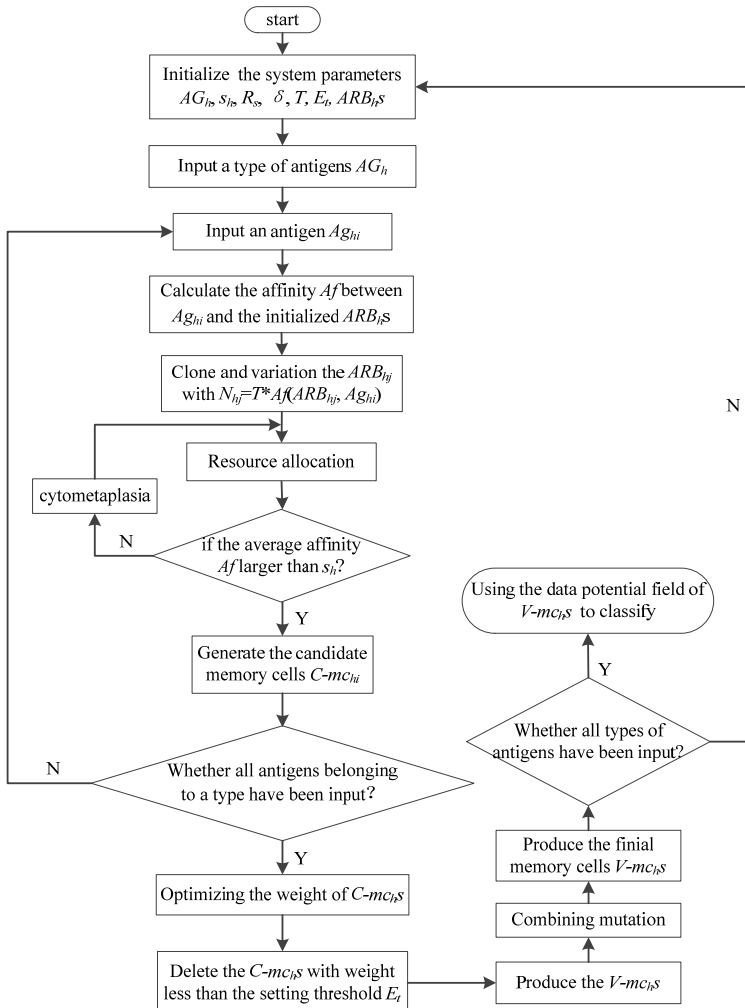


Fig. 2. The flow diagram of V-AIR

In general, the V-AIR can be divided into two stages: the limited resource competition stage and the quadratic programming phase of $C\text{-}mc_h\text{s}$. The limited resource competition stage mainly uses the clone variation and limited resources allocation mechanism to imitate the somatic hyper mutation, clone suppression and immune homeostasis process in human immune system. In the human immune system, there is also network suppression among B cells: when the external antigens invade, some B cells in the immune system will clone and amplify and its concentration would increase; however, some other B cells will die and its concentration would decrease. In V-AIR, we use the quadratic programming to achieve this function. Fig. 2 gives the description of this algorithm.

4. Application of V-AIR to classification

This section briefly touched on the effect of parameters and some comparisons between the existing techniques presented in [18]. The focus of parameters' discussion was mainly on two of the important features of V-AIR algorithms: classification accuracy and memory cell reduction. In the comparisons, experiments were carried out using four UCI datasets [25]: the Fisher Iris, Pima diabetes, Ionosphere and Sonar data sets.

4.1. Classification accuracy

In this study, the classification accuracies for the datasets were measured according to Eq. (19) [20, 21]:

$$\text{accuracy}(T) = \frac{\sum_{j=1}^T \text{assess}(Y_j)}{T}, \quad Y_i \in T, \tag{19}$$

$$\text{assess}(Y) = \begin{cases} 1, & \text{if } \text{classify}(Y) = Y.c, \\ 0, & \text{otherwise,} \end{cases}$$

where T is the set of data items to be classified (the test set), $Y.c$ is the class of the item Y , and $\text{classify } Y$ returns the classification of Y by V-AIR or AIRS.

4.2. Effect of parameters

V-AIR is a multi-parameter classification system, so it is important to evaluate that the behavior of V-AIR is altered with respect to the user defined parameters in actual engineering. In order to establish this, investigations were undertaken to determine what affect two of the most important features: the number of memory cells and the classification accuracy. For V-AIR has 8 parameters to analyze, so the effect of each parameter is analyzed to determine which one is more likely to affect the above two items.

In the V-AIR classification method, once a set of memory cells has been developed, the resultant cells can be used for classification. As described in 3.1, obviously we can conclude that the mutation rate M_r , clonal rate T and number of resource R_s allowed in V-AIR only associate with the evolution speed of ARB_h but have nothing to do with the classification accuracy. Thus, in the analysis of classification accuracy and number of memory cells, we need not to consider the 3 parameters. The average stimulation threshold s_h is an evaluation parameter to determine the degree of affinity between antigens and antibodies. In the process of clonal variation, if s_h is set too large, ARB_h would spend more time to cycle the clone and mutation process until the termination condition satisfied; if s_h is set too small, ARB_h would have small affinity with the target antigen Ag_{hi} which can affect the classification accuracy in some degree. Therefore, s_h should be set in a suitable range. According to Eq. (5), s_h is a parameter changing in $(0, 1]$, so we give the instructional scope between 0.6 and 0.94. The setting threshold E_t , which also changes between 0 and 1, determines the number of reserved $C\text{-}mc_h\text{s}$ and can affect the number of memory

cells and classification accuracy in some degree. Because when E_t is set large, less $C-mc_h$ s are used for generating $V-mc_h$ s. Meanwhile the classification accuracy and number of memory cells are both reduced in some degree. When E_t is set small, more $C-mc_h$ s are used for generating $V-mc_h$ s which increase the number of memory cells. The classification accuracy would also be affected in some extent for more $C-mc_h$ s are reserved to produce the $V-mc_h$ s. However, according to Eq. (5), when E_t is little enough, the reserved $V-mc_h$ with small weight would have little influence to final data classification for the A_f calculated by Eq. (5) can be ignored. Hence, we give the guidance value of $E_t = 0.001$ at here. The k value of V-AIR determines the number of nearest neighbor used for classification and has nothing to do with the number of memory cells but relates with the classification accuracy. In the process of application, we give the guidance value between 3 to 7 if the number of $V-mc_h$ s greater than 3 or use all of the $V-mc_h$ s when the number of $V-mc_h$ s below 3.

In our method, the number of memory cells and the detection rate are mainly affected by the parameters r and δ . In order to illustrate the varying rules of them, investigations are firstly undertaken with the diabetes, ionosphere, sonar and iris dataset to determine how do the changes of δ and r affect the number of memory cells (Fig. 3 and 5) and the classification accuracy (Figs. 4 and 6).

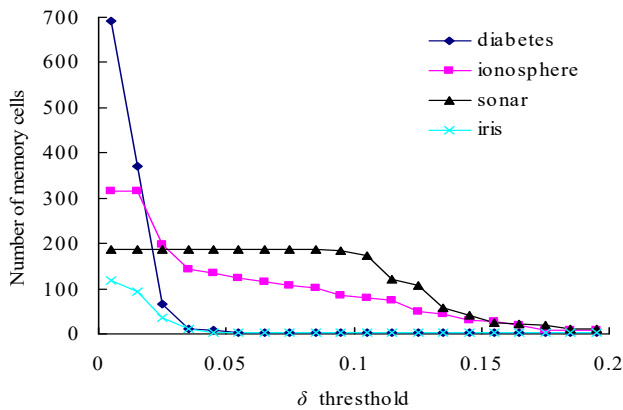


Fig. 3. The change of memory cells with changing values of the δ threshold

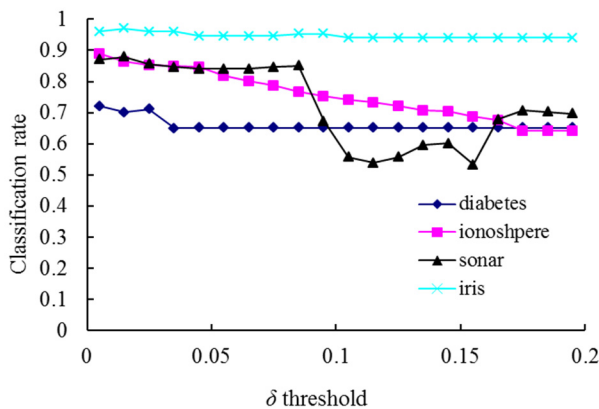


Fig. 4. The change of classification rate with changing values of the δ threshold

The relationships between the parameter δ and the number of memory cells in V-AIR for the four UCI datasets (Diabetes, Ionosphere, Sonar and Iris dataset) are shown in Fig. 3. The variation tendencies of the classification accuracy following with the parameter δ are also shown in Fig. 4. In Fig. 3, the number of memory cells decreases gradually as the increase of the parameter δ , for

more memory cells satisfies the Eq. (5). At last, all of the memory cells belonging to one training class will be combined into one memory cell and the curves for memory cells' number don't change again. During the change of the memory cells' number, the classification accuracy of memory cells also fluctuate significantly but at last, the classification accuracy will becomes calm and steady ultimately, for the number of memory cells never changes again. In Figs. 5-6, altering the parameter r with a fixed parameter δ for each type of datasets, the number of memory cells and the classification accuracy show the same varying patterns with Figs. 3-4.

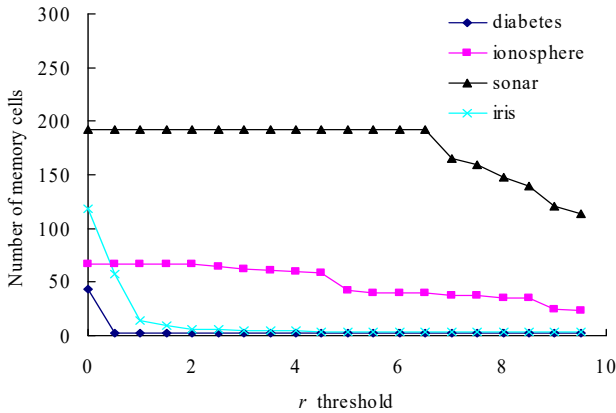


Fig. 5. The change of memory cells with changing values of the r threshold

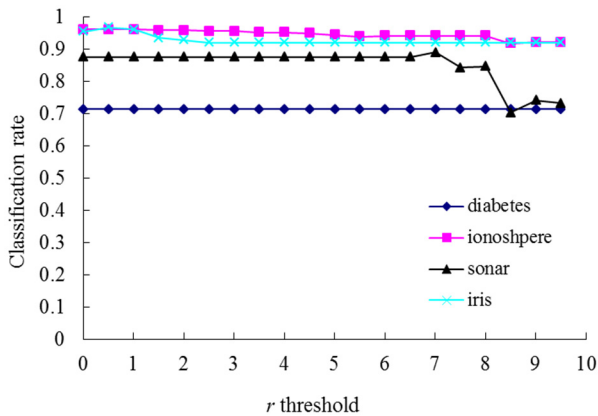


Fig. 6. The change of classification rate with changing values of the r threshold

4.3. Comparison of V-AIR with AIRS

To have a stark comparison, we show the performance of V-AIR on these data sets with the results of AIRS [18] in Tables 1-2. The parameters used in V-AIR are set in Table 3. Since AIRS and V-AIR use all the memory cells to classify the unknown data, the classification cost is proportional to the number of cells. Like the method used in [18], these results are also obtained from the averaging multiple runs of V-AIR, typically consisting of three or more runs and five-way, or greater, cross validation. For the Iris data set, a five-fold cross validation scheme is employed with each result representing an average of three runs across these five divisions. For the Ionosphere data set, we remain the division method as detailed in AIRS: 200 instances here are carefully split almost 50 % positive and 50 % negative are used for training with the remaining 151 as test instances, consisting of 125 “good” and only 26 “bad” instances. The results reported here also represent an average of three runs. For the Diabetes data set a ten-fold cross validation scheme is used, again with each of the 10 testing sets being disjoint from the others and the results

are averaged over three runs across these data sets. The Sonar data set is divided randomly into 13 disjoint sets with 16 cases in each. 12 of these sets are used as training data with 1 as the test date.

The accuracy of V-AIR for Iris, Diabetes, Ionosphere and Sonar is about 0.6 %, 0.3 %, 0.7 % and 1.9 % higher than the AIRS's in Table 1. The accuracy of V-AIR for these four UCI data sets is sufficiently comparable with that of the AIRS classifier but this method requires more memory cells in the classification of Diabetes, Ionosphere and Sonar (Table 2), which means higher computational complexity. However, in the classification of Iris, the V-AIR uses only average of 23.8 memory cells and implements the 97.3 % classification accuracy. Thus, to further validate the advantage of our method, we use it in bearing analog failure with the data set of Case Western Reserve University.

Table 1. Comparisons of accuracy between AIRS and V-AIR

Training set	V-AIR: accuracy (%)	AIRS: accuracy (%)
Iris	97.3 (0.1)	96.7 (3.1)
Diabetes	74.5 (0.7)	74.2 (4.4)
Ionosphere	96.3 (0.1)	95.6 (1.7)
Sonar	86.8 (0.1)	84.9 (9.1)

Table 2. Comparisons of the average number of memory cells between V-AIR and AIRS. Size refers to the number of training data and the percentage refers to the compression ratio of memory cells

Training set	Size	V-AIR: memory cells	AIRS: memory cells
Iris	120	23.8/81 % (5.6)	30.9/74 % (4.1)
Diabetes	691	691/0 % (0.0)	273.4/60 % (20.0)
Ionosphere	200	169.6/15 % (5.5)	96.3/52 % (5.5)
Sonar	192	189.7/1 % (0.8)	177.7/7 % (4.5)

Table 3. Used parameters in V-AIR for UCI datasets

Used parameters	Diabetes	Ionosphere	Sonar	Iris
Mutation rate	0.7	0.7	0.7	0.7
Average stimulation threshold	0.9	0.9	0.9	0.9
Clonal rate	100	100	100	100
Number of resources allowed in V-AIR	40	40	40	40
<i>k</i> value for <i>k</i> -weight nearest neighbor	7	7	3	7
<i>r</i> value	0.705	50	7	0.4
δ value	0.001	0.0015	0.025	0.02
E_t value	0.001	0.001	0.001	0.001

5. Application of V-AIR in equipment fault diagnosis

The fault of many equipment usually has the characteristics of complexity and nonlinear. In addition, they also often have small signal noise ratio (SNR) with the weak fault. As a fault simulation instance, the ball bearing data set of Case Western Reserve University [25] was firstly introduced to do the bearing fault diagnosis experiment in this section. Then, as an industrial case, the V-AIR was conducted using the gas valves failure data for piston compressors to validate the feasibility of our failure diagnosis method with the appropriate parameters. The influence of parameters' change on the classification rate and number of memory cells is also discussed in this section.

5.1. The diagnosis of bearing analog failure

In this paper, three kinds of fault samples are used for verifying the effectiveness of the proposed method. Fig. 7 gives the time domain vibration waveform of rolling bearing in normal state, 0.007" lesions size of inner race, outer race and roller fault with the bearing type and parameters referring to Table 4. From Fig. 7, the vibration signal of inner race and outer race fault

have obvious cycle shock, however, the normal and ball fault vibration signal are not obvious and show serious noise interference.

To form the training and testing samples, the normal and various fault time-domains signal data are decomposed through orthogonal wavelet and high frequency of wavelet energy feature extraction [26] then form many $7d$ energy eigenvectors. Respectively, 100 normal and fault eigenvectors are used for training samples. The other 100 normal and fault eigenvectors are used for testing samples to detect the classification accuracy of V-AIR. The calculation parameters used in V-AIR are shown in Table 5.

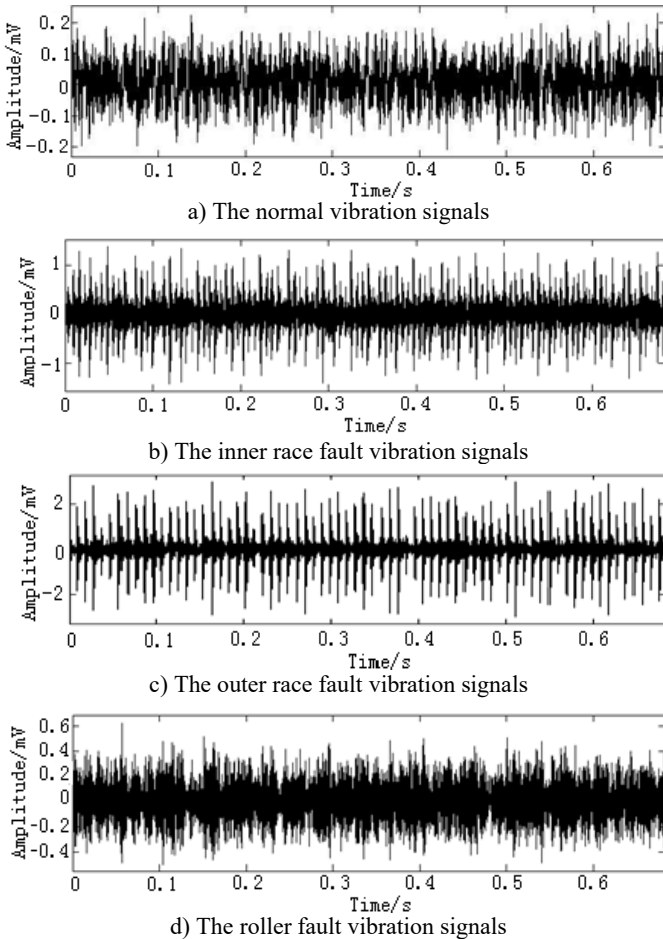


Fig. 7. The time domain vibration waveform of rolling bearing

Table 4. The bearing data classification accuracy of V-AIR

Type of samples	The classification accuracy of V-AIR (%)	
	Accuracy	Standard deviation
Normal samples	98.3	0.2
Ball fault with size 0.007"	98.5	0.1
Inner race fault with size 0.007"	100	0
Outer race fault with size 0.007"	100	0

Fig. 8 shows the distribution of training samples and $V-mc_h$ s with different failure state, which is disposed by PCA. Through clonal variation, quadratic programming and combined mutation, it can be seen that each training samples only forms a memory cell to represent the distribution of a

class of samples, which largely reduces the complexity of calculation. Because the r value and δ value are set too much and more combining mutations are happened among $V-mc_h$ s. Table 4 shows the average classification accuracy of 10 times test. It can be seen that the proposed V-AIR algorithm have a good classification effect for all kind of samples: the testing accuracy for the inner race and outer race faults are 100 % respectively; for the normal samples, it is 98.3 %; for the ball fault, it is 98.5 %. Thus, our fault diagnosis method is effective in analog fault diagnosis.

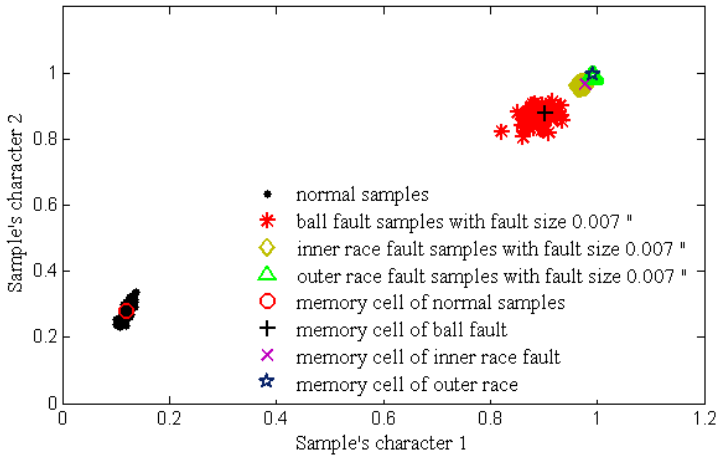


Fig. 8. The training samples of bearing and their memory cells

Table 5. Used parameters in V-AIR for bearing data

Used parameters	The parameter value
Mutation rate	0.6
Average stimulation threshold	0.9
Clonal rate	50
Number of resources allowed in V-AIR	40
k value for k -weight nearest neighbor	1
r value	0.4
δ value	0.0075
E_t value	0.001

5.2. The fault diagnosis of reciprocating compressor valves

The reciprocating compressor is a class of large general machinery widely used in petroleum chemical industry, mining, refrigeration and other industries. Its structure is complex, widely excitation source and the vibration is mainly put on multi-source impact signal such as air flow in and out of the cylinder, valve plates dropped into the seat or piston striking the block etc., which can't be diagnosed by the traditional fault diagnosis methods.

To large reciprocating compressor, valve is the key components and according to some statistics, more than 60 % of the failure happens on valve. As valve is a typical reciprocating working part, it needs to be monitored and diagnosed based on intelligent fault diagnosis techniques.

Fig. 9 and Fig. 10 show the test site condition and vibration testing principle respectively. In the valve vibration monitoring, we used the high-frequency acceleration sensor with measurement frequency range 0.0002-10 kHz to gather the high frequency vibration signal of valve. The collected signal is transmitted to computer through 32 channel programmed multifunctional signal conditioner and intelligent data collection analyzer, which prepares for the subsequent data analysis. The sample length is 200000 points with sampling frequency 50 kHz and 200 groups of data samples for each fault condition.



a) The tested reciprocating compressor



b) The used test instruments

Fig. 9. The test site condition

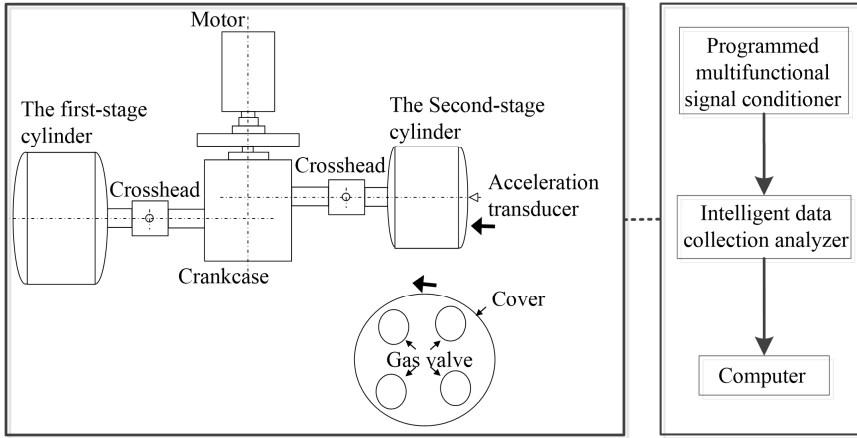


Fig. 10. The reciprocating compressor structure and the vibration testing principle

Fig. 11 gives the vibration signal of valve gap, plate fracture and spring deficiency fault. Due to the shock wave and impact time changing all the time, it is very difficult to detect the symptom of time domain waveform. Thus, in order to effectively diagnose the reciprocating compressor valve failure, we use the method of V-AIR for fault identification. Fig. 12 shows the distribution of training samples and $V-mc_h$ s with different failure state, which is also disposed by PCA. In the V-AIR method, 100 groups of data samples are used for training with the rest 100 groups of data samples for testing. From Fig. 12, we can learn that different fault samples are completely non-linear separable and have different concentration degree.

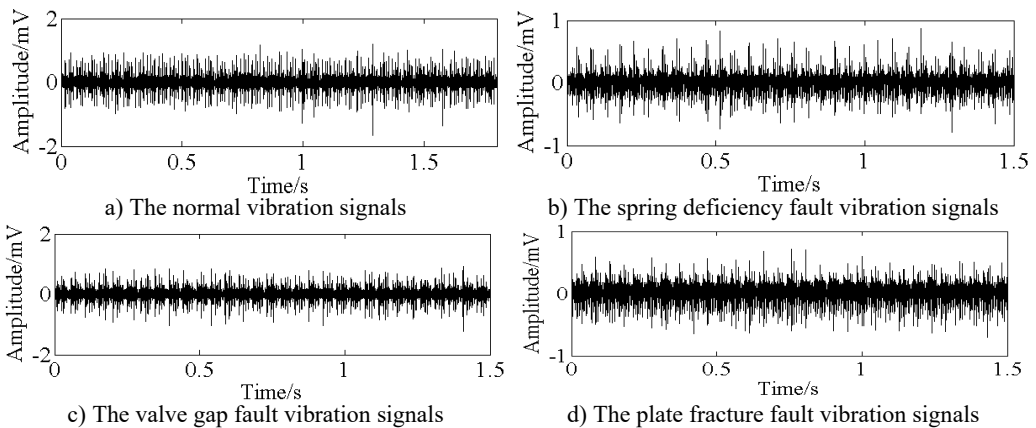


Fig. 11. The time domain vibration waveform of reciprocating compressor valve

Like the method used in bearing analog fault diagnosis, Tables 6 and Table 7 show the average classification accuracy of 10 times test and the parameter values used in V-AIR respectively. It can be seen that the proposed V-AIR algorithm can effectively classify all kind of fault samples in the highly nonlinear separable condition: the testing accuracy for the normal samples are 89.5 % with 86 number of memory cells; with only 1 memory cell, the plate fracture and spring deficiency fault reach 90.5 % and 90.4 % respectively; for the valve gap fault, it is 95.2 %. Thus, the result proves that V-AIR is a useful method in equipment fault diagnosis.

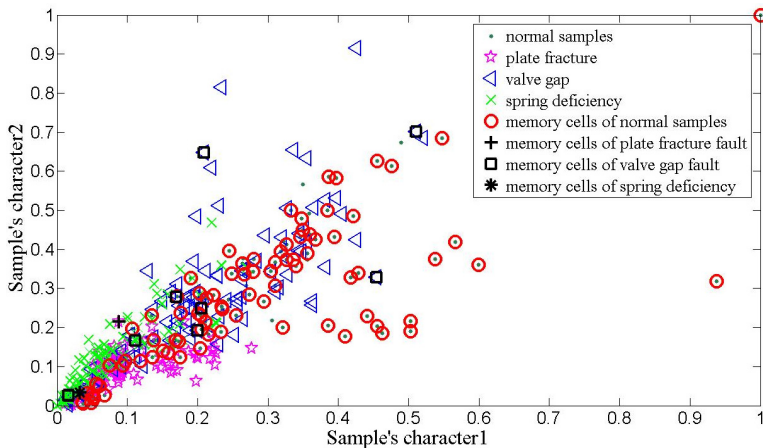


Fig. 12. The training samples of reciprocating compressor valve fault and their memory cells

Table 6. The reciprocating compressor valve fault classification accuracy of V-AIR

Type of samples	Number of memory cells	The classification accuracy of V-AIR (%)	
		Accuracy	Standard deviation
Normal samples	86	89.5	0.5
Plate fracture fault	1	90.5	0.3
Value gap fault	9	95.2	0.2
Spring deficiency fault	1	90.4	0.6

Table 7. Used parameters in V-AIR for reciprocating compressor valve fault

Used parameters	The parameter value
Mutation rate	0.6
Average stimulation threshold	0.5
Clonal rate	100
Number of resources allowed in V-AIR	40
k value for k -weight nearest neighbor	1
r value	0.75
δ value	0.02
E_t value	0.001

6. Conclusions

In this study, the inhibition mechanism between antibodies of AIRS that is among the most important classification systems of Artificial Immune Systems was changed with a new one that was formed with combining mutation. The fixed weight memory cells were also replaced by the variable weight memory cells with the weight changing according to the training samples' density distribution. After training, the k -weight nearest neighbor algorithm is used to determine the classes of test samples.

In the application phase of this study, four important UCI datasets: Diabetes, Ionosphere, Sonar and Iris were used. In the data classifications stage, the analyses were conducted both for the

comparison of reached classification accuracy with other classifiers in UCI web site and the effects of the number of memory cells. According to the application results, V-AIR showed a considerably high performance with regard to the classification accuracy for all of the four dataset. The reached classification accuracy of V-AIR for the Diabetes, Ionosphere, Sonar and Iris were 74.5 %, 96.3 %, 86.8 % and 97.3 % respectively, which were all higher than the AIRS method. V-AIR is going one step ahead than the original AIRS with the aid of improvements done in the algorithm. The proposed change in this study has not only produced very satisfactory results, but also decreased the number of memory cells for some UCI datasets.

As an application example, the V-AIR also shows high classification accuracy for bearing data with only one memory cell for a kind of bearing data. In the process of reciprocating compressor valve fault diagnosis, it also shows great fault diagnosis effect. Thus, this method can be used for fault diagnosis successfully.

Acknowledgements

This work was supported by National Natural Science Foundation of China (50475183), the Specialized Research Fund for the Doctoral Program of Higher Education (20103 108110006) and Shanghai Science and Technology Commission Basic Research Project (11JC140 4100).

References

- [1] **Dasgupta D., Yu S., Nino F.** Recent advances in artificial immune systems: models and applications. *Applied Soft Computing*, Vol. 11, Issue 2, 2011, p. 1574-1587.
- [2] **Dasgupta D., Yu S., Majumdar N. S.** MILA-multilevel immune learning algorithm and its application to anomaly detection. *Soft Computing*, Vol. 9, Issue 3, 2011, p. 172-184.
- [3] **Forrest S., Perelson A. S., Allen L., et al.** Self-nonsel self discrimination in a computer. *IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, p. 202-212.
- [4] **Mousavi M., Abu Bakar A., Zainudin S., et al.** Negative selection algorithm for dengue outbreak detection. *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 21, Issue 2, 2013, p. 2345-2356.
- [5] **Gonzalez F., Dasgupta D., Kozma R.** Combining negative selection and classification techniques for anomaly detection. *Congress on Evolutionary Computation*, 2002, p. 705-710.
- [6] **González F., Dasgupta D., Gómez J.** The effect of binary matching rules in negative selection. *Genetic and Evolutionary Computation – GECCO*, First Edition, Springer Inc., Chicago, 2003.
- [7] **Ji Z., Dasgupta D.** V-detector: an efficient negative selection algorithm with “probably adequate” detector coverage. *Information Science*, Vol. 179, Issue 10, 2009, p. 1390-1406.
- [8] **Ji Z., Dasgupta D.** Real-valued negative selection algorithm with variable-sized detectors. *Genetic and Evolutionary Computation – GECCO*, First Edition, Springer Inc., Seattle, 2004.
- [9] **Shapiro J. M., Lamont G. B., Peterson G. L.** An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. *Conference on Genetic and evolutionary computation*, First Edition, ACM., Washington, 2005.
- [10] **Zeng J., Tang W., Liu C., et al.** Real-valued negative selection algorithm with variable-sized self radius. *Information Computing and Applications*, First Edition, Springer Inc., Chengdu, 2012.
- [11] **Jerne N. K.** Towards the network theory of the immune system. *Annales d’Immunologie*, Vol. 125, 1974, p. 373-389.
- [12] **Farmer J. D., Packard N. H., Perelson A. S.** The immune system adaptation and machine learning. *Physica D*, Vol. 22, 1986, p. 187-204.
- [13] **Ultsch A.** U*-Matrix: a Tool to Visualize Clusters in High Dimensional Data. *Technical Report*, Philipps-University Marburg, Marburg, Germany, 2003.
- [14] **Kohonen T.** *Self-organizing Maps*. Springer Inc., Berlin, 2001.
- [15] **Bezerra G. B., Barra T. V., De Castro L. N., et al.** Adaptive radius immune algorithm for data clustering. *4th International Conference on Artificial Immune Systems*, First Edition, Springer Inc., Banff, 2005.
- [16] **Timmis J., Neal M., Hunt J.** An artificial immune system for data analysis. *Biosystems*, Vol. 55, Issues 1-3, 2000, p. 143-150.

- [17] **Timmis J., Neal M.** A resource limited artificial immune system for data analysis. *Knowledge Based System*, Vol. 14, Issue 3-4, 2001, p. 121-130.
- [18] **Watkins A. B.** AIRS: a Resource Limited Artificial Immune Classifier. Mississippi State University, Mississippi, 2001.
- [19] **Polat K., Şahan S., Güneş S.** A new method to medical diagnosis: Artificial immune recognition system (AIRS) with fuzzy weighted pre-processing and application to ECG arrhythmia. *Expert System Application*, Vol. 31, Issue 2, 2006, p. 264-269.
- [20] **Polat K., Güneş S., Tosun S.** Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing. *Pattern Recognition*, Vol. 39, Issue 11, 2006, p. 2186-2193.
- [21] **Polat K., Şahan S., Kodaz H., et al.** Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism. *Expert System with Application*, Vol. 32, Issue 1, 2007, p. 172-183.
- [22] **Ozsen S., Gunes S.** Performance evolution of a newly developed general-use hybrid AIS-ANN system: AaA-response. *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 21, Issue 6, 2013, p. 1703-1709.
- [23] **Li D. Y., Du Y.** *Artificial Intelligence with Uncertainty*. First Edition, National Defense Industry Press, Beijing, 2005.
- [24] **Wang S. L., Cheng G. Q., Li D. Y., et al.** A try for handling uncertainties in spatial data mining. 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, First Edition, Springer Inc., Wellington, 2004, p. 149-153.
- [25] **Igawa K., Ohashi H., et al.** A negative selection algorithm for classification and reduce of the noise effect. *Applied Soft Computing*, Vol. 9, Issue 1, 2009, p. 431-438.
- [26] **Zheng H. B., Chen X. Z., Li Z. Y., et al.** Implementation and application of a neural network fault diagnosis system based on wavelet transform. *Transactions of the Chinese Society of Agricultural Machinery*, Vol. 5, Issue 1, 2002, p. 73-76.



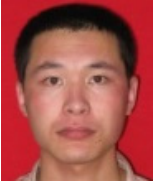
Hongli Zhang received the B.S. degree in Packaging Engineering from North University of China, in 2008, and his M.S. and Ph.D. degrees in Mechanical Engineering from Lanzhou University of Technology and Shanghai University, China, in 2011 and 2014, respectively. He is a Postdoctoral researcher in School of Mechatronic Engineering and Automation and Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University. His research interests include intelligent fault diagnosis, pattern recognition, and signal processing.



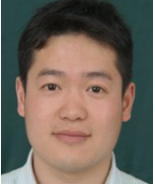
Liu Jicheng received the M.S. and Ph.D. degrees in Mechanical Design and Theory from Harbin Institute of Technology, China, in 2004 and 2009, respectively. He is an Associate Professor in School of Mechatronic Engineering and Automation, Shanghai University. His research interests include mechanism, structure design and reliability analysis.



Erping Zhou received her Bachelor degree in Mechanical Engineering in 1982 from the Shanghai University of Science and Technology, China and her Ph.D. degree in applying artificial intelligent system into manufacturing processes in 1996 from Staffordshire University, UK. She started her academic career at the Shanghai University of Science and Technology in 1982. In 1988, she received an overseas' visiting academic scholarship and came to Britain in School of Electrical Electronic and Systems Engineering, University of Wales College, Cardiff, (UK), and then as a Research Assistant at Staffordshire University in 1990. She joined the University of Bolton in 1992. Currently she is a Reader in School of Engineering, Sports and Sciences. Her research interests span the fields of control systems design, neural networks and mechatronics systems with applications specifically in the manufacturing industry. Other experience relevant to higher education includes the program leaders of the M.Sc. Systems Engineering and Engineering Management collaboration with South Westphalia University of Applied Sciences Germany and program leader of B.Sc. (Hons) Mechatronics top-up.



Li Dong received his B.S. degree from Shanxi Normal University, Shanxi, China, in 2004, the M.S. degree from China University of Petroleum (Huadong), Qingdao, China, in 2007. Now he is working toward the Ph.D. degree at Shanghai University, Shanghai, China. His research interests include machine learning and mechanical fault diagnosis.



Wang Bo received the B.S. degree in Mechanical Engineering from Henan University of Science and Technology of China, in 2005, and his M.S. degrees in Aeronautical and Astronautical Manufacturing Engineering from Nanjing University of Aeronautics and Astronautics, China, in 2009. He is pursuing the Ph.D. degree in School of Mechatronic Engineering and Automation, Shanghai University. His research interests include intelligent fault diagnosis, pattern recognition, and signal processing.



Kunju Shi graduated in Measurement and Control Technology from Heibei University of Technology, China, in 2009 and completed his Master Degree in Machine Design Manufacture and Automation from Lanzhou University of Technology, China, in 2013. Now he is pursuing his doctoral degree in Mechanical-Electronic Engineering from Shanghai University, China. His research direction is mechanical fault diagnosis.