

Artificial immune pattern recognition for structure damage classification

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, United States

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

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

ARTICLE INFO

Article history:

Received 30 December 2008

Accepted 14 August 2009

Available online 10 September 2009

Keywords:

Structural health monitoring

Artificial immune pattern recognition

Structure damage classification

ABSTRACT

Damage detection in structures is one of the research topics that have received growing interest in research communities. While a number of damage detection and localization methods have been proposed, very few attempts have been made to explore the structure damage classification problem. This paper presents an Artificial Immune Pattern Recognition (AIPR) approach for the damage classification in structures. An AIPR-based structure damage classifier has been developed, which incorporates several novel characteristics of the natural immune system. The structure damage pattern recognition is achieved through mimicking immune recognition mechanisms that possess features such as adaptation, evolution, and immune learning. The damage patterns are represented by feature vectors that are extracted from the structure's dynamic response measurements. The training process is designed based on the clonal selection principle in the immune system. The selective and adaptive features of the clonal selection algorithm allow the classifier to evolve its pattern recognition antibodies towards the goal of matching the training data. In addition, the immune learning algorithm can learn and remember different data patterns by generating a set of memory cells that contains representative feature vectors for each class (pattern). The performance of the presented structure damage classifier has been validated using a benchmark structure proposed by the IASC–ASCE (International Association for Structural Control–American Society of Civil Engineers) Structural Health Monitoring (SHM) Task Group and a three-story frame provided by Los Alamos National Laboratory. The validation results show that the AIPR-based pattern recognition is suitable for structure damage classification. The presented research establishes a fundamental basis for the application of the AIPR concepts in the structure damage classification.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The civil structures, such as bridges and buildings, play an important role in people's daily life. Maintaining safe and reliable civil structures is important to the well being of all of us [1]. The sudden failure and collapse of the I-35W Interstate system bridge in Minneapolis has raised policy concerns in US Congress regarding the condition of the nation's transportation infrastructure [2]. Based on the CRS (Congressional Research Service) Report for Congress [2], in 2006, about 26% of bridges were classified as either structurally deficient, functionally obsolete, or both. About 12% of bridges in that year, approximately 74,000, were classified as structurally deficient. To ensure civil structures meeting life-safety standards over their operational lives, early identification and assessment of structural damage are necessary [3].

* Corresponding author. Address: Department of Mechanical Engineering – Engineering Mechanics, Michigan Technological University, 815 R.L. Smith Building, 1400 Townsend Drive, Houghton, MI 49931, United States. Tel.: +1 906 487 3537.
E-mail address: bochen@mtu.edu (B. Chen).

Structural Health Monitoring (SHM) holds promise for monitoring structure performance with an excellent cost/benefit ratio. The SHM process involves the observation of a structure's dynamic response measurements from a group of sensors, the extraction of damage-sensitive features from these measurements, and analysis of these features to determine the current state of the structure [4]. Damage identification is one of the research topics that have been extensively investigated. The vibration-based damage assessment of the bridge structures and buildings has been studied since the early 1980s. Doebling et al. [5] summarized the researches on the vibration-based damage identification and health monitoring. Sohn et al. [6] reviewed the technical papers in structural health monitoring, published between 1996 and 2001. Most conventional structural health monitoring methods are modal analysis based. Modal parameters, such as natural frequencies, damping ratios, and mode shape curvature, have been the primary features used to identify damage in structures. Recently, a number of new approaches, such as statistical pattern recognition [7,8] and neural network [9–11], have been proposed for the damage diagnosis. For example, Sohn and Farrar [7] proposed a statistical pattern

recognition method for the damage diagnosis using time series analysis of vibration signals. The residual error ratio of ARX models for test signal and the reference signal is defined as the damage-sensitive feature. Lee et al. [9] presented a method for damage detection in a plate structure and damage localization using neural network technique.

While a lot of efforts have been made in detecting damages in structures, few researches have been conducted for the structure damage classification. The damage classification is not only in detecting damage but also to categorize detected damage pattern to one of a number of possible damage categories. Note that the term “pattern” in the paper has the same meaning as “class” or “category.” This paper presents an Artificial Immune Pattern Recognition (AIPR) method for structure damage detection and classification. An AIPR-based Structure Damage Classifier (AIPR-SDC) has been developed for the supervised structure damage classification. This classification method can be applied to any problem where time series are involved. The AIPR-SDC presents a number of advantages. First, it is adaptive. The type of antibodies and memory cells can adapt to the antigenic stimulation through clonal selection algorithm. Second, it has learning capability. Different patterns are recorded in the memory via memory cells. Third, the advanced selection mechanism allows the classifier to keep best memory cells in the memory cell set. The AIPR-SDC algorithm is based on the CLONALG algorithm in [12] and the AIRS in [13]. The contributions of the paper include (1) design memory cell update strategies to obtain good representatives for each damage pattern, (2) the modification of general AIPR algorithm in order to be able to cope with time series data, (3) providing an immune inspired solution to the structure damage classification problem. The AIPR-based structure damage classifier has been used to classify structure damage patterns using a benchmark structure proposed by the IASC-ASCE (International Association for Structural Control–American Society of Civil Engineers) SHM Task Group [14] and a three-story frame provided by Los Alamos National Laboratory (LANL) [15].

The rest of the paper is organized as follows: Section 2 introduces basic concepts of natural and artificial immune systems. Section 3 presents the algorithm design of the AIPR-based structure damage classifier. Section 4 shows the classifier validation results and discusses the impact of system parameters on the performance of the AIPR-SDC. Section 5 concludes the presented work.

2. Natural and artificial immune systems

The natural immune system is a rapid and effective defense mechanism for a given host against infections [16]. 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. When a B-cell encounters a nonself antigen that has sufficient affinity with its receptor (antibody), the B-cell is activated. It, therefore, undergoes a clonal selection process that generates B-cells with similar receptors as the activated B-cell. The 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 [17]. The *secondary response* occurs when the same antigen is encountered again. 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 compare to a newly activated B-cell [18].

The Artificial Immune Systems (AIS) can be defined as the abstract or metaphorical computational systems developed using ideas, theories, and components, extracted from the immune

system [19]. The AIS seems best suited to handle the great complexity of the reality [17]. 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 (e.g., immune network of B-cells), adaptive learning through memory (e.g., memory B-cells), and an advanced selection mechanism of the best B-cells [20]. The AIS has found various applications in the fields of pattern recognition, fault diagnosis, and intrusion detection. In the pattern recognition area, a number of researchers have exploited immune concepts for supervised and unsupervised classification [12,13], remote sensing image classification [21,22], and medical classification problems [23].

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes [24]. The pattern recognition method classifies data (patterns) based on either a priori knowledge or on the statistical information extracted from the patterns. The patterns to be classified are usually the groups of measurements, defining points in an appropriate multi-dimensional space. The measurements used for the classification are known as features. If p features are used f_i , $i = 1, 2, \dots, p$, these p features can form a feature vector $F = (f_1, f_2, \dots, f_p)^T$, where T denotes transposition. The generation of the feature vector is problem dependant, and the feature selection is critical to the success of the design of a classification system. The pattern recognition has two types: supervised and unsupervised. For supervised pattern recognition, training data for each class are available for the design of the classifier, while in unsupervised pattern recognition, the class label information of training data are not available. For the unsupervised pattern recognition problem, the goal is to cluster “similar” feature vectors by unraveling their underlying similarities [24].

3. AIPR-based structure damage detection and classification

3.1. An AIPR-based structure damage classifier

This section introduces structure damage classification based on time series data from sensor nodes. The classification system is designed using concepts derived from the natural immune system. The component correspondence between the natural immune system and the AIPR-based structure damage classifier, AIPR-SDC, is shown in Table 1. The AIPR-SDC algorithm consists of two major stages as shown in Fig. 1. The first stage is the data pre-processing and feature extraction. In this stage, all the training data (sensor data) need to be standardized and the feature vectors need to be generated. In addition, memory cell set and antibody set for all the classes are initialized. In the second stage, the training antigen stimulates the antibody set and thus causes some of antibodies to produce clones. The cloned antibodies are mutated to increase the affinity level of the antibody set with the invading antigen. The antibody having highest affinity with the stimulating antigen is chosen as a candidate memory cell for updating memory cell set.

3.1.1. Major components and parameters of the AIPR-SDC

This section defines the major components and parameters used in the AIPR-SDC algorithm.

Table 1
Mapping between the natural immune system and the AIPR-SDC.

Natural immune system	AIPR-SDC
Antibody	Feature vector with class information
Antigens	Training and classification data
B-cells	Artificial B-cells
Immune memory	Memory cells
Primary response	Antigenic stimulation to the antibody set

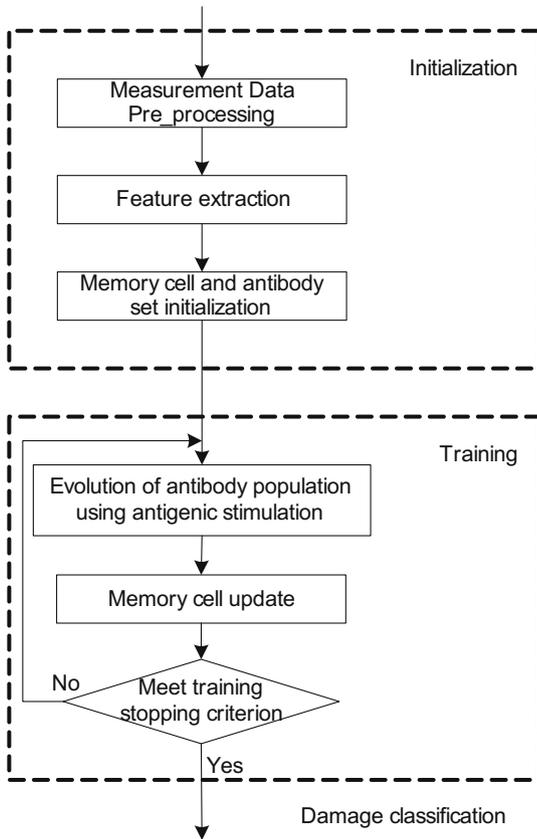


Fig. 1. The major stages of the AIPR-SDC.

1. Feature vector and feature space: a feature vector is a p -dimensional vector consisting of p numerical features to represent an object. A feature space is an abstract space where the data of each pattern is represented as a point in a p -dimensional real vector space R^p . The dimension of a feature space is determined by the number of features in the feature vector that describes the patterns. For example, when a time series is represented by an auto regressive (AR) model of order p , the feature vector of the time series could be a p -dimensional real vector consisting of p coefficients of the AR model.
2. Antigen: an antigen is a substance that stimulates the generation of antibodies and updates memory cells in the same class. In artificial immune pattern recognition systems, an antigen could be a training or classification data that has two attributes, a feature vector and the class to which the antigen belongs.
3. Antibody and antibody set: an antibody has the same data structure (representation) as an antigen. The antibody set contains nc number of subsets. The number of nc is the number of patterns to be classified. The antibodies in i th subset are able to recognize i th pattern.
4. Artificial B-cell: an artificial B-cell is analogous to a B-cell in the natural immune system, which contains antibodies for recognizing antigens.
5. Affinity: the degree of match between an antigen and an antibody of a B-cell or a memory cell. The affinity value depends on the distance between two feature vectors of two objects (defined in Eq. (12)). Longer distance means lower affinity, and shorter distance means higher affinity. In our implementation, the value of the affinity is limited between 0 and 1.

6. Memory cell and memory cell set: a memory cell is an antibody that has highest affinity with a previously invading antigen. The memory cell set consists of several subsets. The number of subsets is equal to the number of patterns. The memory cells for one class are representatives of this class and contained in one subset.
7. Matched memory cell ($MC_{matched}$): the memory cell that has the highest affinity with the training antigen in the same class.
8. Candidate memory cell ($MC_{candidate}$): the antibody that has highest affinity with the antigen after the antibody set is stimulated by the antigen.
9. Clonal value (CV): a value that measures the response of an artificial B-cell to an antigen. The clonal value combined with the clonal rate defined below determines how many clones are generated for the activated B-cell. This value is equal to the affinity between an antibody and the stimulating antigen.
10. Mutation value (MV): a value used to indicate the mutation degree of the feature vector of an antibody.
11. Clonal rate (CR): an integer value used to control the number of antibody clones allowed for the activated B-cell. The number of clones is $CR * CV$.
12. Hyper-clonal rate (HCR): an integer value to control the number of antibodies cloned from the matched memory cell.
13. Memory cell replacement threshold (MCRT): a threshold value to determine the replacement of an existing memory cell with the candidate memory cell.
14. Memory cell injection threshold (MCIT): a threshold value to determine if the candidate memory cell is added into the memory cell set.
15. $MaxABN$: the maximum number of the antibodies for each class in the antibody set.
16. σ : the standard deviation of a normal random variable $N(0, \sigma)$.

3.1.2. Notational convention

Before introducing the AIPR-based damage classifier in detail, following notational convention is giving for ease of understanding the algorithm.

- Let ab denote the single antibody. Let $ab.f$ and $ab.c$ denote the feature vector and the class (pattern) of the antibody ab , respectively, where $ab.f \in R^p$, $ab.c \in C = \{1, 2, \dots, nc\}$, R^p is a p -dimensional real value space and nc is the number of classes.
- Let ABS denote the Antibody Set that contains antibodies for all the classes. Let ABS_i denote the antibody subset of the i th class such that $ABS_i = \{ab|ab.c = i\}$, $1 \leq i \leq nc$ and $\bigcup_{i=1}^{nc} ABS_i = ABS$.
- Let mc denote the single memory cell. Let $mc.f$ and $mc.c$ denote the feature vector and the class information of the memory cell, respectively, where $mc.f \in R^p$, $mc.c \in C = \{1, 2, \dots, nc\}$, R^p is a p -dimensional real value space and nc is the number of classes.
- Let MCS denote the Memory Cell Set, consisting of memory cells for all the classes. Let MCS_i denote the memory cell subset of the i th class such that $MCS_i = \{mc|mc.c = i\}$, $1 \leq i \leq nc$ and $\bigcup_{i=1}^{nc} MCS_i = MCS$.
- Let ag denote an antigen. Let $ag.f$ and $ag.c$ denote the feature vector and the class of the antigen ag respectively, where $ag.f \in R^p$, $ag.c \in C = \{1, 2, \dots, nc\}$, R^p is a p -dimensional real value space and nc is the number of classes.

3.2. AIPR-based structure damage classifier algorithm

3.2.1. Data pre-processing, feature extraction, and initialization

This subsection introduces how to process measurement data, calculate feature vectors, and initialize the memory cell set and

the antibody set for the classifier training. The steps of this process are shown in Fig. 2. The algorithm for each step is described in the following subsections.

3.2.1.1. Data standardization. The measurement data are standardized to reduce the environmental effects. Let matrix $Z = (z_{ij})_{m \times n}$ denote the time series of measurement data, where each row is corresponding to the n number of data generated by one sensor and each column is the measurement 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 measurement data of i th sensor. The standardized measurement data $Y = (y_{ij})_{m \times n}$ can be calculated by Eq. (1):

$$y_{ij} = \frac{z_{ij} - \mu_i}{\sigma_i}, \quad j = 1, 2, \dots, n, \quad (1)$$

where y_{ij} is the standardized value of the corresponding z_{ij} , μ_i and σ_i are the mean and standard deviation of the time series z_i .

3.2.1.2. Dimensionality reduction using principle component analysis (PCA) method. For monitoring a structure, multiple sensors are usually used to collect data from different locations. To extract feature vectors for a local area, time series measurement data sets from multiple sensors are reduced to lower dimensions by the Principal Component Analysis (PCA) method. The PCA is a statistical technique that uses a substantially smaller set of uncorrelated variables to represent the maximum amount of information from the original set of variables [8]. The PCA method involves the calculation of the eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centering the data for each attribute.

Let Ψ denote the $m \times m$ covariance matrix of the standardized time signals Y . The matrix Ψ can be obtained by

$$\Psi = \frac{1}{n-1} YY^T. \quad (2)$$

Let λ_i and \bar{v}_i denote the i th eigenvalue and eigenvector of matrix Ψ , respectively and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$. Then Ψ, λ_i and v_i satisfy:

$$\Psi \bar{v}_i = \lambda_i \bar{v}_i \quad i = 1, 2, \dots, m, \quad (3)$$

where eigenvector \bar{v}_i is called the i th principal component. In order to reduce the m -dimensional measurement data set into a d -dimensional data set, Y should be projected onto the eigenvectors corresponding to the first d largest eigenvalues:

$$X = (\bar{v}_1, \bar{v}_2, \dots, \bar{v}_d)^T Y, \quad (4)$$

where X is the compressed time signals.

In our study, the time series from m number of sensors are compressed into a single time series. It means that all the sensor measurement data are projected onto the principal component that has the biggest eigenvalue. Let \bar{v}_1 denote the vector that is corresponding to the biggest eigenvalue. So, the relationship between the compressed data $\bar{x} = (x_1, x_2, \dots, x_n)$, \bar{v}_1 and Y is shown in Eq. (5)

$$\bar{x} = \bar{v}_1^T Y. \quad (5)$$

3.2.1.3. Feature extraction using multiple regression analysis. Once the measurement data Z are compressed into a one dimensional data set, the next step is to extract the feature vector from the compressed data for the classification. Since the compressed data \bar{x} consists of a large number of data points, it is not suitable to be used as a feature vector directly. The auto regressive algorithm is chosen to model the compressed time series data. Each compressed time series x is fitted to an AR model of order p as shown in Eq. (6)

$$x_k = \sum_{i=1}^p \alpha_i x_{k-i} + r_k \quad k = p+1, \dots, n, \quad (6)$$

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

There are several ways to calculate the feature vector α , such as Least Square (LS) and Yule Walker (YW). The LS method is used in our implementation. Eq. (6) is rewritten as follows:

$$A\alpha = b, \quad (7)$$

where

$$A = \begin{bmatrix} x_p & x_{p-1} & \dots & x_1 \\ x_{p+1} & x_p & \dots & x_2 \\ \dots & \dots & \ddots & \dots \\ x_{n-1} & x_{n-2} & \dots & x_{n-p-1} \end{bmatrix}, \quad b = \begin{bmatrix} x_{p+1} \\ x_{p+2} \\ \vdots \\ x_n \end{bmatrix} \quad (8)$$

The feature vector α can be calculated as follows:

$$\alpha = (A^T A)^{-1} A^T b. \quad (9)$$

To keep the affinity values within the range of (0, 1), the norm (length) of the feature vectors of all training and classification data are normalized to the unit hyper-sphere. The normalization process uses the maximum norm of the feature vectors. Let f denote the feature vector of one measurement data and $MaxNorm$ denote the maximum norm of all the measurement data, the normalized feature vector $f_{normalized}$ of $f = (f_1, f_2, \dots, f_p)^T$ is as follows

$$f_{normalized} = \frac{1}{MaxNorm} f, \quad (10)$$

where $MaxNorm = \max_{f \in \text{all measurement data}} (norm(f))$ and $norm(f) = \sqrt{(\sum_{i=1}^p f_i^2)}$ is the norm of the feature vector. Since all the feature vectors are located within a unit hyper-sphere, the distance between any feature vectors is less than 2, and their affinity is within the range of (0, 1) based on the affinity definition given in the Section 3.2.2.1. In the remaining part of the paper, all feature vectors are referred to the normalized feature vectors.

3.2.1.4. Initialization of antibody set and memory cell set. The initial antibodies for each class (pattern) are randomly selected from the feature vectors of the training data in each class since the bad antibodies will be easily replaced by cloned antibodies that

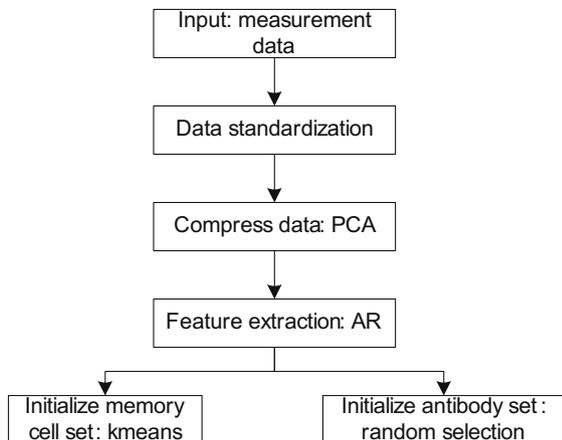


Fig. 2. Data pre-processing, feature extraction, and initialization.

have higher affinity with the antigen. For memory cells, however, the replacement happens only to the candidate memory cells that meet certain criteria as stated in the later Section 3.2.2.2. To obtain good representatives for each class when the memory cells are initialized, the k -means algorithm is applied to the corresponding training data in each class to generate initial memory cells. The k -means algorithm is used one time for each class. The k -means algorithm [24] clusters n number of multi-dimensional points into k partitions, where $k < n$. The resulting k vectors c^i , $i = 1, 2, \dots, k$ are the centroids of k clusters with minimum intra-cluster variance:

$$E = \sum_{i=1}^k \sum_{x^j \in S_i} (x^j - c^i)^2, \tag{11}$$

where S_i is the i th cluster. These k vectors c^i are chosen as the initial memory cells for each class. Since the value k is selected as $k = 4$, the initial number of memory cells for each class is 4. The initial number of antibodies for each class is chosen 10.

3.2.2. Classifier training process

The goal of the training process is to develop memory cells that are good feature representations for each pattern. The classifier training process consists of the antibody set refinement process (evolution of antibody set) and the memory set refinement process (update of memory cell set). The flow chart of the training process is shown in Fig. 3.

3.2.2.1. Evolution of antibody population using antigenic stimulation. The stimulation of antibody set by an invading antigen will cause the evolution of the antibody set. The algorithm to evolve the antibody set by an antigenic stimulation is shown in Fig. 5. Given a training antigen ag , for each antibody ab that is in the same class as the antigen ag , the affinity between an antibody and the antigen is calculated. Let $ab.f = \beta = (\beta_1, \beta_2, \dots, \beta_p)^T$ and $ag.f = \gamma = (\gamma_1, \gamma_2, \dots, \gamma_p)^T$ denote the feature vectors of an antibody ab and an antigen ag , respectively. The affinity between an antibody and the antigen is defined as

$$aff(ab, ag) = 1 - \frac{1}{2} dist(\beta, \gamma), \tag{12}$$

$$dis(\beta, \gamma) = \left(\sum_{i=1}^p (\beta_i - \gamma_i)^2 \right)^{\frac{1}{2}}, \tag{13}$$

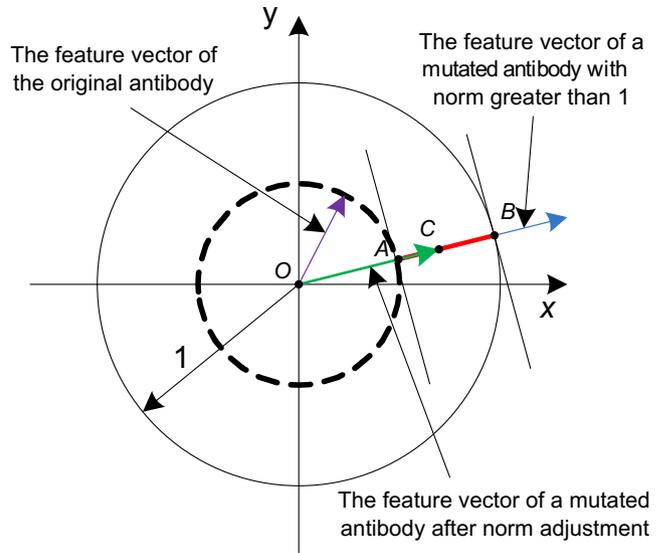


Fig. 4. Norm check for the mutated antibodies.

where $dist(\beta, \gamma)$ is the distance between the feature vectors of β and γ . The probability that an antibody ab is cloned depends on its affinity with the antigen. It means that an antibody with higher affinity has higher probability to be cloned. The number of the cloned antibodies, $CloneNumber$, depends on the clonal rate CR and the clonal value CV . According to the natural immune system, the higher the affinity, the larger the number of antibodies is cloned. The clonal value CV is the reflection of this affinity. We choose the clonal value being equal to the affinity value. Let $CloneNumber$ denote the number of the cloned antibodies, the value of $CloneNumber$ can be calculated by Eq. (14)

$$CloneNumber = round(CR * CV) = round(CR * aff(ab, ag)), \tag{14}$$

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

The cloned antibodies undergo an affinity maturation process that increases the diversity of the antibody set. Let $ab_{mutated}$ denote the mutated antibody, the mutation is performed by mutating the feature vectors of the cloned antibodies as shown in Eq. (15)

$$ab_{mutated}.f = ab.f + MV \times \phi, \tag{15}$$

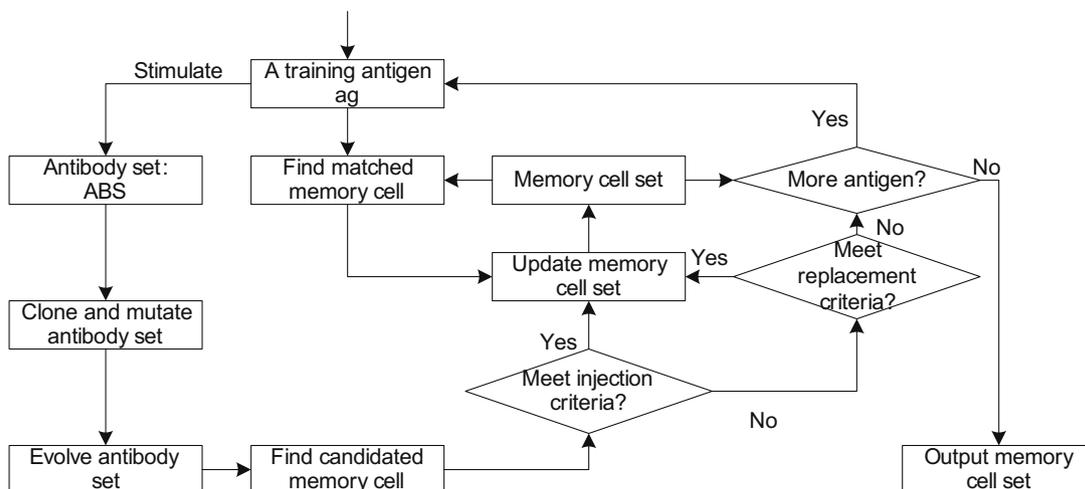


Fig. 3. The training process of the AIPR-SDC.

```

Given an antigen  $ag$ , do{
  for each antibody  $ab$  in the class  $ag.c$ , do{
     $class = ag.c$ ;
     $CV = aff(ab, ag)$ ;
     $MV = 1 - aff(ab, ag)$ ;
     $CloneNumber = CR * CV$ ;
     $CloneNumber = round(CloneNumber)$ ;
    If ( $rand() < affinity$ ) {
      for ( $i = 1, i++, i < CloneNumber$ ) {
         $ab_{clone} = ab$ ;
         $ab_{clone}.c = class$ ;
         $ab_{mutated}.f = ab.f + MV * \bar{N}(0, \sigma^2)$ 
        if ( $norm(ab_{mutated}.f) > 1$ )
           $ab_{mutated}.f = (norm(ab.f) + rand() * (1 - norm(ab.f))) * \left( \frac{ab_{mutated}.f}{norm(ab_{mutated}.f)} \right)$ ;
         $ABS_{class} = ABS_{class} \cup ab_{mutated}$ 
      }
    }
  }
  If  $|ABS_{class}| > MaxABN$  {
     $ABS_{class} = sort(ABS_{class})$ ; /*sort antibodies in a descending order with the affinity values.*/
     $ABS_{class} = \text{the top } MaxABN \text{ number of antibodies in } ABS_{class}$ ;
  }
   $MC_{candidated} = \arg \max_{ab \in ABS_{class}} aff(ab, ag)$ ;
}

```

Fig. 5. Antibody set clone and mutation process.

where $ab_{mutated}$ is the mutated antibody and MV is the mutation value. The mutation value MV is problem dependent. Typically, the higher the affinity is, the smaller the mutation value. In our implementation, the mutation value MV is defined in Eq. (16)

$$MV = 1 - CV. \quad (16)$$

In Eq. (15), 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 of the random vector 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 .

To make sure that the mutated antibody feature vectors stay within the unit hyper-sphere, the norm of the feature vector for each mutated antibody is checked after the mutation. This check process is necessary because the mutated antibody feature vectors

may extend to the outside of the unit hyper-sphere, which would cause negative affinity values. If the norm of the mutated feature vector is greater than 1, Eq. (17) is applied to shrink the feature vector of the mutated antibody back to the unit hyper-sphere without changing the direction of the mutated antibody feature vector. In Eq. (17), $norm(ab.f)$ is the norm of the original antibody feature vector. The $rand()$ is a uniform random function with a value within the range of $[0, 1]$. The term $\frac{ab_{mutated}.f}{norm(ab_{mutated}.f)}$ is a unit vector at the direction of the mutated antibody feature vector $ab_{mutated}.f$. As shown in Eq. (18), the norm of the mutated antibody feature vector after applying Eq. (17), $norm(ab.f) + rand() * (1 - norm(ab.f))$, is greater than original antibody feature vector $norm(ab.f)$ and less than 1. The direction of the adjusted antibody feature vector is determined by the unit vector $\frac{ab_{mutated}.f}{norm(ab_{mutated}.f)}$, which is the same as the mutated antibody feature vector. The segment

AB in Fig. 4 shows the possible range where the resulting feature vector of the Eq. (17) could be located, and the vector \vec{OC} is the adjusted antibody feature vector

$$ab_{mutated}.f = (norm(ab.f) + rand() * (1 - norm(ab.f))) * \left(\frac{ab_{mutated}.f}{norm(ab_{mutated}.f)} \right), \quad (17)$$

$$0 \leq norm(ab.f) \leq norm(ab.f) + rand() * (1 - norm(ab.f)) \leq 1. \quad (18)$$

The mutated antibodies are added into the antibody subset that corresponds to the class to which the ag belongs. Since the maximum number of each antibody subset is limited to $MaxABN$, 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 highest affinity antibody is chosen as the candidate memory cell $MC_{candidate}$ for possible updating of memory cell set, which will be discussed in the next section.

3.2.2.2. Update memory cells. The candidate memory cell generated in the antibody evolution process is used to update the memory cell set to enhance the representation quality of memory cells for each pattern. The pseudo-code of the memory cell update process is shown in Fig. 6. 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 class is greater than a specified threshold value $MCIT$, the candidate memory cell is injected into this class of memory cells. Let $|MCS_{ag.c}|$ denote the total number of the memory cells in the subset $MCS_{ag.c}$. The rms is calculated by Eq. (19)

$$rms = RMS(dist_1, dist_2, \dots, dist_{|MCS_{ag.c}|}) = \frac{1}{\sqrt{|MCS_{ag.c}|}} \sqrt{\sum_{i=1}^{|MCS_{ag.c}|} dist_i^2}, \quad (19)$$

where $dist_i = dist(mc_i, MC_{candidate})$, $mc_i \in MCS_{ag.c}$, $i = 1, 2, \dots, |MCS_{ag.c}|$ and $|MCS_{ag.c}|$ is the number of the memory cells in the subset $MCS_{ag.c}$. If $rms > MCIT$, the candidate memory cell is added into the memory cell subset $MCS_{ag.c}$. According to the second scenario, the candidate memory cell compares with the matched memory cell. The matched memory cell ($MC_{matched}$) is the memory cell that has the highest affinity with the given antigen in the same class. To find the matched memory cell, the affinity values of the training antigen with the memory cells in the same class are calculated. The memory cell that has the highest affinity with the given antigen ag is chosen as the matched memory cell. Let $MC_{matched}$ denote the matched memory cell, it can be found by Eq. (20)

$$MC_{matched} = \arg \max_{mc \in MCS_{ag.c}} aff(ag, mc). \quad (20)$$

If $rms \leq MCIT$, $aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)$ & $aff(MC_{candidate}, MC_{matched}) > MCRT$ (a predefined threshold), the candidate memory cell replaces the matched memory cell. In the third case, if $rms \leq MCIT$, $aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)$ and $aff(MC_{candidate}, MC_{matched}) \leq MCRT$, the candidate memory cell is also injected into the memory cell subset $MCS_{ag.c}$.

3.2.3. Damage classification process

The memory cells generated in the training process are used to classify test data during the classification process as shown in Fig. 7. For a damage-pattern-unknown time series data, the affinities between the feature vectors of the measurement data with

```

Given an antigen  $ag$ , do{
     $class = MC_{candidate}.c$ ;

     $MC_{matched} = \arg \max_{mc \in MCS_{class}} aff(ag, mc)$ ;

    for each memory cell  $mc_i$  in  $MCS_{class}$ , do{

         $dist_i = dist(MC_{candidate}, mc_i)$ 

    }

     $rms = RMS(dist_1, dist_2, \dots, dist_{|MCS_{class}|})$ ;

    If ( $rms > MCIT$ )

         $MCS_{class} = MCS_{class} \cup MC_{candidate}$ 

    else if ( $(aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)) \& (aff(MC_{candidate}, MC_{matched}) > MCRT)$ ){

         $MCS_{class} = MCS_{class} - MC_{matched}$ ;

         $MCS_{class} = MCS_{class} \cup MC_{candidate}$ 

    }

    else if ( $aff(MC_{candidate}, ag) > aff(MC_{matched}, ag)$ )

         $MCS_{class} = MCS_{class} \cup MC_{candidate}$ 

}

```

Fig. 6. Memory cell set update.

memory cells in the memory cell set are calculated. The pattern of the test data is classified to the same class as the memory cell with whom the test data has highest affinity.

4. Structure damage classification

Two civil structures are used to validate the AIPR-SDC algorithm. One is the benchmark structure proposed by the IASC-ASCE SHM Task Group, and the other one is a three-story frame provided by Los Alamos National Laboratory.

4.1. Damage classification for benchmark structure proposed by IASC-ASCE

4.1.1. Benchmark structure

The developed AIPR-based structure damage classifier has been used to classify structure damage patterns using a benchmark structure [14] proposed by the IASC-ASCE SHM Task Group as shown in Fig. 8. The frame is a 4-story, 2 bay by 2 bay steel-frame scale model structure in the Earthquake Engineering Research Laboratory at the University of British Columbia (UBC) [25]. The structure has 16 accelerometers, 2 x-direction and 2 y-direction per floor as shown in Fig. 9. Finite element models based on this structure were developed to generate the simulated data. Five damage patterns are defined by the ASCE SHM Task Group and four of them were used in our study. These damage patterns are (1) all braces of the first floor removed; (2) all braces of the first and the third floor removed; (3) one brace removed from the first floor; (4) one brace removed from each of the first and the third floors. The data generation program is available on the web at <http://mase.wustl.edu/wusceel/asce.shm/>.

4.1.2. Data generation and AR order selection

The training and classification data are generated for normal and four damage patterns under various operational conditions. The excitation force modeled as filtered Gaussian white noise is applied to each floor as shown in Fig. 9. The ranges of each parameter and their steps for generating training data are listed in Table 2. The parameters include the damping, noise level, force intensity, time step (sampling interval), and the time duration (sampling time). Total of 1750 scenarios, 350 for each pattern is used to train

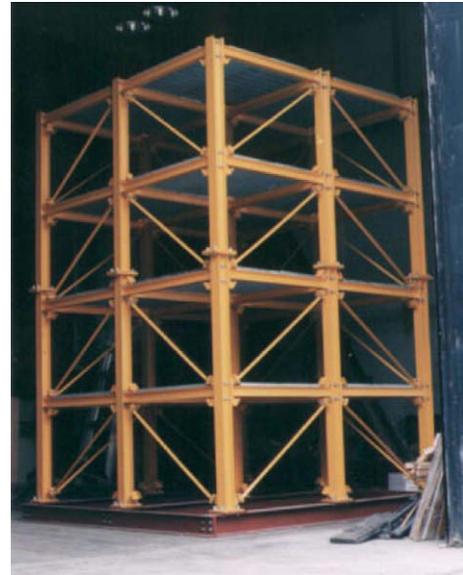


Fig. 8. Benchmark testing structure [14] (Photo courtesy Prof. Carlos Ventura, UBC).

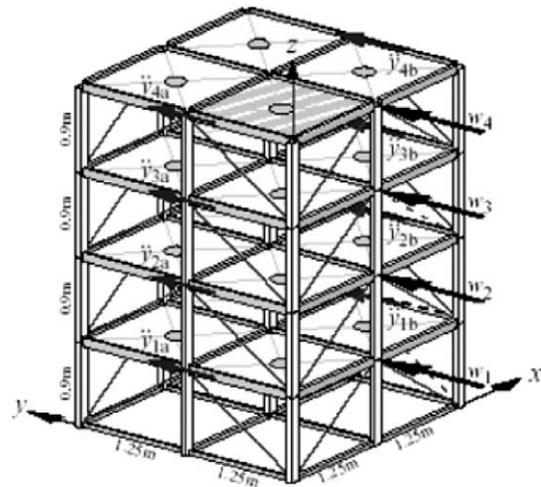


Fig. 9. Analytical model of the left side steel-frame structure [25].

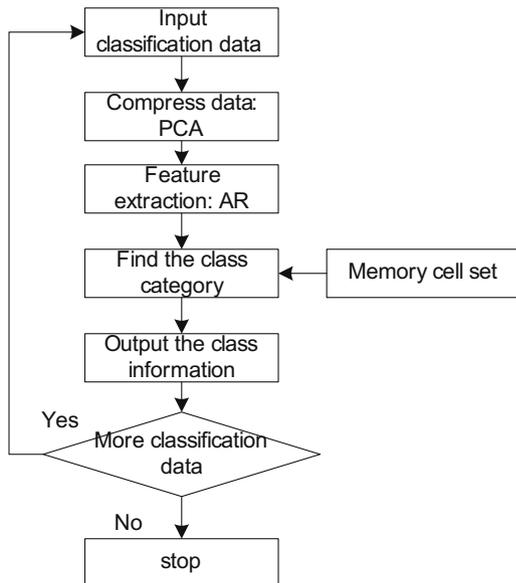


Fig. 7. Classification process of the AIPR-SDC.

the designed classifier. In addition, the classification data under 300 simulation cases are generated for validating the classifier. The parameter ranges and steps for the generation of classification data are shown in Table 3. The acceleration data from 16 accelerometers are reduced to a single time series using PCA method. The compressed acceleration data for the normal pattern and the damage pattern 2 are shown in Figs. 10 and 11 when the damping, noise level, and force intensity are 0.02, 20, and 100, respectively.

The order of AR models is selected 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. For five data sets used in the damage classification, the value of AIC is calculated for different AR

Table 2
The parameters for generating training data.

Parameters	Damping	Noise level	Force intensity	Time (s)
Range	0.01–0.1	10–30	100–250	0–2
Step	0.01	5	25	1/512

Table 3
The parameters for generating classification data.

Parameters	Damping	Noise level	Force intensity	Time (s)
Range	0.015–0.1	12.5–30	112.5–250	0–2
Step	0.02	5	50	1/512

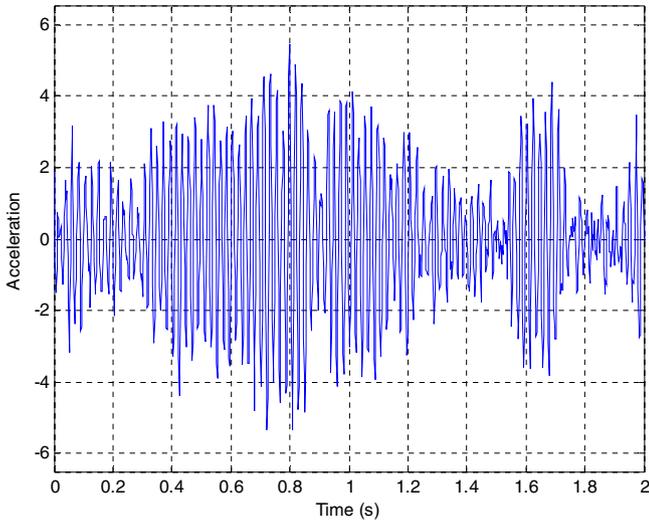


Fig. 10. The compressed acceleration data for the normal pattern.

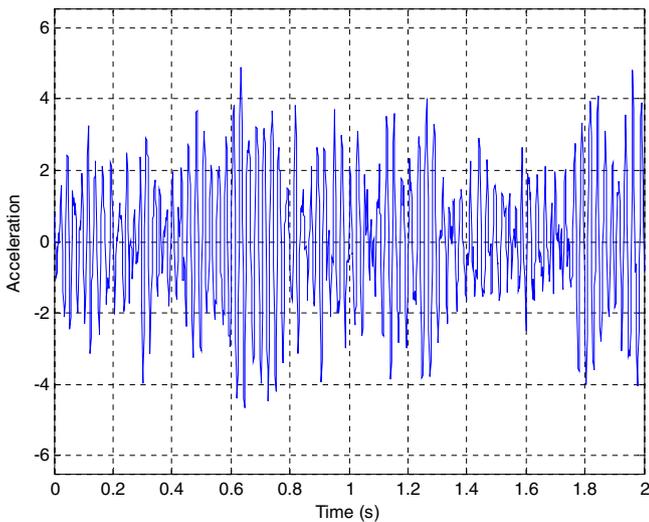


Fig. 11. The compressed acceleration data for the damage pattern 2.

orders as shown in Fig. 12. The AR order is selected to be 20 since the reduction of the value of AIC is small when the AR order is greater or equal to 20 for all data sets.

4.1.3. Classification results

The performance of the AIPR-SDC is validated using the training and classification data described in the previous section. The classification accuracy is compared with other classification algorithms as listed in Table 4, where SVM stands for support vector machines and KNN stands for *k*-nearest neighbor algorithm. The classification success rate is the ratio of correctly classified classification data to the whole set of classification data. The training and classification processes are repeated 20 times, and the average classification success rate is listed in Table 4. The training data are

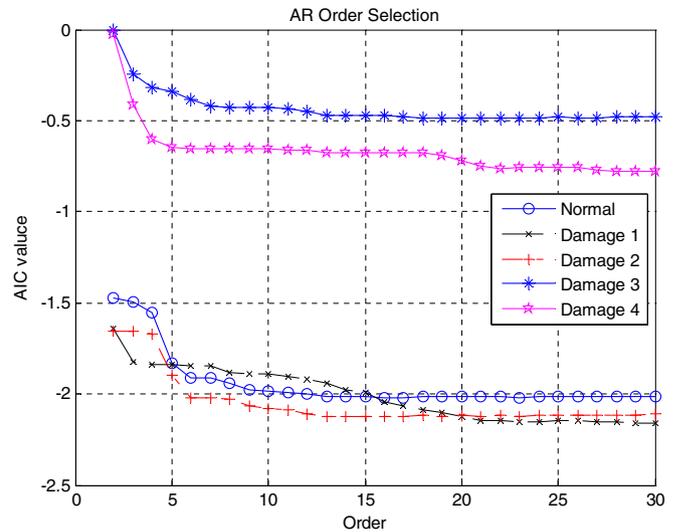


Fig. 12. AR order selection using Akaike's information criterion.

Table 4
Comparison of classification accuracy with other classification algorithms (benchmark structure).

Classification algorithm	AIPR-SDC	SVM	Naive Bayes	KNN-1	KNN-7
Classification success rate	80.2%	85.2%	70%	71%	71%

used in the training process, and the classification data are used in the classification process for calculating the classification success rate. The system parameters selected for the AIPR-SDC are $CR = 8$, $\sigma = 0.5$, $MCRT = 0.985$, and $MCIT = 0.55$. For SVM method, the kernel function is selected to be polynomial; kernel parameter value is 3; and the generalization parameter *C* equals to 3. In Section 4.3, we will discuss the impact of system parameters on

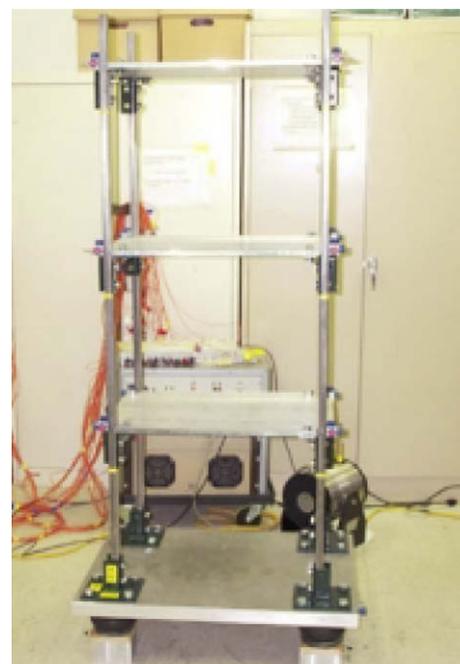


Fig. 13. Three-story frame structure [15].

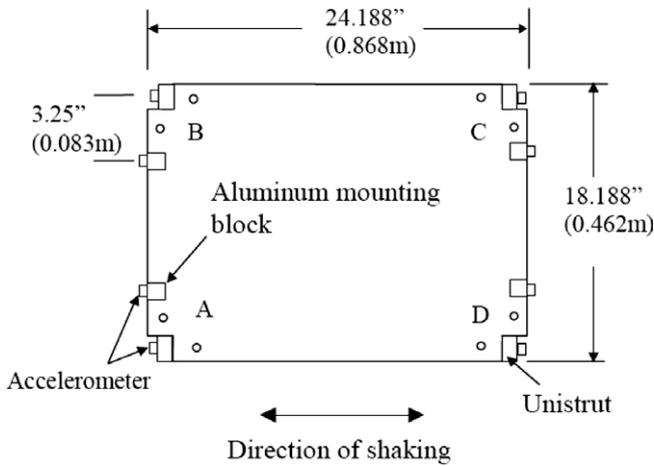


Fig. 14. Floor layout [15].

the classification success rate and the number of memory cells to the AIPR-SDC classifier.

4.2. Damage classification for a three-story frame provided by Los Alamos National Laboratory

4.2.1. Three-story frame structure

The three-story frame shown in Fig. 13 is chosen as a second structure for the validation of the AIPR-SDC algorithm. The details of the structure are given in the online documentation at [15]. A shaker is attached at corner so that both translational and torsional motions can be excited. The structure is instrumented with 24 piezoelectric single axis accelerometers, two per joint as shown in Fig. 14. A number of tests have been conducted by the LANL researchers with different shaker input levels and simulated damages. The acceleration time series of 24 accelerometers are recorded in data files that are available for the download at [15]. These data files are named based on the damage location, damage degree, shaker input level, and the date that the test was conducted.

4.2.2. Damage pattern selection and feature extraction

Four damage patterns listed in Table 5 and the normal pattern are selected for the verification of the AIPR-SDC algorithm. For each pattern, five data files corresponding to 8-V shaker input are used. Each data file contains 24 sensors' data with 8192 number of data points for each sensor. To generate feature vectors for each pattern, the 8192 number of data form 36 of 227-point time series. Time series for 24 accelerometers are reduced to one time series using the PCA method. The compressed 36 time series are then fitted into AR models. Since 36 feature vectors are generated from one file, a total of $36 * 5 = 180$ feature vectors are created for each pattern. For the five patterns used in the validation, the generated feature vectors are $180 * 5 = 900$. One half of the 900 feature vectors (450

Table 5 Selected damage patterns.

Damage patterns	Description
1	The bolts were removed between the bracket and the plate at location 1C
2	The bracket was completely removed at location 1C
3	The bolts were removed between the bracket and the plate at locations 1C and 3A
4	The bracket was completely removed at locations 1C and 3A

Table 6 Comparison of classification accuracy with other classification algorithms (three-story frame structure).

Classification algorithm	AIPR-SDC	SVM	Naive Bayes	KNN-1	KNN-7
Classification success rate	75.2%	70.9%	69.6%	80.3%	81.3%

feature vectors) are used as the training data and the rest are used as the classification data. The selection of the AR order is also based on the AIC method. Applying the strategy followed in the case of the benchmark structure discussed before, the AR order is chosen to be 18 for this structure.

4.2.3. Classification results

The performance of the AIPR-SDC is validated using the training and classification data. The classification accuracy is compared with some of other classifiers, such as SVM, Naive Bayes, and KNN as shown in Table 6. The system parameters selected for the AIPR-SDC are $CR = 8$, $\sigma = 0.5$, $MCRT = 0.99$, and $MCIT = 0.90$. The polynomial kernel function is chosen for SVM classifier with kernel parameter and generalization parameter C equaling to 2.5 and 3, respectively. The classification success rate is the average of 100 training and classification cycles.

4.3. Impact of system parameters on number of memory cells and classification success rate

Figs. 15–20 show the impact of system parameters on the number of memory cells and the classification success rate. The values of parameters used in these plots are $CR = 8$, $\sigma = 0.5$, $MCRT = 0.985$, $MCIT = 0.55$ for the IASC-ASCE structure, and $CR = 8$, $\sigma = 0.5$, $MCRT = 0.985$, $MCIT = 0.95$ for the LANL structure. The memory cell injection threshold, $MCIT$, is an important parameter to control the number of memory cells as shown in Fig. 15. For both IASC-ASCE benchmark structure and LANL three-story frame, the number of memory cells is controlled at a relatively low level when the value of $MCIT$ is greater than a certain value. This $MCIT$ value is 0.6 for the IASC-ASCE benchmark structure and 0.9 for the LANL three-story frame. This result matches the memory cell update algorithm shown in Fig. 6. The bigger the $MCIT$ value, the less chance the candidate memory cell has to

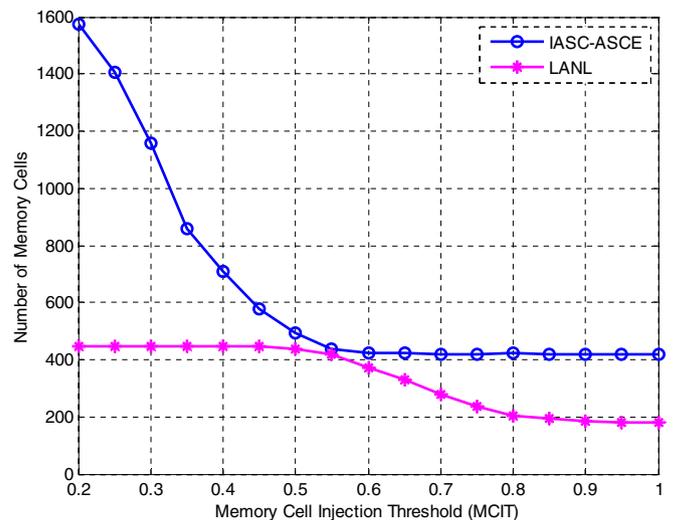


Fig. 15. Number of memory cells vs. MCIT.

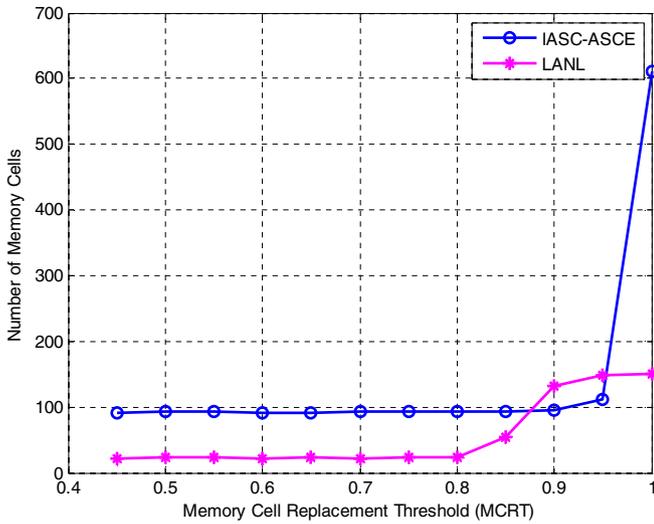


Fig. 16. Number of memory cells vs. MCRT.

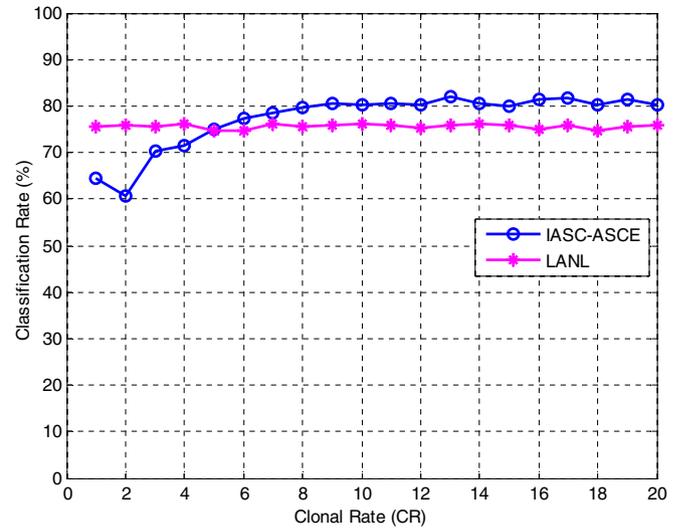


Fig. 19. Classification rate vs. clonal rate.

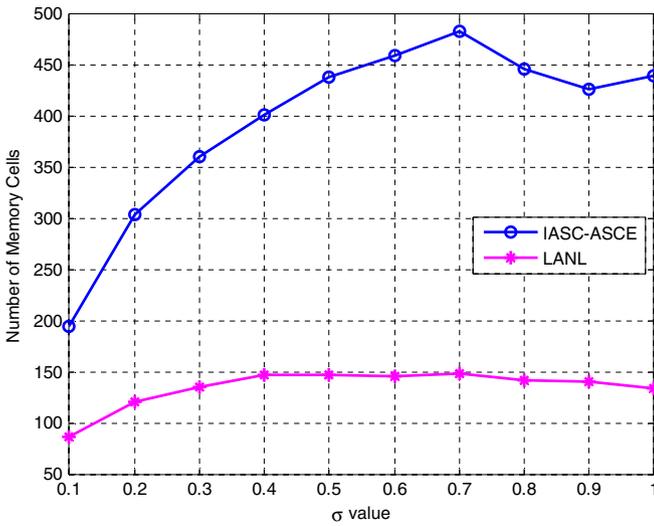


Fig. 17. Number of memory cells vs. σ value.

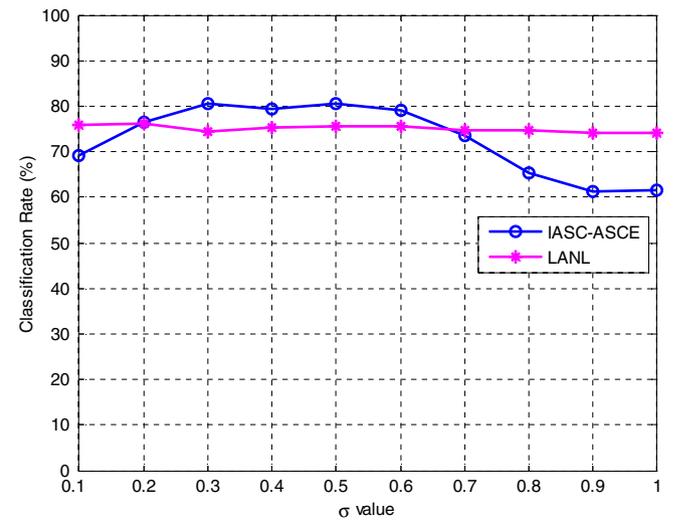


Fig. 20. Classification rate vs. σ value.

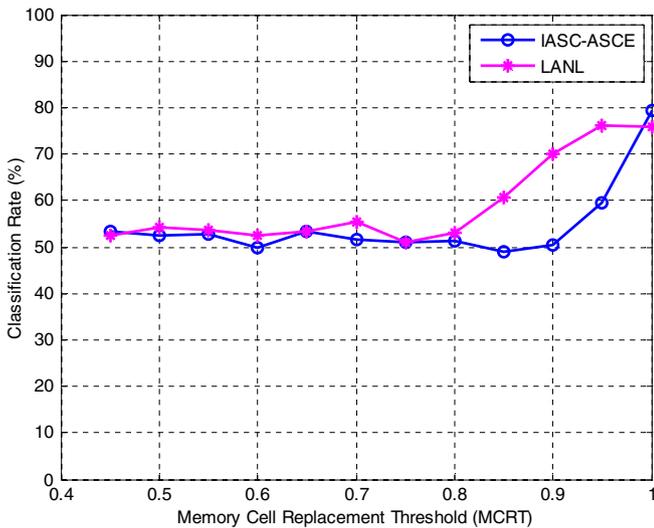


Fig. 18. Classification success rate vs. MCRT.

be injected into the memory cell set. From the memory cell update algorithm, we can also see that when the value of the memory cell replacement threshold, *MCRT*, is close to 1, the candidate memory cell has a higher chance to be added into the memory cell set instead of replacing the match memory cell. As a result, the total number of memory cells increases. This observation is reflected in Fig. 16.

The σ value also affects the number of memory cells as shown in Fig. 17. The parameter σ controls the mutation process, and its value impacts the diversity of the mutated antibody set. When the value of σ is small, the distances between the mutated antibody feature vectors and the original antibody feature vector are very short. The improvement of the diversity of the mutated antibody set is small, which results in a small amount of candidate memory cells are injected into memory cell set. When the value of σ turns a little bigger, the distances between the mutated antibody feature vectors and the original antibody feature vector are getting longer. The number of candidate memory cells to be added into the memory cell set is also increased. When σ reaches to a certain value, the further increase of its value does not have signifi-

cant improvement of the diversity of the antibody set due to the unit hyper-sphere constraint. As a result, the number of memory cells fluctuates.

The value of $MCRT$ has a significant impact on the classification success rate. When the $MCRT$ value is small, the classification rate is only about 50–55%. When the value of $MCRT$ is close to 1, the classification rate rises to 75–80%. The parameter $MCRT$ controls if the candidate memory cell replaces the matched memory cell based on the affinity between the candidate memory cell and the matched memory cell as shown in Fig. 6. When the $MCRT$ is small, the candidate memory cell has a high chance to replace the matched memory cell. This may cause, sometimes, bad candidate memory cells replacing good matched memory cells and result in low classification success rate. When the value of $MCRT$ gets bigger, a candidate memory cell replaces a matched memory cell only when the candidate memory cell has a high affinity with the training antigen and the matched memory cell. In addition, the candidate memory cell having a high affinity with the training antigen will be injected into the memory cell set to increase the diversity of the memory cell set and improve the classification success rate. The drawback of the big $MCRT$ value is that it leads to a big number of memory cells.

Figs. 19 and 20 show the influence of clonal rate CR and σ value to the classification success rate respectively. For the IASC–ASCE benchmark structure, the change of clonal rate from 2 to 8 increases the classification success rate from 60% to 80%. For the LANL three-story frame structure, however, the change of the clonal rate does not have a significant impact on the classification success rate. The similar result is observed for the IASC–ASCE benchmark structure

as shown in Fig. 24 when the number of memory cells is reduced to around 200 (Fig. 22) through the evolution of antibody set using the matched memory cell. The value of σ also affects the classification success rate for the IASC–ASCE benchmark structure. When the σ value is among 0.3–0.6, the classification rate is about 80%. As discussed previously, this range of σ value results in a good distribution of memory cells within the unit hyper-sphere. For the LANL three-story frame structure, the change of the σ value does not have significant impact on the classification success rate.

4.4. The impact of system performance using matched memory cell to evolve antibody population

To investigate the system performance using different antibody evolution approaches, a process to update antibody population through the matched memory cell is added before the training antigen stimulate the antibody set. Since the matched memory cell ($MC_{matched}$) has the highest affinity with the training antigen in the same class, it is anticipated that incorporating offspring and mutated antibodies of the matched memory cell into the antibody set will increase the affinity level of the antibody set with the given antigen. The algorithm of this process is shown in Fig. 21. Using the same method described previously to find the matched memory cell. Then, the matched memory cell is cloned. The number of the cloned memory cell antibodies depends on the Hyper-clonal rate HCR and the clonal value CV . Similar to the antibody set, the higher the affinity, the larger the number of antibodies is cloned. The number of cloned memory cell antibodies $CloneNumber$ can be calculated by Eq. (21)

```

Given an antigen  $ag$ , do{
     $class = ag.c$ ;
     $MC_{matched} = \arg \max_{mc \in MCS_{class}} aff(ag, mc)$ ;
     $CV = aff(MC_{matched}, ag)$ ;
     $MV = 1 - aff(MC_{matched}, ag)$ ;
     $CloneNumber = HCR * CV$ ;
     $CloneNumber = round(CloneNumber)$ ;
    for ( $i = 1, i++, i < CloneNumber$ ) {
         $mc_{clone} = MC_{matched}$ ;
         $mc_{mutated}.c = class$ ;
         $mc_{mutated}.f = mc_{clone}.f + MV * \vec{N}(0, \sigma^2)$ ;
        if ( $norm(mc_{mutated}.f) > 1$ )
             $mc_{mutated}.f = (norm(MC_{matched}.f) + rand()) * (1 - norm(MC_{matched}.f)) * \left( \frac{mc_{mutated}.f}{norm(mc_{mutated}.f)} \right)$ ;
         $ABS_{class} = ABS_{class} \cup mc_{mutated}$ 
    }
}

```

Fig. 21. Evolve antibody set using matched memory cell.

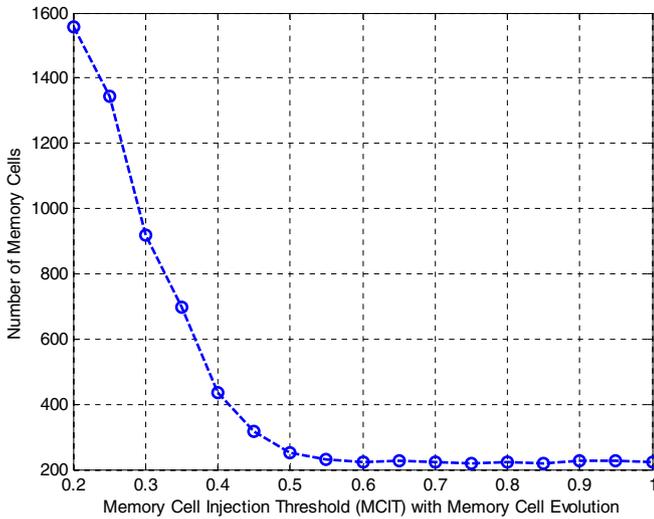


Fig. 22. Number of memory cells with the stimulation of matched memory cell.

$$CloneNumber = round(HCR * CV) = round(HCR * aff(MC_{matched}, ag)). \tag{21}$$

The cloned memory cell antibodies are mutated to increase the diversity of the antibody set (note that the cloned memory cell antibodies will be added into the antibody set). The mutation is performed by mutating the feature vectors of the cloned memory cell antibodies as shown in Eq. (22)

$$mc_{mutated}.f = MC_{matched}.f + MV \times \phi. \tag{22}$$

For more details regarding mutation and norm check, please refer to the antibody set mutation process. The cloned and mutated memory cell antibodies are injected into the antibody subset to which the given antigen belongs to.

The results of adding the stimulation of matched memory cell to the antibody set are shown in Figs. 22–25 for the benchmark structure. Each point in these figures is the average of 10 training and classification cycles. From these figures, we can see that the stimulation of matched memory cell to the antibody set helps to decrease the number of memory cells. The classification success rate, however, drops a little bit when this process is introduced. The reason is that the candidate memory cells are very likely

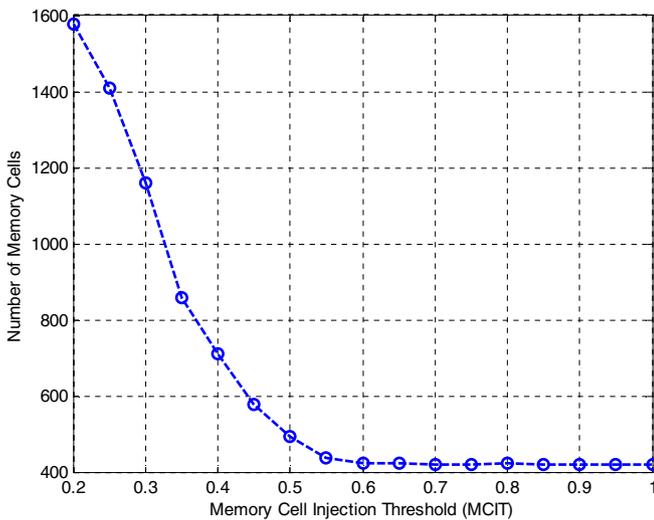


Fig. 23. Number of memory cells without the stimulation of matched memory cell.

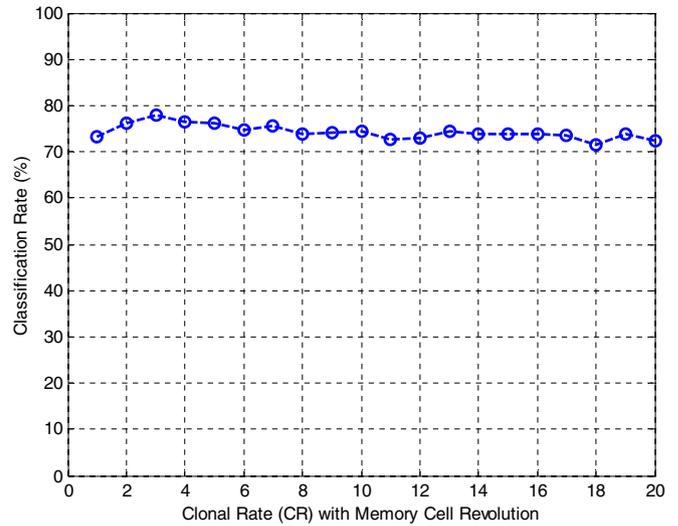


Fig. 24. Classification rate with the stimulation of matched memory cell.

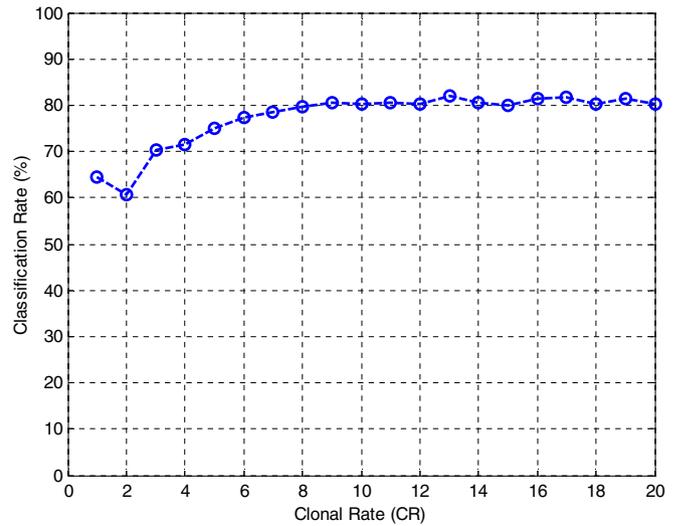


Fig. 25. Classification rate without the stimulation of matched memory cell.

selected from the mutated antibodies of the matched memory cells in this process. These candidate memory cells may be injected into the memory cell set or used to replace the matched memory cells. Since the candidate memory cells are closely related to the original matched memory cells, the introduction of these candidate memory cells to the memory cell set do not improve the diversity of the memory cell set significantly.

5. Conclusions

A classification algorithm, inspired by the natural immune system, for classifying structure damages is presented in this paper. The presented classifier is designed based on the novel immune system characteristics such as adaptation, evolution, and immune learning. The evolution and immune learning algorithms make it possible for the classifier to generate a high quality memory cell set for recognizing various structure damage patterns. The AIPR-based structure damage classifier has been used to classify structure damage patterns using a benchmark structure proposed by the IASC–ASCE SHM Task Group and a three-story frame provided

by Los Alamos National Laboratory. The validation results show that the AIPR-based pattern recognition is suitable for structure damage classification. The verification results also show that some of the system parameters have crucial impacts on the classification performance and the number of memory cells generated for the classification. The comparison study of the classification accuracy to other classifiers has also been conducted. For the benchmark structure, the AIPR-SDC has a higher classification success rate comparing to Naive Bayes classifier and KNN method, and a lower success rate comparing to the SVM. For the LANL three-story frame, the AIPR-SDC has a higher classification success rate comparing to SVM and Naive Bayes classifiers. However, the classification success rate of the KNN method is better than that of the AIPR-SDC.

References

- [1] Chang PC, Flatau A, Liu SC. Review paper: health monitoring of civil infrastructure. *Struct Health Monitor* 2003;2:257–67.
- [2] Kirk RS, Mallett WJ. Highway bridges: conditions and the federal/state role, 2007.
- [3] Lynch JP. An overview of wireless structural health monitoring for civil structures. *Philos Trans A: Math Phys Eng Sci* 2007;365:345–72.
- [4] Kolakowski P. Structural health monitoring – a review with the emphasis on low-frequency methods. *Eng Trans, IPPT* 2007;55:239–75.
- [5] Doebling SW, Farrar CR, Prime MB, Shevitz DW. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review, Los Alamos National Laboratory Report LA-13070-MS, 1996.
- [6] Sohn H, Farrar CR, Hemez FM, Czarnecki JJ, Shunk DD, Stinemates DW, et al. A review of structural health monitoring literature: 1996–2001, Los Alamos National Laboratory Report LA-13976-MS, 2003.
- [7] Sohn H, Farrar CR. Damage diagnosis using time series analysis of vibration signals. *Smart Mater Struct* 2001;10:446–51.
- [8] da Silva S, Dias Jr M, Lopes Jr V. Damage detection in a benchmark structure using AR-ARX models and statistical pattern recognition. *J Brazil Soc Mech Sci Eng* 2007;29:174–84.
- [9] Lee JJ, Lee JW, Yi JH, Yun CB, Jung HY. Neural networks-based damage detection for bridges considering errors in baseline finite element models. *J Sound Vib* 2005;280:555–78.
- [10] Ni YQ, Zhou XT, Ko JM. Experimental investigation of seismic damage identification using PCA-compressed frequency response functions and neural networks. *J Sound Vib* 2006;290:242–63.
- [11] Kao CY, Hung SL. Detection of structural damage via free vibration responses generated by approximating artificial neural networks. *Comput Struct* 2003;81:2631–44.
- [12] de Castro LN, Von Zuben FJ. Learning and optimization using the clonal selection principle. *IEEE Trans Evolut Comput* 2002;6:239–51.
- [13] Watkins A, Timmis J, Boggess L. Artificial immune recognition system (AIRS): an immune-inspired supervised learning algorithm. *Genet Program Evolvable Mach* 2004;5:291–317.
- [14] Structural health monitoring benchmark problem. <<http://mase.wustl.edu/wusceel/asce.shm/benchmarks.htm>>.
- [15] Test Structures and Applications in Los Alamos National Laboratory: 3 story structure. <http://www.lanl.gov/projects/damage_id/data.shtml>.
- [16] de Castro LN. Fundamentals of natural computing: basic concepts, algorithms, and applications. Chapman & Hall/CRC; 2006.
- [17] Castiglione F, Motta S, Nicosia G. Pattern recognition by primary and secondary response of an Artificial Immune System. *Theory Biosci* 2001;120:93–106.
- [18] Carter JH. The immune system as a model for pattern recognition and classification. *J Am Med Inform Assoc* 2000;7:28–41.
- [19] de Castro LN, Timmis J. Artificial immune systems: a new computational intelligence approach. Springer; 2002.
- [20] Lanaridis A, Karakasis V, Stafylopatis A. Clonal selection-based neural classifier. In: 2008 8th international conference on hybrid intelligent systems (HIS), 2008. p. 655–60.
- [21] Zhong YF, Zhang LP, Huang B, Li PX. An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery. *IEEE Trans Geosci Remote Sens* 2006;44:420–31.
- [22] Zhong YF, Zhang LP, Gong JY, Li PX. A supervised artificial immune classifier for remote-sensing imagery. *IEEE Trans Geosci Remote Sens* 2007;45:3957–66.
- [23] Polat K, Gunes S, Tosun S. Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing. *Pattern Recognit* 2006;39:2186–93.
- [24] Theodoridis S, Koutroumbas K. Pattern recognition. Academic Press; 2008.
- [25] Johnson EA, Lam HF, Katafygiotis LS, Beck JL. A benchmark problem for structural health monitoring and damage detection. In: Proceedings of 14th engineering mechanics conference, Austin, Texas, 2000.