



A hybrid immune model for unsupervised structural damage pattern recognition

Bo Chen^{a,b,*}, Chuanzhi Zang^{a,c}

^a Department of Mechanical Engineering – Engineering Mechanics, Michigan Technological University, 815 R.L. Smith Building, 1400 Townsend Drive, Houghton, MI 49931, USA

^b Department of Electrical and Computer Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA

^c Shenyang Institute of Automation, Chinese Academy of Science, Nanta Street 114, Shenyang, Liaoning 110016, PR China

ARTICLE INFO

Keywords:

Structural health monitoring
Unsupervised structural damage pattern recognition
Fuzzy clustering
Artificial immune pattern recognition

ABSTRACT

This paper presents an unsupervised structural damage pattern recognition approach based on the fuzzy clustering and the artificial immune pattern recognition (AIPR). The fuzzy clustering technique is used to initialize the pattern representative (memory cell) for each data pattern and cluster training data into a specified number of patterns. To improve the quality of memory cells, the artificial immune pattern recognition method based on immune learning mechanisms is employed to evolve memory cells. The presented hybrid immune model (combined with fuzzy clustering and the artificial immune pattern recognition) has been tested using a benchmark structure proposed by the IASC–ASCE (International Association for Structural Control–American Society of Civil Engineers) Structural Health Monitoring Task Group. The test results show the feasibility of using the hybrid AIPR (HAIPR) method for the unsupervised structural damage pattern recognition.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The structural health monitoring (SHM) is a process of observing a structure's dynamic response measurements from a group of sensors, extracting damage-sensitive features from these measurements, and analyzing these features to determine the current state of the structure (Kolakowski, 2007). Due to high instrument and installation costs of wired SHM systems (Sazonov, Janoyan, & Jha, 2004), the wireless sensor-network-based SHM is emerging as a feasible approach since it allows dense sensing through many inexpensive sensor nodes and is easy for deployment and maintenance (Xu et al., 2004). While sensor network approach presents a number of advantages, SHM sensor network systems currently face a number of challenges (Farrar & Worden, 2007). Major challenges in SHM sensor networks include: (1) how can we provide sustainable monitoring and control in an autonomous manner? For complex structures, a monitoring sensor network may consist of hundreds or thousands of sensor nodes and may be deployed in environments that are difficult to access (embedded in physical structures). Given such a deployment size and environment, sensor networks are required to monitor structural changes and perform damage diagnosis autonomously; (2) can we develop adaptable approaches to SHM that are able to dynamically adapt to changing monitoring conditions? Due to resource constraints in sensor net-

works, a SHM sensor network that is able to manage its resources effectively under different circumstances is critical; (3) how can we detect and identify structural damages in an active way? The passive monitoring of structures by continuously gathering real-time structural data causes data transmission problem due to limited bandwidth and power available in wireless sensor networks; (4) how can we establish an unsupervised damage diagnosis methodology?

The natural immune system is an effective defense mechanism for a given host against infections (De Castro, 2006). From a pattern recognition perspective, the most appealing characteristic of the immune system is its immune cells (B-cells and T-cells) carrying surface receptors that are capable of recognizing and binding antigens. The antibodies are soluble forms of the B-cell receptors that are released from the B-cell surface to cope with the invading non-self antigen. Antibodies bind to antigens leading to their eventual elimination by other immune cells (De Castro & Timmis, 2002). When a B-cell encounters a nonself antigen that has sufficient affinity with its receptors, coupled with a stimulation signal from T-cells, the B-cell is activated. It, therefore, undergoes a clonal selection that increases the number of the activated B-cell and the diversity of the antibody set. The generated B-cells with high antigenic affinities are selected to become memory cells that remain in the immune system for months or years. The first exposure of a B-cell to a specific type of antigen triggers the *primary response* in which the pattern is recognized and the memory is developed (Castiglione, Motta, & Nicosia, 2001). The memory cell for a specific antigen that had stimulated in the primary response will respond to previously recognized antigen in a much shorter time

* Corresponding author at: Department of Mechanical Engineering – Engineering Mechanics, Michigan Technological University, 815 R.L. Smith Building, 1400 Townsend Drive, Houghton, MI 49931, USA.

E-mail address: bochen@mtu.edu (B. Chen).

comparing to a newly activated B-cell (Carter, 2000). The novel characteristics of the immune system have inspired the development of artificial immune systems for various applications (Dasgupta, 2006; Hart & Timmis, 2008). The major application areas include data mining (Freitas & Timmis, 2007), pattern recognition (Watkins, Timmis, & Boggess, 2004; Zhong, Zhang, Gong, & Li, 2007), fault diagnosis (Dasgupta, KrishnaKumar, Wong, & Berry, 2004; Taylor & Corne, 2003), and medical classification problems (Polat, Gunes, & Tosun, 2006).

Due to the similarities of the human immune system and the SHM systems, the artificial immune system (AIS) model could be used as the basis for SHM strategies (Chen, 2009). This approach is well suited to address SHM problems because: (1) the AIS-based SHM is autonomous. The AIS-based SHM systems can automatically manage structural monitoring tasks by dynamically generating and distributing the mobile monitoring agents; (2) the AIS-based SHM is adaptive. The amount and type of molecules of the immune system can adapt themselves to the antigenic challenges via clonal selection (Cesana et al., 2005). The adaptive mechanism of the natural immune system has great value in SHM sensor networks. The selective generation of mobile monitoring agents is essential for producing large enough amount of specialized mobile monitoring agents in resource-constrained sensor networks (Negota, 2005); (3) the AIS-based SHM is active. The concept of active dispatching mobile monitoring agents (mimicking B-cells) helps the distribution of specialized monitoring agents to the sites where they are needed; and (4) the immune learning and memory mechanisms help the development of unsupervised damage detection and pattern recognition, which is desirable in SHM.

This paper presents a hybrid immune model for unsupervised structural damage pattern recognition based on the fuzzy clustering technique and the artificial immune pattern recognition (HAIPR). The fuzzy clustering (FC) algorithm is employed to generate initial memory cells for damage patterns. These initial memory cells are then evolved by an immune learning process to improve the quality of memory cells to represent damage patterns. The presented unsupervised structural damage pattern recognition algorithm has been tested using a benchmark structure (Structural Health Monitoring Benchmark Problem) proposed by the IASC-ASCE (International Association for Structural Control-American Society of Civil Engineers) Structural Health Monitoring Task Group. The test results show the feasibility of using the HAIPR method for the unsupervised structural damage pattern recognition. The rest of the paper is structured as follows. Section 2 presents the algorithm design of the HAIPR approach. Section 3

describes how to use the HAIPR method for the unsupervised damage pattern recognition for the IASC-ASCE benchmark structure and shows the validation results. Section 4 discusses the impact of the system's parameters on the performance of the HAIPR unsupervised pattern recognition and the comparison of the HAIPR method with conventional classification algorithms. Section 5 concludes the presented work.

2. The HAIPR approach for unsupervised structural damage pattern recognition

2.1. The HAIPR approach

In unsupervised structural damage pattern recognition, the pattern information of the training data is not available. The presented HAIPR unsupervised pattern recognition method employs fuzzy clustering algorithms to establish the initial representative for each pattern of the training data. The representative for each pattern generated by the fuzzy clustering algorithms, however, includes limited information. For example, the fuzzy clustering algorithms use one point in a multidimensional space to represent each cluster (pattern) for a compact data set. To obtain more informative pattern representative (memory cells) and provide the evolution capability, the artificial immune pattern recognition method is employed to improve the quality of memory cells for each damage pattern.

Fig. 1 shows the major components of the HAIPR algorithm. The measurement data from multiple sensors are compressed from n -dimensional space (n sensors) into one dimensional space by the Principal Component Analysis (PCA) algorithm. The features of the compressed time series sensor data are extracted from the auto regression (AR) model of the time series. The initial memory cells for sensor data patterns are generated by the fuzzy clustering algorithm. These initial memory cells are also used to classify the training data into a specified number of patterns based on the nearest neighbor criterion. The classified training data are then used to evolve memory cells in an immune learning process based on clonal selection principle. The evolved memory cells are then used for the structural damage pattern recognition.

2.2. Feature extraction from sensor data

In structural damage pattern recognition, damage patterns are represented by feature vectors extracted from dynamic response data of a structure. A feature vector consists of a number of

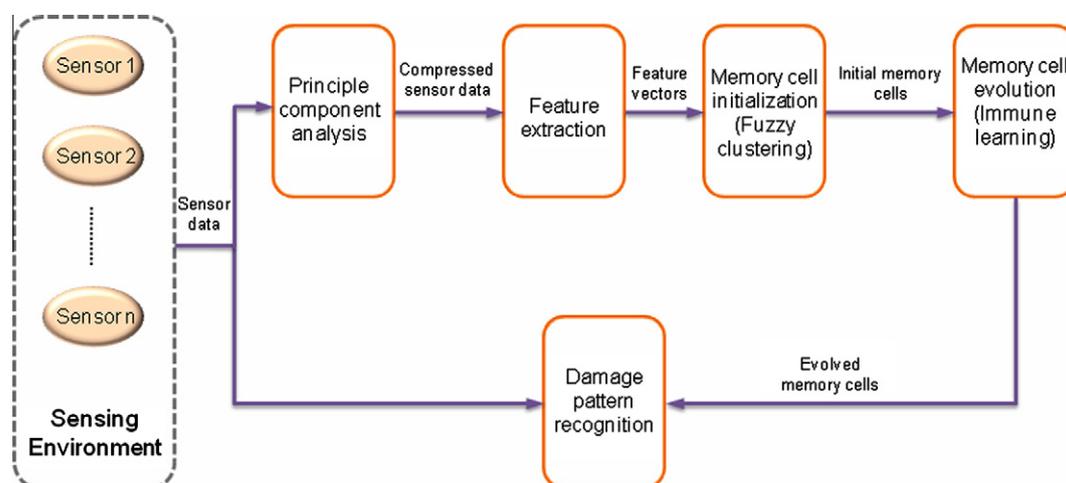


Fig. 1. Overview of the HAIPR approach.

features. The feature selection is critical to the success of the damage pattern recognition (Theodoridis & Koutroumbas, 2008). Feature selection is the process to identify the measurable quantities that make damage patterns distinct from each other. In the HAIPR algorithm implementation, following steps are designed to process raw sensor data and extract damage-sensitive feature vectors. First, the time series sensor data are normalized using the mean and standard deviation. Let matrix $Z = (z_{ij})_{m \times n}$ denote the time series of sensor data, where each row is corresponding to the n number of data generated by one sensor and each column is the sensor data collected by the m sensors at a given time. Let $\bar{z}_i = (z_{i1}, z_{i2}, \dots, z_{in})$, $i = 1, 2, \dots, n$ denote the i th row of the matrix Z , which is the sensor data of the i th sensor. The standardized sensor data $Y = (y_{ij})_{m \times n}$ can be calculated by $y_{ij} = \frac{z_{ij} - \mu_i}{\sigma_i}$, $j = 1, 2, \dots, n$, where y_{ij} is the standardized value of the corresponding value of z_{ij} , μ_i and σ_i are the mean and standard deviation of the time series \bar{z}_i . Second, time series sensor data sets from multiple sensors are reduced to lower dimensions by the Principal Component Analysis (PCA) (Pearson, 1901) method for extracting a feature vector for a local area. In our implementation, the multiple data sets from m number of sensors are compressed into one data set. It means that all the sensor data are projected onto the principal component that has the biggest eigenvalue. Let Ψ denote the $m \times m$ covariance matrix of the standardized time series signals Y . The matrix Ψ can be calculated by $\Psi = \frac{1}{n-1} Y Y^T$. Let λ_i and \bar{v}_i denote the i th eigenvalue and eigenvector of matrix Ψ , respectively. So, $\Psi \bar{v}_i = \lambda_i \bar{v}_i$, where eigenvector \bar{v}_i is called the principal component. Let \bar{v}_1 denote the vector that is corresponding to the biggest eigenvalue. The relationship between the compressed data $\bar{x} = (x_1, x_2, \dots, x_n)$, \bar{v}_1 and Y is $\bar{x} = \bar{v}_1^T Y$. Third, the feature vector for a local area is extracted from the compressed time series. The AR algorithm is chosen to model the compressed time series data. Each compressed time series x is fitted into an AR model of order p as shown in Eq. (1).

$$x_k = \sum_{i=1}^p a_i x_{k-i} + r_k \quad k = p + 1, \dots, n \quad (1)$$

where the r_k is the residual between the sensor data and the AR model value. The order of the AR model is chosen based on the Akaike's information criterion (AIC). An AIC is a measure of the

goodness of fit of an estimated statistical model. Given a data set, the model having the lowest AIC is the best model. The vector $\alpha = (a_1, a_2, \dots, a_p)^T \in R^p$, is selected as the **feature vector** of the time series. The feature vector is calculated using the Least Square (LS) method. Rewrite Eq. (1) in following format:

$$A\alpha = b \quad (2)$$

where

$$A = \begin{bmatrix} x_p & x_{p-1} & \dots & x_1 \\ x_{p+1} & x_p & \dots & x_1 \\ \dots & \dots & \dots & \dots \\ x_{n-1} & x_{n-2} & \dots & x_{n-p-1} \end{bmatrix}, \quad b = \begin{bmatrix} x_{p+1} \\ x_{p+2} \\ \vdots \\ x_n \end{bmatrix} \quad (3)$$

The feature vector α can be calculated as follows:

$$\alpha = (A^T A)^{-1} A^T b \quad (4)$$

The effectiveness of the AR-model-based feature vectors is tested using the experimental data of the benchmark structure (Structural Health Monitoring Benchmark Problem) proposed by the IASC-ASCE SHM Task Group. The feature vectors of four damage patterns and the normal pattern are visualized using the Sammon nonlinear mapping algorithm (Sammon, 1969) as shown in Fig. 2. Although some overlapping among different patterns exists, the AR-model-based feature vectors are able to distinguish these five data patterns to a certain extent.

2.3. Initial memory cell generation

To generate the initial memory cell for each data pattern and cluster training data into a specified number of patterns, the fuzzy clustering method is employed. Since the damage-sensitive feature vectors are compact clusters as shown in Fig. 2, a point representation is used to represent each pattern. A fuzzy k -clustering of $A = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ is defined by a set of functions $u_j: A \rightarrow [0, 1]$, $j = 1, 2, \dots, k$, where $\alpha_i \in R^p$, $i = 1, 2, \dots, N$ is the feature vector of the training data in p -dimensional real value space and N is the number of feature vectors. Let $\theta_j \in R^p$ denote the parameterized representative of the j th cluster, $\theta \equiv [\theta_1^T, \theta_2^T, \dots, \theta_k^T]^T$; U denote an $N \times k$ matrix whose (i, j) element equals to $u_j(\alpha_i)$; and

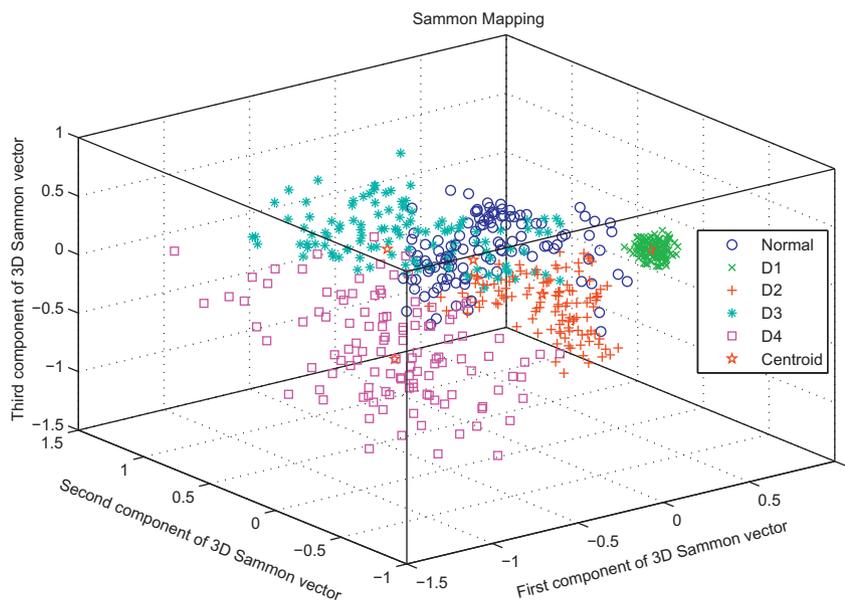


Fig. 2. The AR-model-based feature vectors for five data patterns.

$d(\alpha_i, \theta_j)$ denote the dissimilarity between α_i and θ_j . For the point representative case, the dissimilarity $d(\alpha_i, \theta_j)$ could be any type of distance between two points. To achieve the goal of the estimation of θ that best characterizes the clusters underlying \mathcal{A} , the fuzzy clustering algorithm is derived by minimizing the cost function in Eq. (5) with respect to θ and U , subject to the constraints $\sum_{j=1}^k u_{ij} = 1, i = 1, 2, \dots, N$, where $u_{ij} \in [0, 1], i = 1, 2, \dots, N, j = 1, 2, \dots, k$ and $0 < \sum_{i=1}^N u_{ij} < N, j = 1, 2, \dots, k$. The parameter $q(>1)$ is called a fuzzifier.

$$J(\theta, U) = \sum_{i=1}^N \sum_{j=1}^k u_{ij}^q d(\alpha_i, \theta_j) \quad (5)$$

The outputs of the fuzzy k -means are the point representative of patterns (clusters). These points are used as initial memory cells for k patterns. These initial memory cells are also used to classify the training data by using the nearest neighbor criterion. Given a training data, the distance to the memory cells are calculated. The training data is classified to the pattern with whose memory cell the training data has the shortest distance. The classified training data are then used in the memory cell evolution process for improving the quality of the memory cells.

2.4. Memory cell evolution using immune learning

The initial memory cell set (a collection of memory cells for all the data patterns) generated by the fuzzy clustering algorithm only has one memory cell for each pattern. To improve the quality of memory cell set, the AIPR method (Chen & Zang, 2009) is used to evolve the memory cell set. The AIPR algorithm in Chen and Zang (2009) is based on the CLONALG algorithm in De Castro and Von Zuben (2002) and the AIRS in Watkins et al. (2004). The antibody set evolution is similar to the CLONALG. The memory cell set update, however, is specifically designed to obtain better representative for each damage pattern. For example, the memory cell replacement threshold defined in Chen and Zang (2009) is effective to improve the pattern recognition success rate. The evolution of the memory cell set includes two sub-processes: the evolution of the antibody set and the update of the memory cell set. The flow chart of the memory cell set evolution process is shown in Fig. 3. The training data clustered by the fuzzy clustering algorithm and nearest neighbor criterion are used to stimulate this process. The initial antibody set is generated by the random selection of antibodies from the classified training data.

2.4.1. Evolution of the antibody set

The stimulation of the antibody set by an invading antigen (a training data) will cause the evolution of the antibody set. The description of the antibody set evolution algorithm for each antigen stimulation is given in Table 1. For a training antigen ag , the affinity between the antigen and each antibody ab that is in the same pattern as the antigen is calculated. Let $ab \cdot f = \beta = (\beta_1, \beta_2, \dots, \beta_p)^T \in R^p$ and $ag \cdot f = \gamma = (\gamma_1, \gamma_2, \dots, \gamma_p)^T \in R^p$ denote the feature vectors of an antibody ab and the antigen ag , respectively. The affinity between an antibody and the antigen is defined as (6).

$$aff(ab, ag) = 1 - \frac{1}{2} dist(\beta, \gamma) \quad (6)$$

where $dist(\beta, \gamma)$ is the Euclidian distance between the feature vectors β and γ . The probability that an antibody ab is cloned depends on its affinity with the antigen. The number of the cloned antibodies, $CloneNumber$, depends on the clonal rate CR and the clonal value CV . The CR is an integer value used to control the number of antibody clones allowed for the activated B-cell. The CV is a value that measures the response of a B-cell to an antigen. According to the natural immune system, the higher the affinity, the larger the

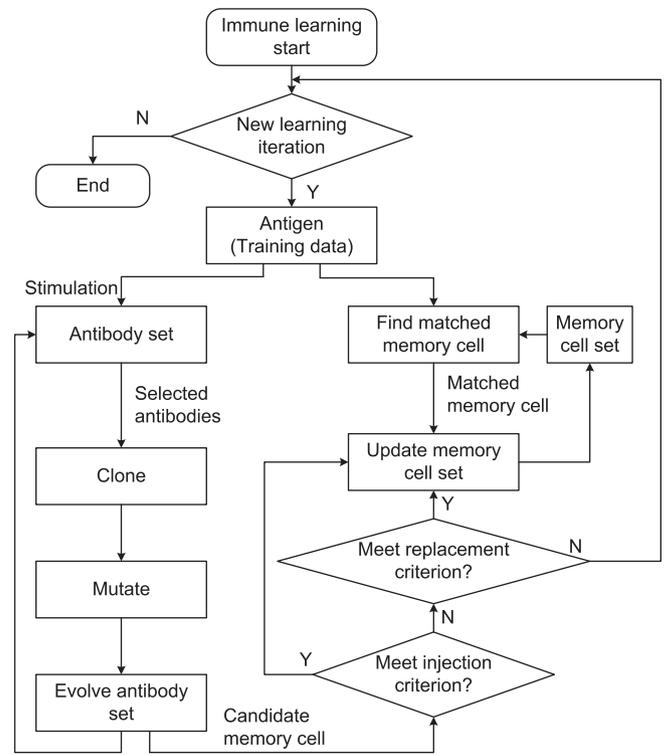


Fig. 3. The evolution of memory cells.

Table 1

The antibody set evolution algorithm for each antigen stimulation.

<p>Begin Input an antigen ag; For each antibody ab which is in the same pattern as ag do Clone antibody ab based on the affinity with the ag; Mutate the cloned antibodies; Keep the mutated antibodies staying within the unit hyper-sphere; Form a new antibody set using top MaxABN number of antibodies; End for-loop Select the highest affinity antibody as the candidate memory cell; End</p>

number of antibodies is cloned. We choose the clonal value being equal to the affinity value. The $CloneNumber$ is then calculated by the Eq. (7).

$$CloneNumber = round(CR * CV) = round(CR * aff(ab, ag)) \quad (7)$$

where $round(\cdot)$ is an operator that rounds its value to the closest integer.

The cloned antibodies undergo a maturation process that increases the diversity of the antibody set. The mutation is performed by mutating the feature vectors of the cloned antibodies as shown in Eq. (8).

$$ab_{mutated} \cdot f = ab \cdot f + MV \times \phi \quad (8)$$

where $ab_{mutated}$ is the mutated antibody and MV is the mutation value. Typically, the higher the affinity is, the smaller the mutation value. In our design, the mutation value MV is defined as $MV = 1 - CV$. In Eq. (8), the vector $\phi = (\phi_1, \phi_2, \dots, \phi_p)^T$ is a randomly generated vector whose dimension is the same as that of the feature vector. Each element ϕ_i is a normal random variable defined by $\phi_i \sim N(0, \sigma^2)$, where $N(0, \sigma^2)$ is a normal random variable with the standard deviation of σ .

The mutated antibodies are added into the antibody subset to which the ag belongs. Since the maximum number of each antibody subset is limited to a predefined threshold, $MaxABN$, the resulting antibody subset is sorted in a descending order according to the affinity values of the antibodies with the given antigen. The top $MaxABN$ number of antibodies is selected to form the evolved antibody set. The rest of antibodies are discarded. The antibody with the highest affinity is chosen as the candidate memory cell $MC_{candidate}$ for the updating of memory cell set.

2.4.2. Update memory cell set

The candidate memory cell generated in the antibody set evolution process is used to update the memory cell set to enhance the representative quality of memory cells for each pattern. The description of the memory cell set update algorithm is given in Table 2. The memory cell update occurs in the following scenarios. First, when the root mean square distance, rms , between the candidate memory cell and the memory cells in the same pattern is greater than a specified threshold value Memory Cell Injection Threshold ($MCIT$), the candidate memory cell is injected into this pattern of memory cells. Let $ag \cdot c$ denote the pattern label of the antigen ag ; $MCS_{ag \cdot c}$ denote the memory cell subset with the same pattern as the given antigen ag ; and $|MCS_{ag \cdot c}|$ denote the total number of the memory cells in the subset $MCS_{ag \cdot c}$. The rms is defined by the Eq. (9).

$$rms = RMS(dist_1, dist_2, \dots, dist_{|MCS_{ag \cdot c}|}) \\ = \frac{1}{\sqrt{|MCS_{ag \cdot c}|}} \sqrt{\sum_{i=1}^{|MCS_{ag \cdot c}|} dist_i^2} \quad (9)$$

where $dist_i = dist(mc_i, MC_{candidate})$, $mc_i \in MCS_{ag \cdot c}$, and $i = 1, 2, \dots, |MCS_{ag \cdot c}|$. If the rms is greater than the threshold $MCIT$, the candidate memory cell is added into the memory cell subset $MCS_{ag \cdot c}$. In the second case (the rms is less than or equal to $MCIT$), the candidate memory cell compares with the matched memory cell. The matched memory cell is the memory cell that has the highest affinity with the given antigen in the same pattern. To find the matched memory cell, the affinity values of the training antigen with the memory cells in the same pattern are calculated. The memory cell that has the highest affinity with the given antigen ag is chosen as the matched memory cell, which is denoted by $MC_{matched}$. When the affinity between $MC_{candidate}$ and the given antigen ag is greater than the affinity between $MC_{matched}$ and antigen ag , the candidate memory cell replaces the matched memory cell if the affinity between $MC_{candidate}$ and $MC_{matched}$ is greater than the Memory Cell Replacement Threshold ($MCRT$), otherwise the candidate memory cell is added into the memory cell subset $MCS_{ag \cdot c}$.

Table 2
The memory cell set update algorithm.

Begin
Input antigen ag ;
Find the matched memory cell;
Calculate the root mean square rms for the candidate memory cell;
If $rms > MCIT$
Add the candidate memory cell into the memory cell set;
Else if (($aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)$) and ($aff(MC_{candidate}, MC_{matched}) > MCRT$))
Replace the matched memory cell by the candidate memory cell;
Else if ($aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)$)
Add the candidate memory cell into the memory cell set
End if
End

3. Using HAIPR approach for unsupervised civil structural damage pattern recognition

The HAIPR method has been tested using a benchmark structure (Structural Health Monitoring Benchmark Problem) proposed by the IASC–ASCE SHM Task Group as shown in Fig. 4. The structural data used in our study are the experimental data. In the experimental setup, a variety of damage cases were simulated by removing braces or loosening bolts in the test structure. The details of the simulated damage patterns are listed in Table 3. The excitation methods used in the structure test are listed in Table 4. In the experimental study, a total of 15 accelerometers were used to measure the acceleration data of the structure, three accelerometers for each level. Fig. 6 shows one of normal acceleration time series measured by the 6th accelerometer, and Fig. 7 shows one of damage pattern 1 acceleration time series measured by the same accelerometer. The acceleration data for each damage pattern or the normal pattern were recorded in a data file. Four damage patterns (configuration 2, 4, 5, and 7) and the normal pattern (configuration 1) were selected to validate the HAIPR unsupervised pattern recognition method. To generate feature vectors for each data pattern, 24,000 points of data in each data file formed 116 of 1000-point time series by advancing 200 points each time. Time series data for 15 accelerometers were reduced to one time series using the PCA method. The information contained in the principal component was investigated by observing the percentage of the largest eigenvalue in the total amount of the eigenvalues. Fig. 5 shows



Fig. 4. Benchmark test structure (Structural Health Monitoring Benchmark Problem).

Table 3
The configurations for simulated damage patterns.

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

Table 4
Excitation methods used in the experimental structure test.

Excitation method	Description
Sledge hammer	Hammer test (DYTRAN Dynapulse 5803A 12 lb sledge hammer)
Electro-dynamic shaker	Shaking induced by the electrodynamic shaker with moving mass – random tests – sine sweep tests (s) – sine sweep tests (s1) – sine sweep tests (s2) Note: not all sine sweep tests were needed for all configurations
Ambient vibration	Ambient test (no excitation beyond ambient vibrations)

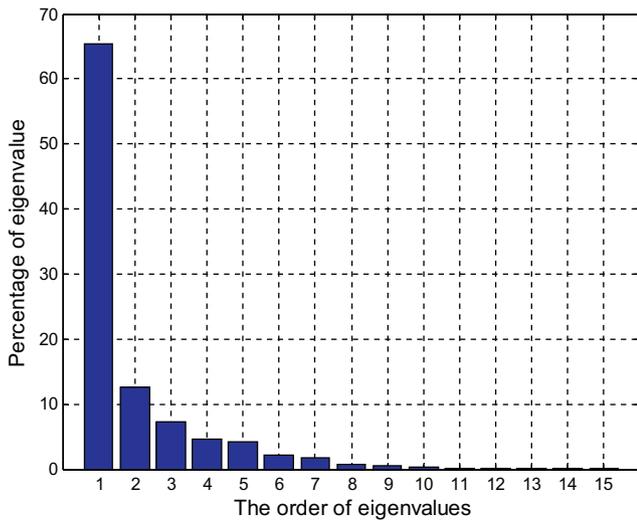


Fig. 5. The percentage of eigenvalue in one of covariance matrix.

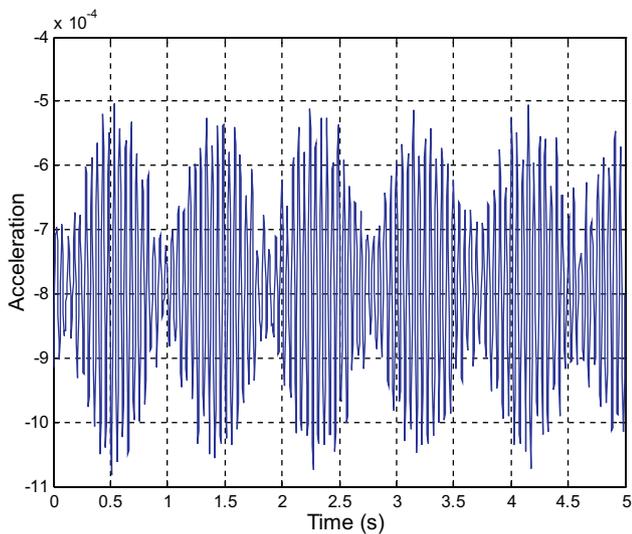


Fig. 6. Normal acceleration data measured by the 6th accelerometer.

the percentage of each eigenvalue in one of covariance matrix generated by 15 accelerometer measurement. The compressed time series sensor data were then fitted into AR models. The AR order was selected to be 20 since the AIC values are small when the AR

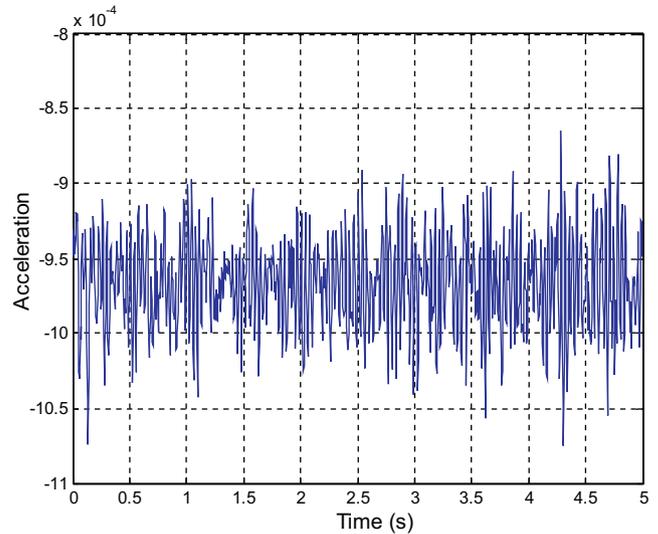


Fig. 7. Damage pattern 1 acceleration data measured by the 6th accelerometer.

Table 5
The assignment of the training data to each pattern.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Percentage of success
Normal	3	4	0	8	101	87.07
Damage 1	0	0	116	0	0	100.00
Damage 2	0	0	0	95	21	81.90
Damage 3	0	84	0	4	28	72.41
Damage 4	90	22	0	0	4	77.59

order is greater than or equal to 20. Since each pattern has 116 feature vectors, a total number of $116 \times 5 = 580$ feature vectors were generated for four damage patterns and the normal pattern.

These 580 feature vectors of experimental data were used to verify the unsupervised pattern recognition algorithm. During the training process, the pattern labels of the 580 feature vectors were erased. The fuzzy clustering algorithm was applied to find the pattern representative for each pattern. Since the feature vectors of the structural data are compact clusters as shown in Fig. 2, a point representative is used to represent each pattern. The outputs of the fuzzy clustering algorithm are five point representatives for five patterns. Each point representative is a vector in R^{20} . The dimension of the point representative is the same as the order of the AR model that is used to represent feature vectors. These five point representatives were used to form the initial memory cell set and also used to classify the training data to five patterns based on the nearest neighbor criterion. The classified training data were then used to improve the quality and increase the number of memory cells through the immune learning. To test the memory cells generated by the immune learning process, the previously created 580 feature vectors were reused with pattern labels. These feature vectors were classified by the memory cells to five clusters. Table 5 shows the number of the feature vectors assigned to each cluster.

To find the statistical distribution of the pattern recognition success rate, the HAIPR algorithm was used to recognize 580 feature vectors for 100 times. The resulting distribution of the pattern recognition success rate is shown in Fig. 8. The numbers on the top of each bar stand for the times that the pattern recognition success rate falls into the range indicated on the x-axis. For example, the pattern recognition success rate within the range of 82.83–83.36% occurs 21 times among 100 tests. The pattern recognition success rate is defined as the ratio of data whose pattern are

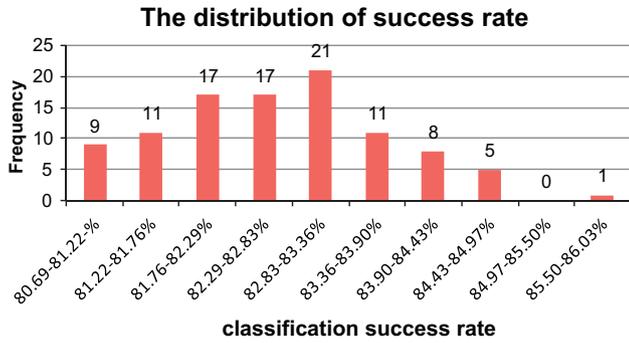


Fig. 8. The distribution of HAIPIR success rate.

correctly recognized to the whole set of test data. The system parameters used in the test are $CR = 8$, $\sigma = 0.5$, $MCRT = 0.95$, $MCIT = 0.60$, and $q = 2$.

4. Performance analysis of the HAIPIR method

The performance of the HAIPIR-based unsupervised structural damage pattern recognition is evaluated. To investigate the effect of the *fuzzifier* parameter, the pattern recognition success rate with different *fuzzifier* values are calculated and plotted in Fig. 9. Three types of distances: Euclidian distance, Diagonal distance, and Mahalanobis distance, are used in Fig. 9. From Fig. 9, we can see that the value of *fuzzifier* has a significant impact on the pattern recognition success rate. For the diagonal distance, the pattern recognition success rate is over 79% when the value of the *fuzzifier* is within the range of 1–3. Further increase the value of the *fuzzifier*, the pattern recognition success rate will gradually drop to 55%. Since the overall performance of the diagonal distance measure is better than other types of distances, the diagonal distance is used for the following analysis and plots. The definition of the diagonal distance is as follows. Assume that there is N number of feature vectors $\vec{\alpha}^i = (\alpha_1^i, \alpha_2^i, \dots, \alpha_p^i)^T \in R^p$, $i = 1, 2, \dots, N$ in p -dimensional space. The N number of feature vectors forms a matrix B as below:

$$B = (\vec{\alpha}^1, \vec{\alpha}^2, \dots, \vec{\alpha}^N) \tag{10}$$

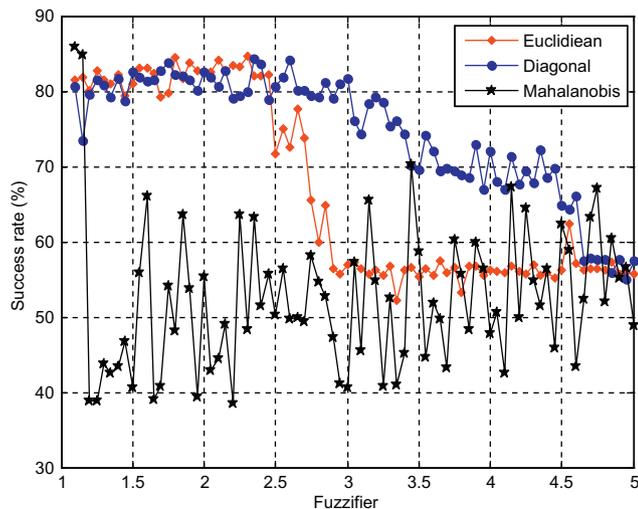


Fig. 9. Comparison of the pattern recognition success rate using different type of distance.

The covariance matrix C of the matrix B is defined as $C = \frac{1}{N-1}BB^T$. Let $C = (c_{ij})_{p \times p}$ and define a diagonal matrix D as follows:

$$D = \begin{bmatrix} c_{11} & 0 & \dots & 0 & 0 \\ 0 & c_{22} & \dots & 0 & 0 \\ \vdots & \dots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & c_{p-1p-1} & 0 \\ 0 & 0 & 0 & 0 & c_{pp} \end{bmatrix} \tag{11}$$

The diagonal distance between two feature vectors $\vec{\alpha}^i$ and $\vec{\alpha}^j$ is defined as:

$$d(\vec{\alpha}^i, \vec{\alpha}^j) = (\vec{\alpha}^i - \vec{\alpha}^j)^T D (\vec{\alpha}^i - \vec{\alpha}^j) \tag{12}$$

The impact of the AIPIR parameters, $MCRT$ and CR , on the performance of the algorithm is shown in Figs. 10 and 11. Fig. 10 shows how the value of the $MCRT$ impacts the pattern recognition success rate. When the value of the $MCRT$ is smaller than 0.85, the pattern recognition success rate is only about 70%. The pattern recognition success rate rises rapidly when the $MCRT$ value is greater than 0.85. The reason is that more candidate memory cells are injected into the memory cell set. Fig. 11 shows the impact of the CR and $MCRT$

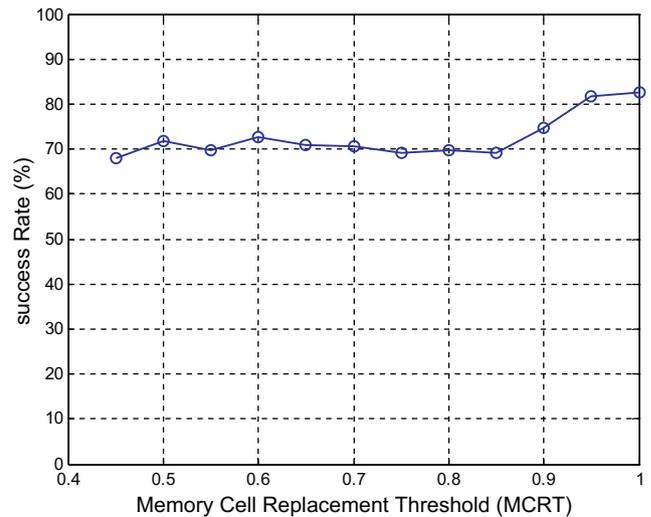


Fig. 10. Pattern recognition success rate vs. memory cell replacement threshold (MCRT).

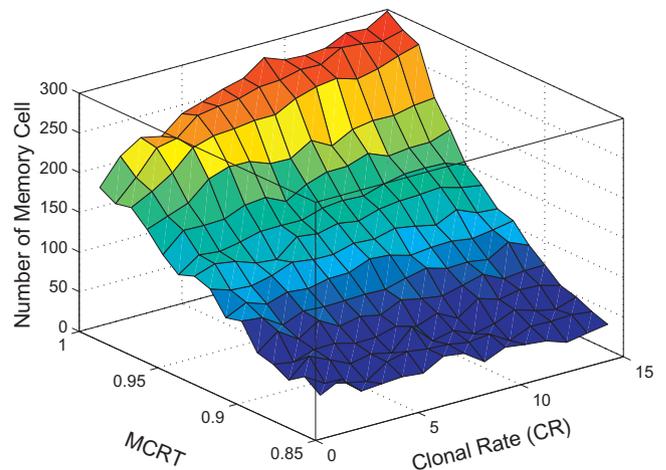


Fig. 11. Number of memory cells vs. CR and MCRT.

on the number of memory cells. The number of memory cells is one of major performance measurements for a pattern recognition algorithm. The number of memory cells is critical in the SHM sensor networks. Although a big memory cell set may raise the pattern recognition success rate, it will result in heavy computational load and slow system response. The value of the *MCRT* has a significant impact on the number of memory cells as shown in Fig. 11. When the value of the *MCRT* gets bigger, less matched memory cells are replaced, while more candidate memory cells are added into the memory cell set. The value of the *CR* also affects the number of memory cells. The appropriate values of the *CR* and *MCRT* should be chosen to limit the number of memory cells and achieve a reasonable pattern recognition success rate.

The impact of the number of training antigens on the pattern recognition success rate is also investigated. Fig. 12 shows the pattern recognition success rate with various number of training antigens. Given a number of training antigens, 10 cycles of training and pattern recognition are preformed. The mean value of 10 success rates is plotted in Fig. 12. Fig. 12 shows that the pattern recognition success rate fluctuates when the number of the training data is small. The increase of the number of training data stabilizes the pattern recognition success rate.

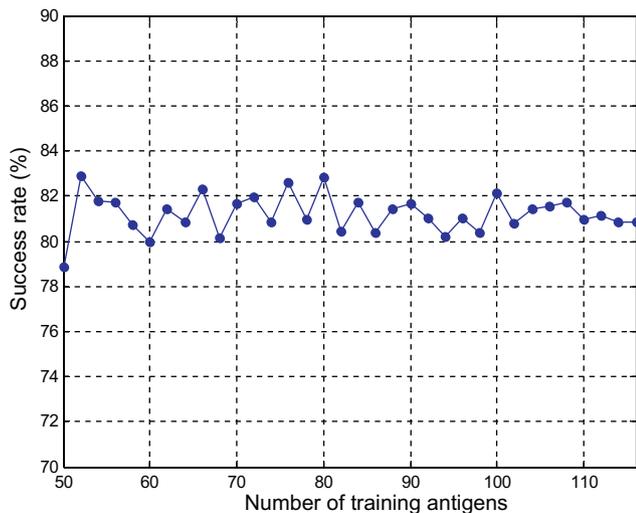


Fig. 12. Pattern recognition success rate vs. the number of training antigens.

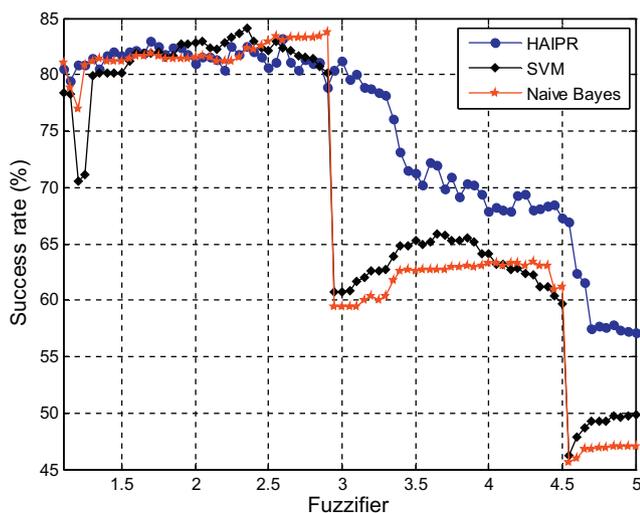


Fig. 13. Comparison of the pattern recognition success rate among the HAIPR, FC-SVM, and FC-Naive Bayes.

A comparison study of the pattern recognition success rate among the HAIPR, FC-SVM, and FC-Naive Bayes is shown in Fig. 13. The 580 feature vectors generated above are used in the comparison study. The system parameters selected for the HAIPR algorithm are $CR = 8$, $\sigma = 0.5$, $MCRT = 0.95$, $MCIT = 0.60$. When the value of the fuzzifier q varies from 1 to 5 with step 0.05, the pattern recognition success rates for different pattern recognition algorithms are calculated and the results are shown in Fig. 13. From Fig. 13, we can see that the HAIPR, FC-SVM, and FC-Naive Bayes methods have similar pattern recognition success rate if the value of the fuzzifier q is less than 2.8. When the q value is greater than 2.8, the HAIPR method outperforms significantly comparing to the FC-SVM and FC-Naive Bayes algorithms.

5. Conclusions

This paper presents an unsupervised pattern recognition algorithm based on the fuzzy clustering technique and the artificial immune pattern recognition. The fuzzy clustering method is used to generate initial memory cell for each damage pattern based on the structure's dynamic response data. The initial memory cells are evolved using immune learning mechanism to improve the representative quality of memory cells. The HAIPR method has been used for unsupervised structural damage pattern recognition with a benchmark structure proposed by the IASC-ASCE SHM Task Group. The performance analysis of the HAIPR-based unsupervised structural damage pattern recognition illustrates that some of the system's parameters, such as *fuzzifier*, distance types in fuzzy clustering algorithm, and the memory cell replacement threshold in the artificial immune pattern recognition algorithm, have a significant impact on the pattern recognition success rate and the number of memory cells. The comparison of the HAIPR, FC-SVM, and FC-Naive Bayes algorithms shows that the HAIPR method outperforms other two methods for the unsupervised damage pattern recognition using the IASC-ASCE benchmark structure.

References

- Carter, J. H. (2000). The immune system as a model for pattern recognition and classification. *Journal of the American Medical Informatics Association*, 7, 28–41.
- Castiglione, F., Motta, S., & Nicosia, G. (2001). Pattern recognition by primary and secondary response of an artificial immune system. *Theory in Biosciences*, 120, 93–106.
- Cesana, E., Beltrami, S., Laface, A. E., Urthaler, A., Folci, A., Clivio, A. (2006). Current paradigms in immunology. Neural nets. In *16th Italian workshop on neural nets, WIRN 2005 and international workshop on natural and artificial immune systems, NAIS 2005. Revised selected papers (lecture notes in computer science (Vol. 3931))* (pp. 244–260).
- Chen, B. (2009). A biologically inspired sensor network framework for autonomous structural health monitoring. In *Proc of SPIE: Sensors and smart structures technologies for civil, mechanical, and aerospace systems, San Diego, California*.
- Chen, B., Zang, C. (2009). Artificial immune pattern recognition for damage detection in structural health monitoring sensor networks. In *Proc of SPIE: Smart sensor phenomena, technology, networks, and systems, San Diego, California*.
- Dasgupta, D. (2006). Advances in artificial immune systems. *IEEE Computational Intelligence Magazine*, 1, 40–49.
- Dasgupta, D., Krishnakumar, K., Wong, D., Berry, M. (2004). Negative selection algorithm for aircraft fault detection. Artificial immune systems. In *Third international conference, ICARIS 2004. Proceedings (lecture notes in Comput. Sci. (Vol. 3239))* (pp. 1–13).
- De Castro, L. N. (2006). *Fundamentals of natural computing: basic concepts, algorithms, and applications*. Chapman & Hall/CRC.
- De Castro, L. N., & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. Springer.
- De Castro, L. N., & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6, 239–251.
- Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A – Mathematical Physical and Engineering Sciences*, 365, 303–315.
- Freitas, A. A., & Timmis, J. (2007). Revisiting the foundations of artificial immune systems for data mining. *IEEE Transactions on Evolutionary Computation*, 11, 521–540.

- Hart, E., & Timmis, J. (2008). Application areas of AIS: the past, the present and the future. *Applied Soft Computing Journal*, 191–201.
- Kolakowski, P. (2007). Structural health monitoring – A review with the emphasis on low-frequency methods. *Engineering Transactions, IPPT*, 55, 239–275.
- Negoita, M. (2005). Artificial immune systems – An emergent technology for autonomous intelligent systems and data mining. Autonomous intelligent systems: Agents and data mining. In *International workshop, AIS-ADM 2005. Proceedings (lecture notes in artificial intelligence (Vol. 3505))* (pp. 19–36).
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 559–572.
- Polat, K., Gunes, S., & Tosun, S. (2006). Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing. *Pattern Recognition*, 39, 2186–2193.
- Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18, 401–409.
- Sazonov, E., Janoyan, K., Jha, R. (2004). Wireless intelligent sensor network for autonomous structural health monitoring. In *Proceedings of the SPIE – The international society for optical engineering* (Vol. 5384, pp. 305–314).
- Structural Health Monitoring Benchmark Problem. <<http://mase.wustl.edu/wusceel/asce.shm/benchmarks.htm>>.
- Taylor, D. W., Corne, D. W. (2003). An investigation of the negative selection algorithm for fault detection in refrigeration systems. In *Proceeding of second international conference on artificial immune systems (ICARIS)* (pp. 34–45).
- Theodoridis, S., & Koutroumbas, K. (2008). *Pattern recognition*. Academic Press.
- Watkins, A., Timmis, J., & Boggess, L. (2004). Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5, 291–317.
- Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R., et al. (2004). A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on embedded networked sensor systems, Baltimore, MD, USA, 2004* (pp. 13–24).
- Zhong, Y. F., Zhang, L. P., Gong, J. Y., & Li, P. X. (2007). A supervised artificial immune classifier for remote-sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45, 3957–3966.