Smart Structures and Systems, Vol. 8, No. 1 (2011) 69-92
DOI: http://dx.doi.org/10.12989/sss.2011.8.1.069

Emergent damage pattern recognition using immune network theory

Bo Chen*[1,2] and Chuanzhi Zang[1,3]

[1]*Department of Mechanical Engineering - Engineering Mechanics, Michigan Technological University, 815 R.L. Smith Building, 1400 Townsend Drive, Houghton, MI 49931, USA*
[2]*Department of Electrical and Computer Engineering, Michigan Technological University, USA*
[3]*Shenyang Institute of Automation, Chinese Academy of Science, Nanta Street 114, Shenyang, Liaoning, P.R. China, 110016*

(Received April 15, 2010, Accepted November 28, 2010)

Abstract. This paper presents an emergent pattern recognition approach based on the immune network theory and hierarchical clustering algorithms. The immune network allows its components to change and learn patterns by changing the strength of connections between individual components. The presented immune-network-based approach achieves emergent pattern recognition by dynamically generating an internal image for the input data patterns. The members (feature vectors for each data pattern) of the internal image are produced by an immune network model to form a network of antibody memory cells. To classify antibody memory cells to different data patterns, hierarchical clustering algorithms are used to create an antibody memory cell clustering. In addition, evaluation graphs and *L* method are used to determine the best number of clusters for the antibody memory cell clustering. The presented immune-network-based emergent pattern recognition (INEPR) algorithm can automatically generate an internal image mapping to the input data patterns without the need of specifying the number of patterns in advance. The INEPR algorithm has been tested using a benchmark civil structure. The test results show that the INEPR algorithm is able to recognize new structural damage patterns.

Keywords: emergent pattern recognition; immune network theory; hierarchical clustering; artificial immune systems.

1. Introduction

One of the major challenges in real-world engineered systems is to identify emerging patterns, so, the appropriate control strategies can be applied to avoid potential disasters. Many engineered systems, such as critical civil infrastructures, power grids, and environmental systems, have experienced unexpected failures due to unpredictable working conditions and the increased complexity of systems. To avoid these unexpected behaviors, there is a need to enhance the ability of engineered systems in terms of adaptability, autonomy, and reliability. The biological immune system is able to handle this challenging problem much more efficient than engineered systems (Negoita 2005). The complex biological systems have their ability to display emergent behaviors, which are often resilient and robust. These emergent pattern recognition and adaptation features are desirable in engineered systems.

*Corresponding Author, Assistant Professor, E-mail: bochen@mtu.edu

Based on this observation, immune-inspired computational approaches have been investigated, and a new area of Artificial Immune Systems (AIS) is formed.

The artificial immune systems can be defined as the abstract or metaphorical computational systems developed using ideas, theories, and components, extracted from the natural immune system (De Castro and Timmis 2002). The natural immune system consists of a number of different organisms and cells, such as *B*-cells. The surface receptor (antibody) of an immune cell can recognize and bind antigens (virus). When a *B*-cell encounters a nonself antigen that has sufficient affinity (similarity) with its receptors, the antibody of the *B*-cell binds to the antigen, marking it for destruction. The AIS seems best suited to handle the great complexity of the reality (Castiglione *et al.* 2001). The reason behind this is that the natural immune system incorporates a variety of artificial intelligence techniques, such as pattern recognition through a network of collaborating agents, adaptive learning through memory, and an advanced selection mechanism of the best *B*-cells (Lanaridis *et al.* 2008). The desirable characteristics of the immune system have inspired the development of artificial immune systems for various applications (Hart and Timmis 2008). For example, Zhong *et al.* (Zhong *et al.* 2006) employ an AIS-based unsupervised machine-learning algorithm to perform remote sensing image classification. Chen and Zang (Chen and Zang 2009, Chen and Zang 2011) develop AIS-based classification algorithms for supervised and unsupervised structural damage detection and classification. (Polat *et al.* 2006) adopt a hybrid method that consists of fuzzy weighted pre-processing and artificial immune recognition system for heart disease diagnosis. (Dasgupta *et al.* 2004) investigate a real-value immune negative selection algorithm for aircraft fault detection. Freitas and Timmis (Freitas and Timmis 2007) use a problem-oriented approach for the design of AIS for data mining.

Structural damage detection using structural health monitoring (SHM) systems have been investigated intensively in the past few decades. (Chang *et al*. 2003) review a number of research projects aiming to improve the damage detection methods including the use of novel signal processing, new sensors, and control theory. Sumitro and Wang (Sumitro and Wang 2005) qualitatively compare sensor technologies applied to SHM systems and introduce some new sensory technologies such as GPS-based MMS (movement monitoring systems), PDMD (peak displacement memory devices), and FOS (fiber optic sensors). (Weng *et al*. 2008) present two modal identification methods that extract dynamic characteristics from output-only data sets collected by a wireless structural monitoring network. (Nagayama *et al*. 2007) investigate the effects of time synchronization accuracy and communication reliability in SHM applications and examine coordinated computing for the management of large amount of SHM data. (Lu *et al.* 2008) use a wireless sensor system to detect structural damage with autoregressive (AR) and the autoregressive model with exogenous inputs (ARX) method.

This paper presents an immune-network-based emergent pattern recognition algorithm (INEPR). Different from authors' previous study (Chen and Zang 2009, Chen and Zang 2011) which categorize a detected damage pattern to one of a number of known patterns, the INEPR algorithm aims to recognize new data patterns that are not known in advance. The presented immune network-based approach achieves emergent pattern recognition by dynamically constructing a network of antibody memory cells as an internal image for the input data patterns. The connection of antibody memory cells depends on the affinity among them. The antibody memory cells are evolved through a clonal immune response initiated by each of input data. The newly generated antibody memory cells with high affinities to the input data pattern (antigen) will be recruited into the network. The antibody memory cells with low affinities to the input data pattern will be eliminated from the network. The continuous recruitment and elimination of antibody memory cells allows the internal image dynamically updating its

members to match to the input data patterns.

The antibody memory cells generated by the immune network model are classified into a number of clusters (patterns), which is a mapping of the input data patterns. Many clustering algorithms suffer from the limitation that the number of clusters needs to be determined in advance (Li *et al*. 2008). To address this issue, a number of researchers have investigated different ways to automatically determine the best number of clusters. There are five commonly used approaches to estimate the number of clusters: cross validation, penalized likelihood estimation, permutation tests, resembling, and finding the knee of an error curve (Salvador and Chan 2004). In this paper, evaluation graphs and L method are used to find the best number of clusters for the antibody memory cell clustering.

The rest of the paper is organized as follows. Section 2 gives an overview of the immune-network-based emergent pattern recognition algorithm. Section 3 introduces a benchmark structure used for algorithm verification. Section 4 describes the generation of antibody memory cells using an immune network computational model. Sections 5 and 6 present the construction of antibody memory cell clustering using hierarchical clustering algorithms and the determination of the best number of clusters using evaluation graphs and the L method. Section 7 validates the ability of the INEPR algorithm for the recognition of new damage patterns using a benchmark civil structure. Section 8 discusses the impact of model parameters on the number of memory cells and the impact of cluster dissimilarity, number of input data points, and the dissimilarity measure on the performance of the L method. Finally, conclusions are made in Section 9.

2. Overview of immune-network-based emergent pattern recognition algorithm

The goal of emergent pattern recognition is to recognize new data patterns when they emerge. The immune network allows its components to change and learn patterns by changing the strength of connections between individual components. The immune network theory (Jerne 1974) suggests that the immune system is composed of a regulated network of cells and molecules that recognize one another even in the absence of antigens. As shown in Fig. 1(a), an antigen is recognized by an antibody through the binding between the paratope of the antibody and the epitope of the antigen. An antibody possesses a unique idiotope, which can be recognized by the paratopes of other antibodies. The behavior of the immune network theory is illustrated in Fig. 1(b) (De Castro and Timmis 2002). When the immune system is primed with an antigen *Ag*, its epitopes are recognized by a set of different paratopes, called P1. The set P1 also recognizes a set of idiotopes of antibodies, called I2. Since the set P1 recognizes both the epitopes of the antigen and I2, the set I2 is called the internal image of the epitopes of the antigen. The set I2 is also recognized by a set P2 of paratopes of antibodies. Following this recognizing scheme, an immune network of interaction is formed (Timmis *et al.* 2008).

The presented immune-network-based emergent pattern recognition algorithm uses feature vectors and their affinities to find an internal image for the input data patterns. In pattern recognition, the patterns to be classified are usually the groups of measurements, defining points in an appropriate multidimensional space (Theodoridis and Koutroumbas 2008). The measurements used for the classification are described by features. If p features are used f_i, i=1, 2,…, p, these p features can form a feature vector $F=(f_1, f_2,\ldots, f_p)^T$, where T denotes transposition. A feature vector is a point in P dimensional space R^P. The affinity of two measurements is defined as a function of the distance between the corresponding feature vectors of the measurements. Usually, the shorter distance means

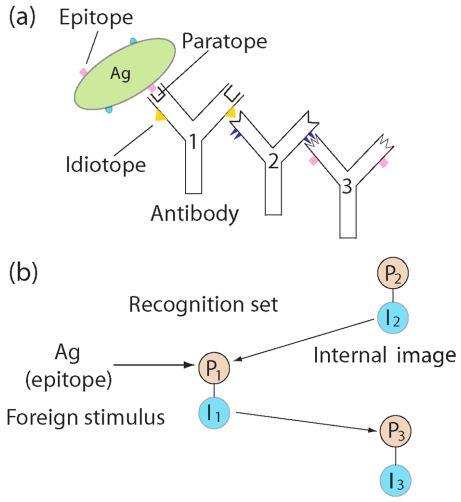

Fig. 1 Immune network (De Castro and Timmis 2002)

higher affinity and the longer distance means lower affinity. Let $\beta = (\beta_1, \beta_2,...,\beta_P)^T$ and $\gamma = (\gamma_1, \gamma_2,..., \gamma_P)^T$ denote two feature vectors. The affinity between the two feature vectors can be defined as Eq. (1), where $dis(\beta, \gamma)$ denotes the Euclidian distance between two feature vectors as shown in Eq. (2).

$$aff(\beta, \gamma) = 1/dis(\beta, \gamma) \tag{1}$$

$$dis(\beta, \gamma) = \left(\sum_{i=1}^{p} (\beta_i - \gamma_i)^2 \right)^{\frac{1}{2}} \tag{2}$$

The INEPR algorithm uses time series of sensor data, such as acceleration, collected by sensors of a monitoring network for damage pattern recognition. Since the dimension of a sensor data time series is too large to be used as feature vectors, it is important to find a representation at a lower dimensionality that preserves the fundamental characteristics of the original time-series data. The dimension reduction is achieved through feature extraction. Feature extraction consists of two equally important parts: feature selection and feature generation. In feature selection, data attributes with high discrimination capability are identified that can lead to large between-class distance and small within-class variance within the feature vector space. In feature generation, the goal is to discover compact and informative representations, based on the feature selection findings, for a given set of sensor measurements. Many high-level representations of time series data have been proposed for data mining, including Single Value Decomposition, Discrete Fourier Transformation, Discrete Wavelet Transformation, and Piecewise Approximation. In this study, autoregressive algorithm is chosen to model a sensor data time series. This method is effective in detecting bulk changes in structural properties (e.g., damage that affects the global mass or stiffness properties of the system). For *AR*-based feature extraction, each sensor data time series X is fitted into an *AR* model of order p as shown in Eq. (3).

$$x_k = \sum_{i=1}^{p} \alpha_i \, x_{k-i} + r_k \quad k = p+1, \ldots, n \tag{3}$$

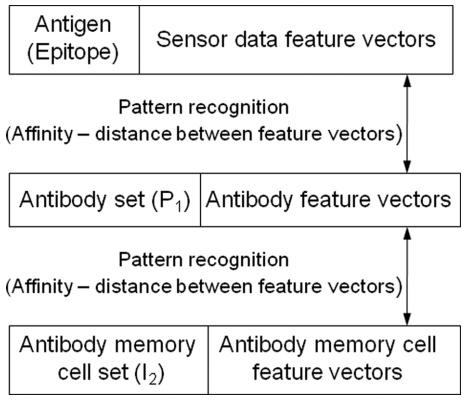

Fig. 2 Immune network components in the INEPR algorithm

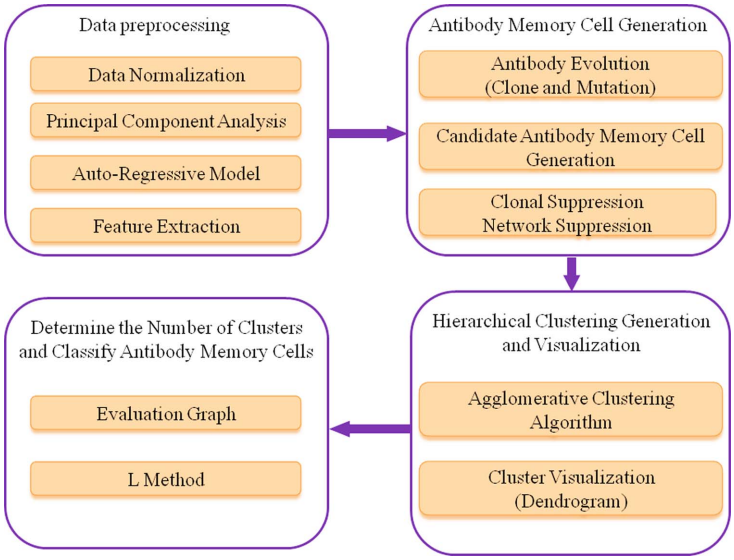

Fig. 3 Overview of the INEPR algorithm

where α_i, i = 1,2,... p is the coefficient of the AR model; r_k, $k = p + 1,..., n$ is the residual between the measurement data and the AR model value. The vector, $\alpha = (\alpha_1, \alpha_2,...,\alpha_p)^T$, a collection of the AR coefficients, is selected as the feature vector of the sensor data time series X.

To generate antibody memory cells, sensor data feature vectors are used as antigens to stimulate an antibody set as shown in Fig. 2. The antibodies whose feature vectors have high affinities with the sensor data feature vectors will remain in the antibody group $P1$. A small percentage of antibodies with highest affinities to the sensor data feature vector will be selected as antibody memory cells to join the $I2$ set. The generated antibody memory cells serve as an internal image of the input data patterns. The INEPR algorithm consists of four major steps as shown in Fig. 3. These steps include data preprocessing, antibody memory cell generation with immune network model, generating hierarchical clustering of the antibody memory cells using hierarchical clustering algorithms, and classifying the antibody memory cells by automatically determining the number of clusters.

In the first step, sensor data are normalized to remove environmental effects. Sensor data from multiple

sensors are reduced to lower dimensions using Principal Component Analysis (PCA) method. To extract feature vectors from sensor data, the resulting time series from the *PCA* method are fitted into *AR* models. The coefficients of *AR* models are used to form feature vectors for each data pattern. In the second step, an immune-network-based computational model is used to dynamically construct a network of antibody memory cells to represent the input data patterns. The number of antibody memory cells should be less than the number of original input data feature vectors. The connection of antibody memory cells depends on the affinity among them. The antibody memory cells are generated through the stimulation of antigens (input data) to the antibody set and keeping the antibodies with high affinities to the antigen. In the third step, a hierarchical clustering algorithm is used to create a hierarchy of nested clusterings for the generated antibody memory cells. In the fourth step, the INEPR algorithm employs the evaluation graphs and the L method proposed in (Salvador and Chan 2004) to determine the best number of clusters and uses this number to classify antibody memory cells into different clusters to represent each data pattern.

3. Benchmark structure used for algorithm verification

To test algorithms presented in this paper, a benchmark structure proposed by the IASC-ASCE (International Association for Structural Control - American Society of Civil Engineers) SHM Task Group (Johnson *et al.* 2000) is selected. The benchmark structure shown in Fig. 4 is a 2×2 bay, four story scaled structure in the Earthquake Engineering Laboratory at the University of British Columbia (UBC). In experimental tests, a number of damage cases were simulated by removing braces and losing bolts in the test structure. The details of the simulated damage patterns are listed in Table 1.

In experimental tests, a DasyLab data acquisition system was used to acquire 16 channels of data in each test (Beck *et al.* 2002). A total of 15 accelerometers were instrumented in benchmark structure to measure the acceleration of the structure, three accelerometers for each level. The 16th channel

Fig. 4 IASC-ASCE benchmark structure (Photo courtesy Prof. Carlos Ventura, UBC)

Table 1 The configurations of the simulated damage patterns

Configuration	Data pattern	Description
1	Normal	Fully braced configuration
2	Damage 1	Missing all east side braces
3	Damage 2	Removed braces on all floors in one bay at south east corner
4	Damage 3	Removed braces on 1st and 4th floors in one bay at south east corner
5	Damage 4	Removed braces on 1st floor in one bay at south east corner
6	Damage 5	Removed braces on all floors on east face, and 2nd floor braces on north face
7	Damage 6	All braces removed on all faces
8	Damage 7	Configuration 7, plus loosen bolts on all floors - both ends of beams on east face, north side
9	Damage 8	Configuration 7, plus loosed bolts on floors 1 and 2 - both ends of beams on east face, north side

collects the excitation or ambient noise. Tests were performed with three types of excitation: ambient vibration, electrodynamic shaker, and sledge hammer for impact testing. The data directory contains three folders, ambient data directory, shaker data directory, and hammer test data directory. Each configuration has one data file. All data files begin with the characters 'shm'. The next two characters describe the test configuration, and the last 1-3 characters describe the excitation type.

To generate feature vectors, 54 time series are formed for each data pattern. Each time series consists of 4000 data points. For each sensor, the first time series starts from 3000th data point of the sensor data file. The second time series is formed by advancing 150 data points from the first time series. In other words, it starts from 3150th data point of the sensor data file. To remove environmental effects, sensor data are normalized. Let matrix $Z = (z_{ij})_{m\times n}$, $i = 1,2,...,15$, $j = 1,2,...,4000$ denote time series from 15 accelerometers and $\vec{z}_i = (z_{i1}, z_{i2}, \ldots, z_{in})$ denote the time series of the ith accelerometer. The normalized acceleration data $Y = (y_{ij})_{m\times n}$ can be calculated by Eq. (4)

$$y_{ij} = \frac{z_{ij} - \mu_i}{\sigma_i} \; j = 1, 2, \ldots, n \tag{4}$$

where y_{ij} is the normalized value of the corresponding z_{ij}, μ_i and σ_i are the mean and standard deviation of the time series $\vec{z}_i$. Fifteen time series are then compressed into one time series using the principal component analysis method. Once 54 of compressed time series are generated for each data pattern, the autoregressive models are used to fit to 54 time series. The AR order is selected to be 20. The coefficients of the AR models are used to form 54 feature vectors for each pattern.

4. Antibody memory cell generation using an immune-network-based computational model

The aim of the immune network model is to generate a set of antibody memory cells for the input data patterns. The antibody memory cells can represent the data structure of the input data set. The memory cells are interconnected by links with associated connection strength. The connection strength among memory cells depends on their affinities. The memory cells are generated by a series of clonal immune responses that are initiated by antigens (input data) to the antibody set. As

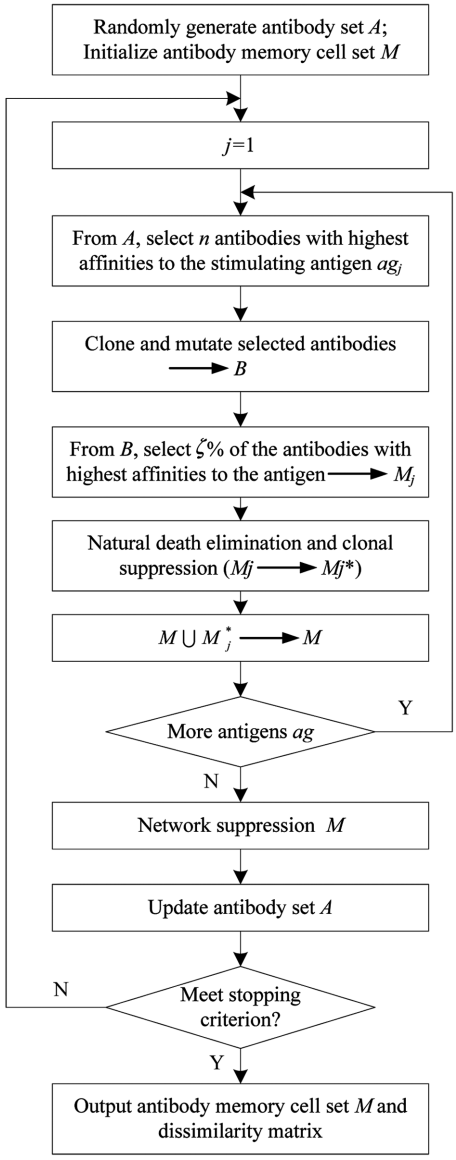

Fig. 5 Flowchart of the immune network model

proposed by immune network theory, antibody memory cells will compete for antigenic recognition. The antibodies that can successfully recognize antigen (having high affinity with the antigen) will be cloned and mutated. The newly generated antibodies with high affinities to the antigen will be recruited into the antibody memory cell set. The antibodies who fail to recognize antigen (having low affinity with the antigen) will be eliminated from the antibody memory cell set to improve the affinity level of the representative antibody memory cells. The continuous recruitment and elimination of memory cells not only provides a competition mechanism to control the survival of memory cells in the network, but also offers great potential to discover memory cells which are able to bind with unpredictable invaders (new data patterns) (De Castro and Timmis 2002).

A number of computational models based on the immune network theory have been developed and applied for data mining, classification, pattern recognition, and multimodal function optimization (Shen *et al.* 2008). aiNet (De Castro and Von Zuben 2001) is one of the artificial immune network models with the goals of clustering and filtering input data set. The output of the aiNet model is a reduced data set of the input data and data structure information, including the spatial distribution of antibody memory cells. Fig. 5 shows the flow chart of the aiNet model. The antibody set A and the antibody memory cell set M are firstly initialized. The initial antibody set A is randomly generated, and the initial antibody memory cell set M is an empty set. Each antigen ag_j stimulates the evolution of the antibody set A. The affinities among the stimulating antigen and the antibodies in the antibody set A are calculated. The antibody set A is sorted in descending order based on the affinity values. The top n antibodies in A are selected. The selected antibodies will be cloned and mutated, and saved to the data set B. The $\zeta\%$ of the cloned and mutated antibodies in B with highest affinities to the antigen ag_j is saved to the clonal matrix M_j. The antibody clones in M_j will suffer a natural death elimination and clonal suppression. In the natural death elimination process, antibodies in M_j, whose distances to the antigen ag_j are greater than the natural death threshold σ_d, are eliminated. The remaining antibody clones in M_j will subject to a clonal suppression process. If the distance between two antibody clones is less than the suppression threshold σ_s, one of the antibodies will be eliminated from M_j. The antibody memory cells after the natural death elimination and clonal suppression process form a resultant clonal memory M_j^* for the antigen ag_j. The antibody memory cells in M_j^* are then added to the antibody memory cell set M.

After all the antigens stimulate the antibody set, the newly generated memory cell set M will suffer a network suppression process which is similar to the clonal suppression process. The purpose of the network suppression is to further reduce the number of the antibody memory cells in M. The resulting memory cell set M will be incorporated to form a new antibody set A for the next iteration. The iteration for the evolution of antibody memory cell set and the generation of antibody memory cell set will continue until the stopping criterion is met.

The aiNet model has been successfully applied to generate antibody memory cells for three data patterns of the benchmark structure: Normal, Damage 1, and Damage 3, defined in Table 1. The generation of feature vectors for these data patterns is described in Sections 2 and 3. To visualize

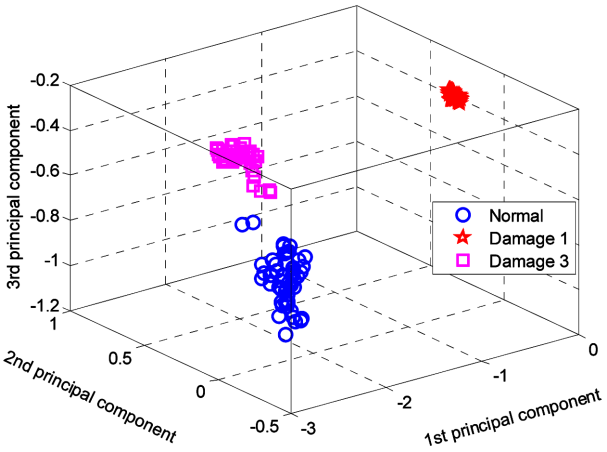

Fig. 6 Three input data patterns to the INEPR model

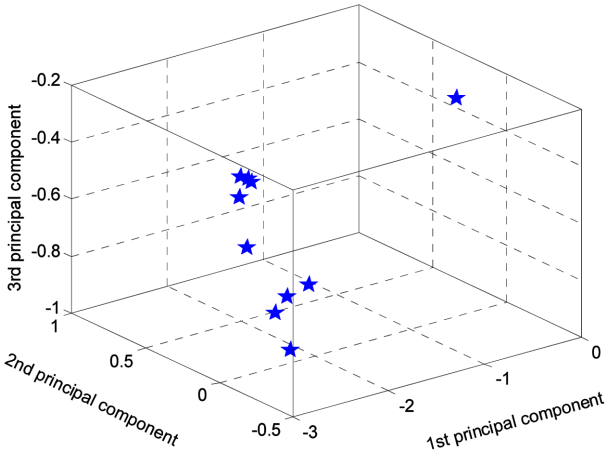

Fig. 7 Antibody memory cells of three input data patterns

feature vectors, the 20-dimensional feature vectors are reduced to three dimensions using *PCA* dimension reduction method. The feature vectors for these three data patterns are shown in Fig. 6. The antibody memory cells generated by the aiNet are a reduced data set to represent the feature vectors of these three data patterns. To visualize antibody memory cells, 20-dimensional antibody memory cells are also reduced to three dimensions using *PCA* method. The antibody memory cells after dimension reduction are shown in Fig. 7. The spatial distribution of antibody memory cells are typically represented by a certain type of distance. The Euclidean distances among antibody memory cells are used in Fig. 7.

5. The hierarchy of nested clusters of the antibody memory cells

To determine the structure of the antibody memory cell set M shown in Fig. 7, we need to find out how many clusters the M has, the number of memory cells belonging to each data pattern, and the spatial distribution of each pattern. Hierarchical clustering techniques (Chiu *et al.* 2001, Theodoridis and Koutroumbas 2008) are robust network interpretation strategies. In this work, hierarchical clustering algorithms are employed to generate a hierarchy of nested clusterings for the antibody memory cell set. The memory cell set M is a set of p-dimensional vectors $M = \{mc_i,\ i = 1,...,K\}$, where mc_i, $i = 1,...,K$ is the i-th memory cell in M and K is the cardinality of the set M. Let Δ, $\Delta=\{C_i,\ i = 1,...,m\}$, denote an m-clustering of the set M. The subset $C_1,...,C_m$ in Δ meets following rules

- $C_i \neq \varnothing, \quad i = 1, \ldots, m$
- $\cup_{i=1}^{m} C_i = M$
- $C_i \cap C_j = \varnothing, i \neq j, i, j = 1, \ldots, m$

For two clusterings Δ_1 and Δ_2, the clustering Δ_1 is said to be nested in the clustering Δ_2, Δ_1 nested in Δ_2, when *each* cluster in Δ_1 is a subset of a set in Δ_2 and the cardinality of Δ_1 is larger than the cardinality of Δ_2. For example, the clustering $\Delta_1 = \{\{mc_1, mc_4\}, \{mc_3\}, \{mc_2, mc_5\}\}$ is nested in Δ_2

$\{\{mc_1, mc_3, mc_4\}, \{mc_2, mc_5\}\}$.

The hierarchical clustering algorithms produce a series of clusterings. If the cardinality of M is K, the hierarchical clustering algorithms have K steps. At each step t, a new clustering is generated based on the clustering created at the step t-1. There are two main types of hierarchical clustering algorithms: the agglomerative and the divisive hierarchical clustering algorithms. For the agglomerative clustering algorithms, there are K clusters in the initial clustering Δ_0 and each cluster contains only one node, one memory cell in M. At each step, two clusters are merged into one new cluster. Finally, in clustering $\Delta_{K\text{-}1}$, there is only one cluster, M. The divisive algorithms follow the inverse path as the agglomerative algorithms. It starts with an initial clustering Δ_0, which contains the set M. At each following step, one selected cluster is divided into two non-empty sub-clusters. At the final step K-1, there are K clusters. In this work, we use agglomerative algorithms to generate a hierarchy of nested clusterings for the antibody memory cell set. The agglomerative algorithm scheme is stated in Algorithm 1. The clusters C_i and C_j are merged into a single cluster C_q if the distance between them is the shortest one for all the possible pairs of clusters at the level t.

Algorithm 1: agglomerative algorithm scheme

Begin
  Let initial clusters to be all memory cells $\Delta_0 = \{C_i = \{mc_i\}, i = 1, \ldots, K\}$;
  $t = 0$;
 For the iteration $t < K$ **do**
  $t = t + 1$;
  Find (C_i, C_j) such that the distance between these two clusters $d(C_i, C_j)$ is the shortest among all cluster distances at level $t = t-1$;
  Merge C_i, C_j into a new cluster C_q and form $\Delta_t = (\Delta_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$
 End iteration if $t=K$
End

When C_i and C_j are merged into a new cluster C_q, the distance between C_q and one of the old clusters C_s, $d(C_q, C_s)$, is a function of the form

$$d(C_q, C_s) = f(d(C_i, C_s), d(C_j, C_s), d(C_i, C_j)) \tag{5}$$

A number of dissimilarity measures (Theodoridis and Koutroumbas 2008) use following update function

$$d(C_q, C_s) = a_i d(C_i, C_s) + a_j d(C_j, C_s) + b d(C_i, C_j) + c|d(C_i, C_s) - d(C_j, C_s)| \tag{6}$$

Different values of a_i, a_j, b, and c correspond to different distance update schemes. Commonly used update schemes include the single link algorithm, complete link algorithm, weighted pair group method average, unweighted pair group method average, weighted pair group method centroid, unweighted pair group method centroid, and ward or minimum variance algorithm. In the following two distance update schemes are used for the performance evaluation of the algorithm presented.

- The *single link algorithm*. In this case, the following parameters are chosen: $a_i = 1/2$, $a_j = 1/2$, $b = 0$ and $c = -1/2$. The distance measure uses the smallest distance between objects in the two clusters as shown below.

$$d(C_q, C_s) = \min\{d(C_i, C_s), d(C_j, C_s)\} \tag{7}$$

- The *complete link algorithm*. The parameters for the complete link algorithm are: $a_i = 1/2$, $a_j = 1/$

2, $b = 0$ and $c = 1/2$. The distance update scheme is shown below.

$$d(C_q, C_s) = \max\{d(C_i, C_s), d(C_j, C_s)\} \tag{8}$$

The distance between two clusters (cluster-to-cluster) or two memory cells (point-to-point) is considered as the dissimilarity between them. Using different dissimilarity measures between clusters in hierarchical clustering algorithms will result in different clusterings. However, the dissimilarity measures between two memory cells will also affect the performance of clustering and classification. The initial clusters in the hierarchical clustering algorithm contain only one memory cell, and the distance between any two clusters is the distance between the two corresponding memory cells. For the antibody memory cell set $M \in R^{K \times P}$, there are K number of P-dimensional memory cells mc_1, mc_2,...,mc_K, which forms a matrix M. The most common dissimilarity measures between two memory cells mc_r and mc_s are defined as follows. These point-to-point dissimilarity measures are used to test the performance of the algorithm presented in section 8.

• Euclidean distance

$$d_{rs}^2 = (mc_r - mc_s)(mc_r - mc_s)^T \tag{9}$$

• Cosine distance

$$d_{rs} = 1 - \frac{mc_r mc_s^T}{(mc_r mc_r^T)^{1/2}(mc_s mc_s^T)^{1/2}} \tag{10}$$

• Standardized Euclidean (Seuclidean) distance

$$d_{rs}^2 = (mc_r - mc_s)D^{-1}(mc_r - mc_s)^T \tag{11}$$

where D is a diagonal matrix with diagonal elements given by v_j^2, which denotes the variance of the jth-feature over the K memory cells.

• Correlation distance

$$d_{rs} = 1 - \frac{(mc_r - \overline{mc_r})(mc_s - \overline{mc_s})^T}{((mc_r - \overline{mc_r})(mc_r - \overline{mc_r})^T)^{1/2}((mc_s - \overline{mc_s})(mc_s - \overline{mc_s})^T)^{1/2}} \tag{12}$$

where $\overline{mc_r} = \frac{1}{p}\sum_j mc_{rj}$, $\overline{mc_s} = \frac{1}{p}\sum_j mc_{sj}$.

The hierarchy of nested clusterings generated by hierarchical clustering algorithms can be visualized by dendrogram plots. The dendrogram is an effective means of representing the sequence of clusterings produced by a hierarchical clustering algorithm. Fig. 8 shows the dendrogram of the antibody memory cells shown in Fig. 7. There are 10 memory cells in Fig. 7, which correspond to leaf nodes at the bottom of the dendrogram in Fig. 8. Each level of the dendrogram corresponds to an agglomerative step, merging two clusters with shortest distance in the previous level to form a new cluster. For all leave nodes, each cluster consists of only one memory cells. At the first level, memory cells 4 and 7 form a new cluster because the distance between these two memory cells is the shortest among any other two memory cells. Memory cells 3 and 6 form a new cluster at the second level. This cluster emerging process continues until one single cluster is formed.

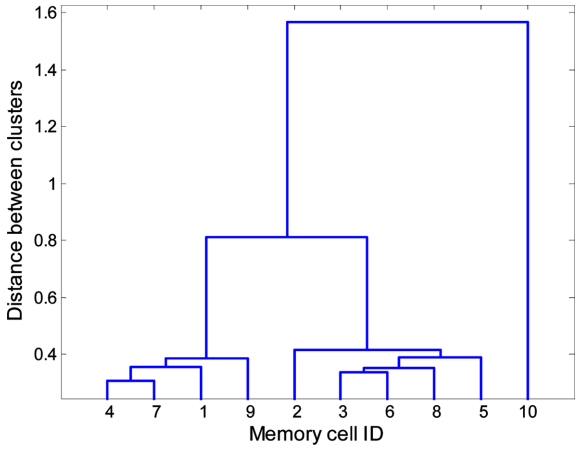

Fig. 8 The dendrogram of the antibody memory cells in Fig. 7

6. Determining the number of clusters of the antibody memory cell set

The output of the hierarchical algorithm is a hierarchy of nested clusters as shown in the dendrogram plot in Fig. 8. The dendrogram can be broken at different levels to yield different number of data patterns. Various methods to determine the best number of clusters are discussed in (Kothari and Pitts 1999, Tibshirani *et al*. 2001, Sugar and James 2003, Salvador and Chan 2004, Cheong and Lee 2008, Qinpei *et al.* 2008). In this paper, we employ evaluation graphs and the *L* method (Salvador and Chan 2004) to determine the number of clusters.

6.1 Evaluation graph

Evaluation graph is one of the evaluation methods aimed to determine an appropriate number of clusters for a given data set by evaluating the cluster quality at different number of clusters. An evaluation graph is a two-dimensional plot where the x-axis values are the possible number of clusters, and the y-axis values are dissimilarity measures of a clustering consisting of x number of clusters. The evaluation metrics used to compute the y-axis values could be dissimilarity (distance) or similarity. These metrics can be computed globally or greedily (Salvador and Chan, 2004). Global measure computes the evaluation metric based on the entire data clustering, while the greedy method computes the evaluation metric by evaluating only two emerging (splitting) clusters. The metric used in the evaluation graph should be the same as the metric used in the clustering algorithm.

To determine the best number of clusters for the antibody memory cells in Fig. 7, an evaluation graph is generated using a greedy approach, the solid blue line in Fig. 9. The y-axis values are the distances between two emerging clusters defined in Eq. (7). The evaluation graph shows that the change of cluster distances is small when the number of clusters is large. However, the cluster distances increase rapidly when the number of clusters is small, which means that very dissimilar clusters are being merged. As a result, a reasonable number of clusters should be in the region of curved area between the left and right regions of the evaluation graph, or the "knee" of the graph.

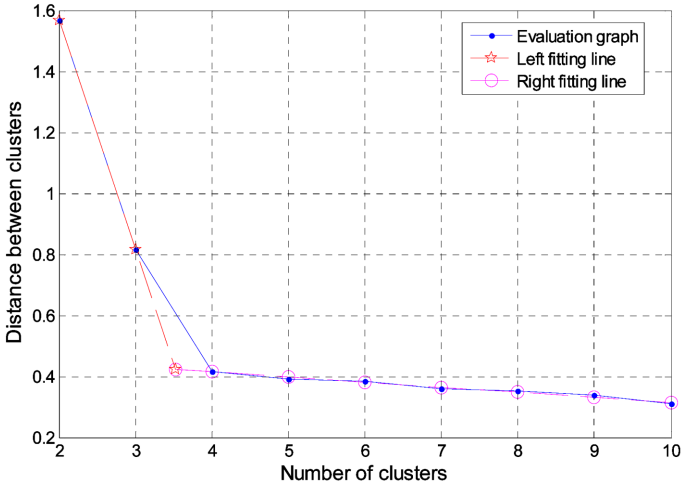

Fig. 9 Evaluation graph of the antibody memory cells and *L* method

6.2 Determining the number of clusters using L method

The evaluation graph in Fig. 9 suggests that the knee of the graph is the best choice of the number of clusters for a given data set. To find the knee of an evaluation graph, the *L* method introduced in (Salvador and Chan 2004) is employed. The *L* method is based on the shape of an evaluation graph. As shown in Fig. 9, the right and left regions of the evaluation graph (the solid blue line) presents linear characteristic, as a result, two straight lines, the dotted red line with pentagon marker on the left side and the dashed magenta line with circle marker on the right side, can be used to fit to data points in these two regions, respectively. The intersection of these two lines is located in the region of the knee, and is an approximation of the knee. An integer *x*-axis value left to the knee is then used as the best number of clusters.

For the *K* number of antibody memory cells in Fig. 7, the number of clusters can vary from 1 to *K*. When the number of the clusters is one, the whole memory cell set is considered as a single cluster. To find the best number of clusters in the memory cell set, we evaluate the cluster quality with various numbers of clusters from 2 to *K* as explained in the Algorithm 2. The *x*-axis values in the evaluation graph are partitioned into two subsets. The points with *x*-values from 2 to *s* ($2 < s < K$-1) form the first subset, denoted by S_L. The remaining points form the second subset, denoted by S_R. Two lines are sought to fit these two subsets, respectively. The best-fit line pairs can be found using various evaluation measurements. One of the possible measures could be the total least mean square error shown below.

$$g_s = \frac{s-1}{K-1} \times g(S_L) + \frac{K-s}{K-1} \times g(S_R) \tag{13}$$

where $g(S_L)$ ($g(S_R)$) is the root mean squared error of the fitting line for the points in S_L (S_R). The location of the knee, $x = s^*$, can be found using Eq. (14).

$$s^* = \arg\min_s g_s \tag{14}$$

Algorithm 2: L method to determine the number of clusters

Begin
Input the *evaluation graph*
$i = 1$;
For the iteration $i < K$-2 **do**
$i = i + 1$
Divide the points in the *evaluation graph* into two separated subsets: the left side i points form subset S_L and the rest points at the right side form subset S_R
Construct two lines that have the best fit to two subsets S_L and S_R
Calculate root mean square error $g(S_L)$ and $g(S_R)$ for the left and right fitting lines
Calculate the total root mean square error gs according to Eq. (13)
End iteration if $i = K$-2
Find the best number of clusters according to Eq. (14)
End

7. Case study

To verify the ability of the INEPR algorithm for the recognition of new data patterns, two tests were conducted using benchmark structural data. In the first test, acceleration data of the Normal, Damage 1, and Damage 3 patterns were used to find appropriate values of the INEPR parameters. As discussed in subsection 8.1, the number of memory cells depends on the values of four aiNet model, σ_s, σ_d, n, and ζ. To have a reasonable number of memory cells, appropriate values for these parameters to generate reasonable number of antibody memory cells, were found to be $\sigma_s = 0.3$, $\sigma_d = 5.5$, $n = 6$, and $\zeta = 0.2$. The feature vectors of the Normal, Damage 1, and Damage 3 patterns are shown in Fig. 6. The generated antibody memory cells using aiNet model are shown in Fig. 7. Once the antibody memory cells were generated, the single link hierarchical clustering algorithm was applied to produce a hierarchical clustering, whose dendrogram plot is shown in Fig. 8. The point-to-point dissimilarity measure was the Euclidean distance. To classify antibody memory cells, the evaluation graph for the clustering of three data patterns was calculated, and the L method was used to determine the best number of clusters as shown in Fig. 9. In this case, the best number of

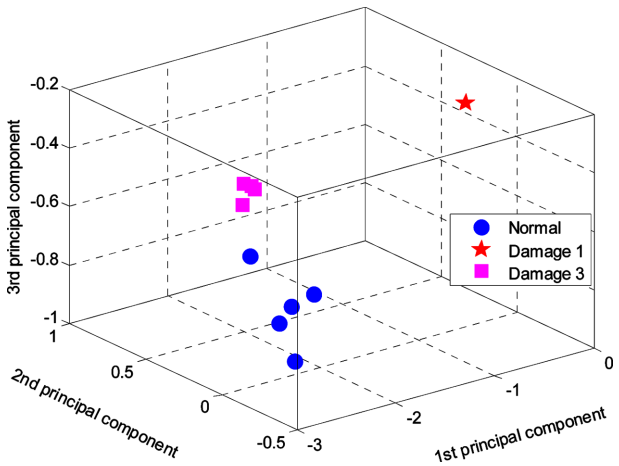

Fig. 10 Generated internal image for three input data patterns in Fig. 6

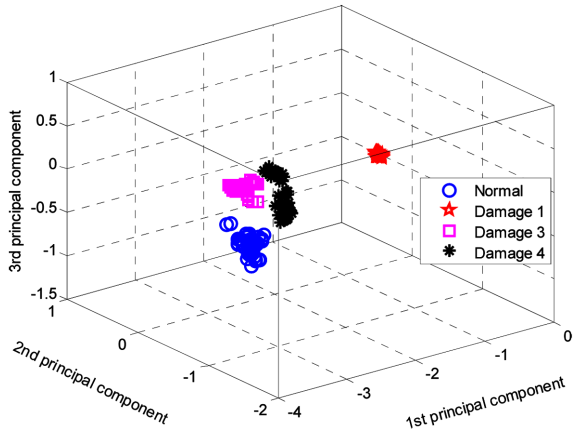

Fig. 11 Feature vectors of Normal, Damage 1, Damage 3, and Damage 4 data patterns

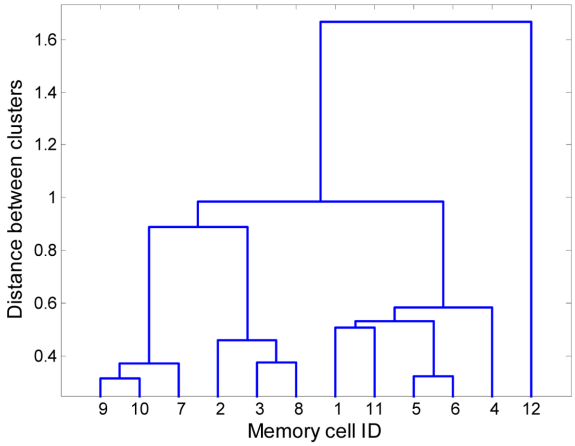

Fig. 12 The Dendrogram of antibody memory cells for four data patterns

clusters is 3, which locates in the curved region of the evaluation graphs. The classified memory cells shown in Fig. 10 form an internal image of the three input data patterns in Fig. 6. In Fig. 10, three distinct clusters of memory cells map three input data patterns. The number of memory cells in each cluster is less than the number of feature vectors in the original data set.

In the second test, four data patterns: Normal, Damage 1, Damage 3, and Damage 4 were used. As described in Section 3, each data pattern generates 54 feature vectors. A total number of 54*4 =216 feature vectors were generated for four data patterns. These feature vectors are shown in Fig. 11. The feature vectors in Fig. 11 were inputted to the INEPR algorithm to generate antibody memory cells using the same set of parameter values in test 1. The dendrogram of antibody memory cells for the four data patterns is shown in Fig. 12. The result of the L method in Fig. 13 shows that the best number of clusters in the second test is 4. The generated internal image for the four input data patterns demonstrates that the INEPR algorithm is able to recognize the new data pattern as shown in Fig. 14.

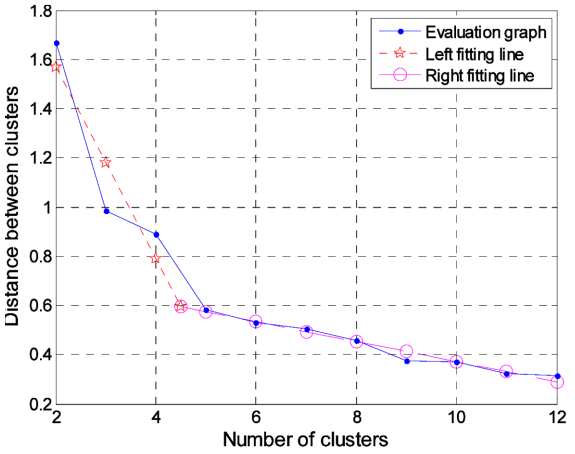

Fig. 13 The best number of clusters for antibody memory cells in Fig. 12

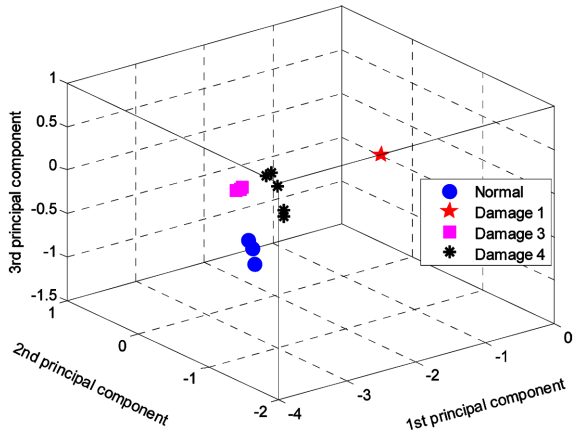

Fig. 14 Generated internal image for four input data patterns in Fig. 11

8. Performance analysis

This section discusses the impact of model parameters on the number of memory cells and the impact of cluster dissimilarity, number of input data points, and the dissimilarity measure on the performance of the *L* method.

8.1 The impact of the aiNet parameters on the number of antibody memory cells

The aiNet model provides a number of parameters that allow users to manipulate to achieve specific goals. These parameters include suppression threshold (σ_s), natural death threshold (σ_d), number of antibodies (n) selected for clone and mutation, and ($\zeta\%$) of mutated antibodies selected as candidate memory cells. In this subsection, the impact of these parameters on the number of antibody memory cells is discussed. Data patterns used in the performance analysis were Normal, Damage 1, and Damage 3 patterns.

Fig. 15 shows the relation of the number of memory cells with the suppression threshold σ_s. In this figure, σ_s changes from 0.05 to 1.25 with a step size of 0.01. The values of rest parameters are chosen as: $n = 6$, $\zeta\% = 20\%$, and $\sigma_d = 7.3$. Fig. 15 illustrates that the number of memory cells is very sensitive to the value σ_s within the range of [0.05, 0.3]. When the value σ_s is greater than 0.3, the number of memory cells is very small. This result may be due to the fact that the dissimilarities of most antibody clones and memory cells in the same pattern fall into the range of [0.05, 0.3]. Within this range, increase the value σ_s dramatically reduces the number of memory cells. The larger the value σ_s, the higher the chance of eliminating antibody clones and memory cells in the clonal suppression and network suppression processes. Contrarily, the smaller the value σ_s, the less the chance of eliminating antibody clones and memory cells. Although, the relative large value σ_s will result in a small number of memory cells, the generated antibody memory cells may lack sufficient information to represent each data pattern. When the value σ_s is small, the resultant memory cell set contains a large number of memory cells. This will cause resource waste and increase computational overhead. The tradeoff decision should be made for both the completeness of representative information and the reasonable number of memory cells.

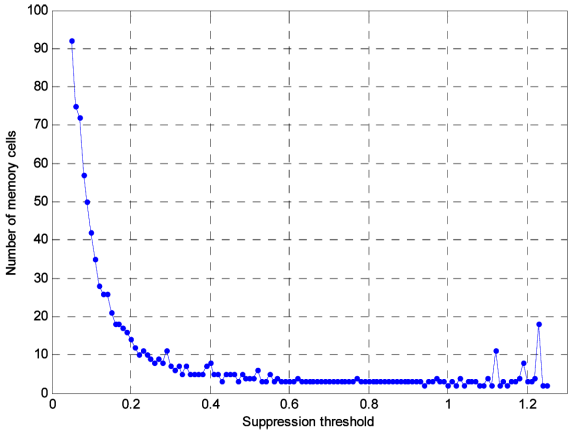

Fig. 15 Number of memory cells vs. suppression threshold σ_s

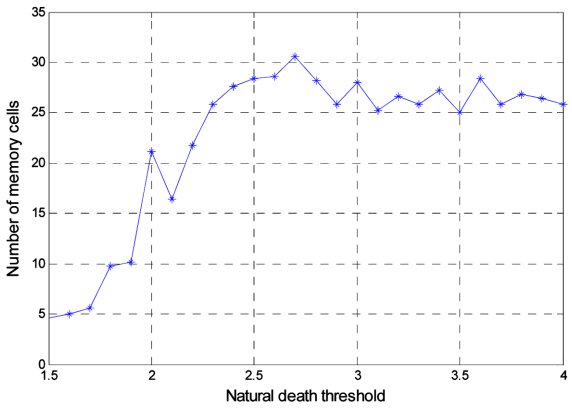

Fig. 16 Number of memory cells vs. natural death threshold σ_d

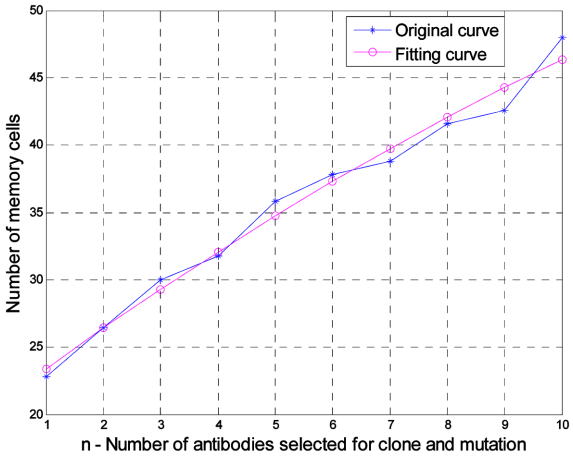

Fig. 17 Number of memory cells vs. n

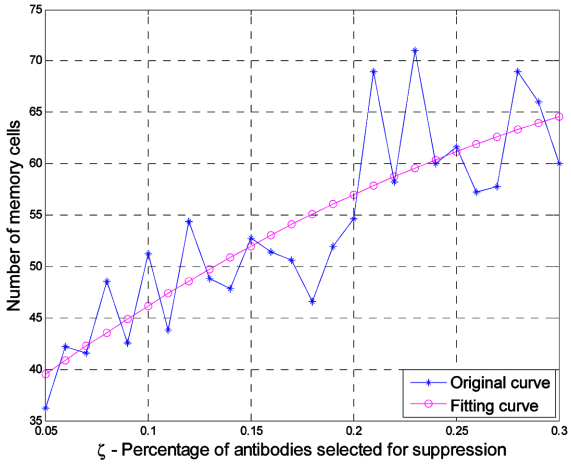

Fig. 18 Number of antibody memory cells vs. ζ

Fig. 16 shows the impact of the natural death threshold σ_d on the number of memory cells. The values of other parameters in Fig. 16 are $n = 6$, $\zeta\% = 20\%$, and $\sigma_s = 0.17$. When the value of σ_d is within the range of 1.5-2.7, the number of memory cells increases quickly as the σ_d value increases. This is because raising the natural death threshold will reduce the number of memory cells that are removed in the natural death elimination. When the value of σ_d is greater than 3, the change of σ_d value does not have a significant impact on the number of memory cells.

Fig. 17 shows the impact of the parameter n on the number of memory cells. In Fig. 17, the values of the parameters σ_s, σ_d, and $\zeta\%$ are chosen as: $\sigma_s = 0.15$, $\sigma_d = 3$, and $\zeta\% = 9\%$. For each value of n, the number of memory cells is the average of five runs. Fig. 17 illustrates that the number of memory cells increases as the value of n increases. The larger value of n means more antibodies being selected for clone and mutation. This increases the diversity of antibody clones and reduces the number of antibody clones and memory cells being eliminated in the clonal suppression and network suppression. Fig. 18 illustrates the relation between the number of memory cells with

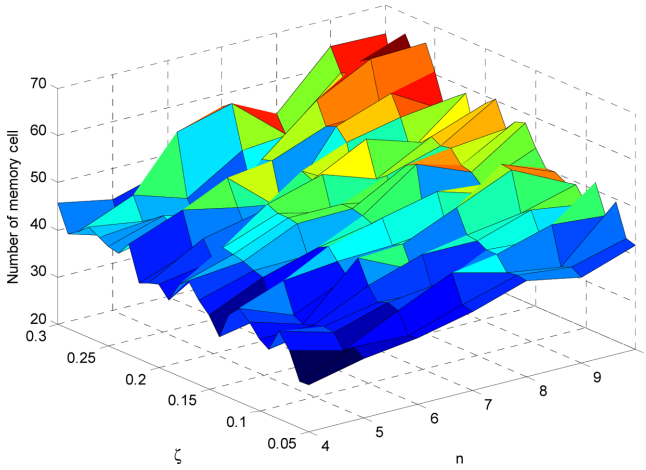

Fig. 19 Number of memory cells vs. n and ζ

the parameter ζ. The values of other parameters in this figure are chosen as: $\sigma_s = 0.15$, $\sigma_d = 3$ and $n = 9$. For each value of ζ, the number of memory cells is the average of five runs. Although the number of memory cells fluctuates when the value of ζ changes, the trend shows the increase of the number of memory cells for large ζ values. This result is reasonable because the large ζ value causes more antibody clones being recruited in the natural death elimination and clonal suppression processes. The combinational impact of the parameters n and ζ on the number of memory cells is shown in Fig. 19. From this figure, similar conclusions can be drawn as in Figs. 17 and 18.

8.2 The impact of the input data size and the hierarchical clustering method on the performance of L method

This subsection examines the impact of L method input data set size and the hierarchical clustering method on the performance of L method. To test the effect of input data size, different sizes of data sets are inputted into the L method algorithm. For example, when the input data set includes all feature vectors of the Normal, Damage 1, and Damage 3 patterns, the calculated evaluation graph and two curve fitting lines are shown in Fig. 20. The best number of clusters is determined to be 10 by the L method algorithm because these two fitting lines intersect at 10. This decision does not match to the right number of clusters. The correct number of clusters should be 3. If we reduce the number of feature vectors using artificial immune network model (aiNet), the resulting antibody memory cells for three patterns are shown in Fig. 7. Re-apply hierarchical clustering algorithm and the L method to these antibody memory cells, the best number of clusters is found to be 3 as shown in Fig. 9. This is the right number of clusters.

A hierarchical clustering algorithm creates a tree for the input data set using a specified method. Methods differ from one another in how they measure the distance between clusters. Commonly available methods include “single” - shortest distance, “complete” - furthest distance, “average” - unweighted average distance, “weighted” - weighted average distance, “centroid” - centroid distance, “median” - weighted center of mass distance, and “ward” - inner square distance. Using different methods in a hierarchical clustering algorithm will result in different output clusterings, as a result, affect the

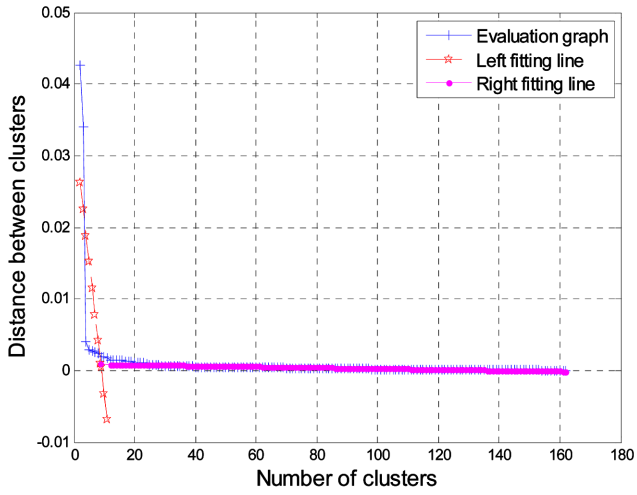

Fig. 20 L method with large number of data points

Table 2 The impact of clustering method on the success rate in determining the number of clusters using L method

	Single	Complete	Average	Weighted	Centroid	Median	Ward
Success rate (%)	93.5	77.8	94.4	89.9	96.6	93.8	46.1

decision of the L method since the outputs of a hierarchical clustering algorithm are the inputs of the L method. Table 2 shows a comparison of the success rate of the L method when different clustering methods are used. Three data patterns used in this test are Normal, Damage 1, and Damage 3 patterns, and the point distance is evaluated by the Euclidean distance.

8.3 The impact of cluster dissimilarity on the performance of L method

The dissimilarity of sensor data clusters has a significant impact on the performance of L method in determining the number of clusters, as a result, the success rate of the emergent pattern recognition. The larger the dissimilarity among clusters, the better the separation of clusters is in a multidimensional space. Fig. 21 shows the feature vectors of 9 data patterns simulated in the experimental tests of the benchmark structure. Table 3 lists the success rate of L method in determining the number of clusters. Two types of point distances: "Correlation" and "Cosine" and two hierarchical clustering methods: "Average" and "Centroid" are used in the performance evaluation of L method. In test 1, five data patterns: Normal (N), Damage 1 (D1), Damage 2 (D2), Damage 3 (D3), and Damage 4 (D4) are evaluated. Since these 5 data patterns have a good separation (dissimilarity) from each other as shown in Fig. 21, the success rate of L method is 90% in Case 1, 89% in Case 2, 90% in Case 3,

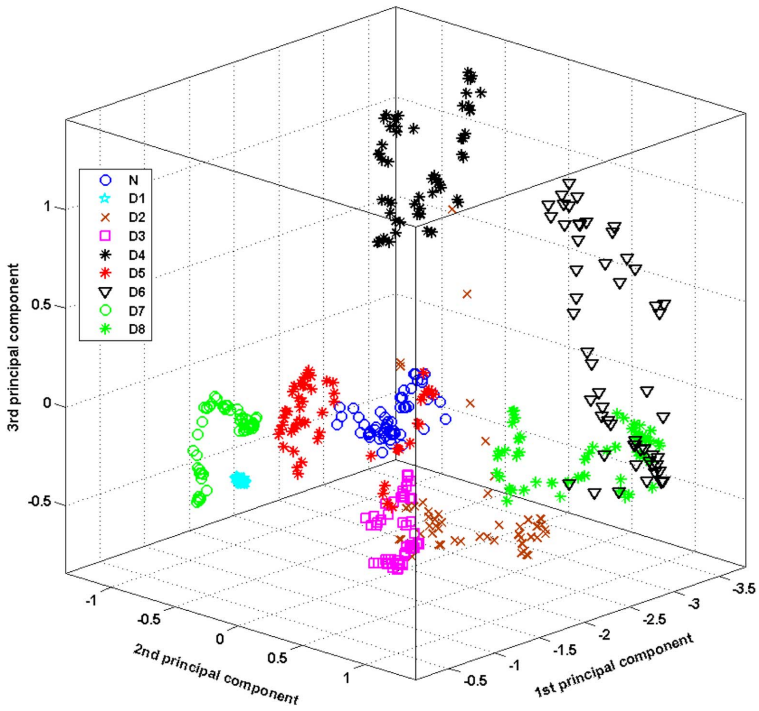

Fig. 21 The feature vectors of 9 data patterns simulated in the experimental tests

Table 3 The impact of cluster dissimilarity on the success rate of L method in determining the number of clusters

Test	Data patterns in test	Success rate correlation - average, case 1 (%)	Success rate correlation - centroid, case 2 (%)	Success rate cosine - average, case 3 (%)	Success rate cosine - centroid, case 4 (%)
1	N, D1, D2, D3, D4	90	89	90	89
2	N, D1, D2, D3, D4, D7, D8	83	72	84	74
3	N, D1, D2, D3, D4, D5, D6, D7, D8	14	9	17	9

and 89% in Case 4, respectively. The success rate is the result of 100 runs. In test 2, two more data patterns: Damage 7 (D7) and Damage 8 (D8) are included. Since Damage 8 has some overlap with Damage 2, the success rates of the L method in test 2 are lower than those in test 1. The increased data points input to the L method may also impact the success rate. In test 3, Damage 5 (D5) and Damage 6 (D6) are also added into the evaluation. From Fig. 21 we can see that Damage 5 and Damage 6 overlap with several data patterns. As a result, the success rates of the L method are dropped significantly as shown in Table 3.

8.4 The impact of the dissimilarity measures on the decision of the number of clusters

This subsection investigates the impact of dissimilarity measures, including both cluster-to-cluster and point-to-point, on the decision of the best number of clusters of the antibody memory cells clustering. Different types of distance measures are applied to the clustering generation and the evaluation graph calculation. Figs. 22, 9 and 23 show the L method results when the Correlation distance, Euclidean distance, and Seuclidean distance are used for the point-to-point dissimilarity measure. The input data used for this test are three data patterns: Normal, Damage 1, and Damage 3 patterns. For the Correlation distance or Euclidean distance, the best number of clusters is decided to be 3, which is consistent with the original data set. For the Seuclidean distance, the best number of clusters is decided to be 4 instead of 3. These test results show that the type of point-to-point

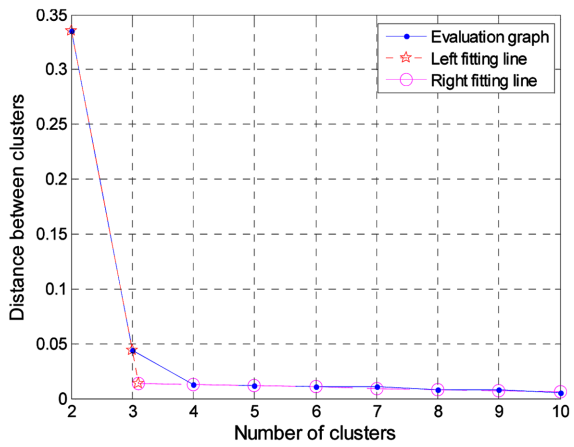

Fig. 22 The best number of clusters with Correlation distance

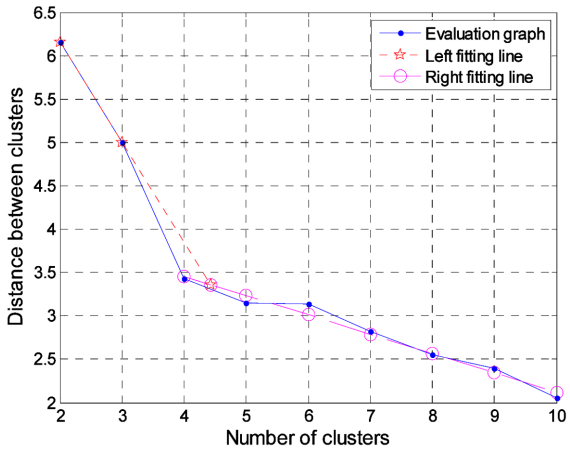

Fig. 23 The best number of clusters with Seuclidean distance

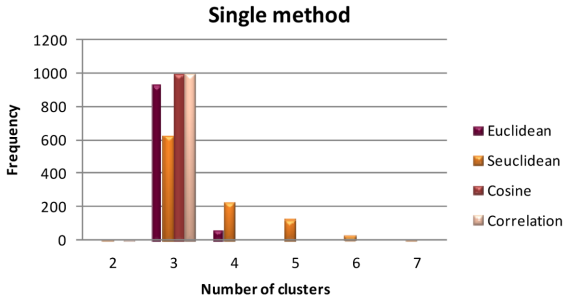

Fig. 24 The distribution of the number of clusters for different distance measures in single link clustering algorithm

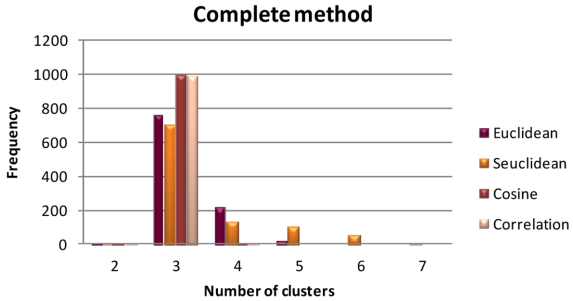

Fig. 25 The distribution of the number of clusters for different distance measures in complete link clustering algorithm

distance measures should be carefully chosen for a specific application.

To find this impact statistically, four types of point-to-point distance measures listed in Eq. (9) to Eq. (12) were applied to the INEPR model. For each type of distance measure, the INEPR model was run 1000 times using same input data. The distributions of the best number of clusters are shown in Figs. 24 and 25. Fig. 24 shows the distribution of L method results for different point-to-point distance measures when the cluster-to-cluster distance measure uses single link method (defined by Eq. (7)), while the uses the complete link method (defined by Eq. (8)). From these two figures, the Correlation distance or Cosine distance performs better comparing to the Euclidean distance and Seuclidean distance for both single link algorithm and complete link algorithm in our application. For the Cosine or Correlation distance shown in Fig. 24, 999 out of 1000 runs obtain the right results of the best number of clusters. The performance of the Euclidean distance is better than the Seuclidean distance and worse than the Cosine or Correlation distance, 932 out of 1000 are correct results. Among four types of distance measures, the Seuclidean distance is the worst one. The success rate on determining the best number of clusters is only 62.3%. Out of 1000 runs, the times corresponding to 3 clusters, 4 clusters, 5 clusters, 6 clusters, 7 clusters, and one cluster are 623 times, 228 times, 123 times, 23 times, 2 times, and one time, respectively.

9. Conclusions

This paper presents a computational model for the emergent pattern recognition. The INEPR algorithm is based on the immune network theory and hierarchical clustering algorithms. The goal of the INEPR is to dynamically generate an internal image mapping to the input data patterns without the need of specifying the number of clusters in advance. This goal is achieved through the construction of a network of antibody memory cells, generation of a hierarchy of antibody memory cells using hierarchical clustering algorithms, determining the best number of clusters for the antibody memory cells using evaluation graphs and L method, and classifying the antibody memory cells to form an internal image for the input data patterns. The INEPR model has been tested using the experimental data of a benchmark civil structure. The test results illustrate that the INEPR model is able to recognize new damage patterns.

The impact of model parameters on the number of memory cells and the impact of cluster dissimilarity, number of input data points, and the dissimilarity measure on the performance of the L

method in determining the number of clusters have also been investigated. The performance analysis shows that the suppression threshold σ_s has a significant impact on the number of memory cells when its value is within the range of [0.05, 0.3]. In addition, increasing the values of σ_d within the range of 1.5-2.7, n, and ζ% will increase the number of memory cells. The investigation also shows that the cluster dissimilarity, number of input data points, and different dissimilarity measures will result in different success rate in determining the number of clusters for the internal image.

Acknowledgements

This research is supported by the National Science Foundation under Grant No. 1049294 and Michigan Tech Research Excellence Fund. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring institutions. The authors would like to thank Wenjia Liu for his contributions to the simulation work described in this article.

References

Beck, J., Bernal, D., Caicedo, J., Dyke, S., Forsyth, M., Lord, J.F. and Mizumori, A. (2002), "Description of Experimental Data from Structural Health Monitoring of UBC Test Structure", file:///C:/users/Old/bchen_new/visiting%20scholar/Chuanzhi/zcz/Matlab%20Shared%20data/ASCEExperimentData/readme.html.

Castiglione, F., Motta, S. and Nicosia, G. (2001), "Pattern recognition by primary and secondary response of an Artificial Immune System", *Theor. Biosci*., **120**(2), 93-106.

Chang, P.C., Flatau, A. and Liu, S.C. (2003), "Review paper: health monitoring of civil infrastructure", *Struct. Health Monit*., **2**(3), 257-267.

Chen, B. and Zang, C. (2009), "Artificial immune pattern recognition for structure damage classification", *Comput. Struct*., **87**(21-22), 1394-1407.

Chen, B. and Zang, C. (2011), "A hybrid immune model for unsupervised structural damage pattern recognition", *Expert Syst. Appli*., **38**(3), 1650-1658.

Cheong, M.Y. and Lee, H. (2008), "Determining the number of clusters in cluster analysis", *J. Korean Statistic. Soc*., **37**(2), 135-143.

Chiu, T., DongPing, F., Chen, J., Yao, W. and Jeris, C. (2001), "A robust and scalable clustering algorithm for mixed type attributes in large database environment", KDD-2001*, Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 263-8|xv+483.

Dasgupta, D., KrishnaKumar, K., Wong, D. and Berry, M. (2004), "Negative selection algorithm for aircraft fault detection", Artificial Immune Systems", *Proceedings of the 3rd International Conference, ICARIS 2004*, Lecture Notes in Comput. Sci., **3239**, 1-13.

De Castro, L.N. and Timmis, J. (2002), Artificial Immune Systems: A New Computational Intelligence Approach Springer.

De Castro, L.N. and Von Zuben, F.J. (2001), aiNet: An Artificial Immune Network for Data Analysis. Data Mining: A Heuristic Approach. H.A. Abbass, R.A. Sarker and C.S. Newton, Idea Group Publishing: 231-259.

Freitas, A.A. and Timmis, J. (2007), "Revisiting the foundations of artificial immune systems for data mining"*, IEEE T. Evolut. Comput*., **11**(4), 521-540.

Hart, E. and Timmis, J. (2008), "Application areas of AIS: the past, the present and the future", *Appl. Soft Comput*., **8**(1), 191-201.

Jerne, N.K. (1974), "Towards a network theory of immune system", *Ann. Immunol*., **C125**(1-2), 373-389.

Johnson, E.A., Lam, H.F., Katafygiotis, L.S. and Beck, J.L. (2000), "A benchmark problem for structural health monitoring and damage detection", *Proceedings of the 14th Engineering Mechanics Conference*, Austin, Texas.

Kothari, R. and Pitts, D. (1999), "On finding the number of clusters", *Pattern Recogn. Lett*., **20**(4), 405-416.

Lanaridis, A., Karakasis, V. and Stafylopatis, A. (2008), "Clonal selection-based neural classifier", *Proceedings of the 8th International Conference on Hybrid Intelligent Systems (HIS)*, 655-60.
Li, M.J., Ng, M.K., Cheung, Y.M. and Huang, J.Z. (2008), "Agglomerative fuzzy K-Means clustering algorithm with selection of number of clusters", *IEEE T, Knowl. Data En*., **20**(11), 1519-1534.
Lu, K.C., Loh, C.H., Yang, Y.S., Lynch, J.P. and Law, K.H. (2008), "Real-time structural damage detection using wireless sensing and monitoring system", *Smart Struct. Syst*., **4**(6), 759-77.
Nagayama, T., Sim, S.H., Miyamori, Y. and Spencer, B.F. (2007), "Issues in structural health monitoring employing smart sensors", *Smart Struct. Syst*., **3**(3), 299-320.
Negoita, M. (2005), "Artificial immune systems - an emergent technology for autonomous intelligent systems and data mining", Autonomous Intelligent Systems: Agents and Data Mining, *Proceedings of the International Workshop, AIS-ADM 2005*, Lecture Notes in Artificial Intelligence, **3505**, 19-36.
Polat, K., Gunes, S. and Tosun, S. (2006), "Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing", *Pattern Recogn*., **39**(11), 2186-2193.
Qinpei, Z., Hautamaki, V. and Friinti, P. (2008), "Knee point detection in BIC for detecting the number of clusters", Advanced Concepts for Intelligent Vision Systems, *Proceedings of the 10th International Conference*, ACIVS 2008, 664-73.
Salvador, S. and Chan, P. (2004), "Determining the number of clusters/segments in hierarchical clustering/ segmentation algorithms", *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, FL, USA, IEEE Comput. Soc.
Shen, X., Gao, X.Z. and Bie, R. (2008), "Artificial immune networks: Models and applications", *Int. J. Intell. Syst*., **1**(2), 168-176.
Sugar, C.A. and James, G.M. (2003), "Finding the number of clusters in a dataset: An information-theoretic approach", *J. Am. Stat. Assoc*., **98**(463), 750-763.
Sumitro, S. and Wang, M.L. (2005), "Sustainable structural health monitoring system", *Struct. Health Monit*., **12**(3-4), 445-467.
Theodoridis, S. and Koutroumbas, K. (2008), *Pattern Recog.*, Academic Press.
Tibshirani, R., Walther, G. and Hastie, T. (2001), "Estimating the number of clusters in a data set via the gap statistic", *J. R. Stat. Soc. B*., **63**(2), 411-423.
Timmis, J., Andrews, P., Owens, N. and Clark, E. (2008), "An interdisciplinary perspective on artificial immune systems", *Evolution. Intell*., **1**(1), 5-26.
Weng, J.H., Loh, C.H., Lynch, J.P., Lu, K.C., Lin, P.Y. and Wang, Y. (2008), "Output-only modal identification of a cable-stayed bridge using wireless monitoring systems", *Eng. Struct*. **30**(7), 1820-1830.
Zhong, Y.F., Zhang, L.P., Huang, B. and Li, P.X. (2006), "An unsupervised artificial immune classifier for multi/ hyperspectral remote sensing imagery", *IEEE T. Geosci. Remote*., **44**(2), 420-431.

Nomenclature

ag	: a single antigen
ag.f	: the feature vector of an antigen *Ag*
ab	: a single antibody
ab.f	: the feature vector of an antibody *Ab*
A	: a set of antibodies ($A \in R^{N \times P}$)
M	: a set of antibody memory cells ($M \in R^{K \times P}$)
G	: a set of antigens ($G \in R^{M \times P}$)
σ_d	: natural death threshold
σ_s	: suppression threshold