Artificial Immune System: An Effective Way to Reduce Model Overfitting

Waseem Ahmad[1(✉)] and Ajit Narayanan[2]

[1] International College of Auckland, Auckland, New Zealand
wahmad@ica.ac.nz
[2] AUT University, Auckland, New Zealand
ajit.narayanan@aut.ac.nz

Abstract. Artificial immune system (AIS) algorithms have been successfully applied in the domain of supervised learning. The main objective of supervised learning algorithms is to generate a robust and generalized model that can work well not only on seen data (training data) but also predict well on unseen data (test data). One of the main issues with supervised learning approaches is model overfitting. Model overfitting occurs when there is insufficient training data, or training data is too simple to cover the structural complexity of the domain being modelled. In overfitting, the final model works well on training data because the model is specialized on training data but provides significantly inaccurate predictions on test data due to the model's lack of generalization capabilities. In this paper, we propose a novel approach to address this model overfitting that is inspired by the processes of natural immune systems. Here, we propose that the issue of overfitting can be addressed by generating more data samples by analyzing existing scarce data. The proposed approach is tested on benchmarked datasets using two different classifiers, namely, artificial neural networks and C4.5 (decision tree algorithm).

Keywords: Artificial immune system · Antibodies · Model overfitting · Artificial neural networks · Vaccination · Ensemble learning

1 Introduction

Data Mining is a subfield of machine learning where hidden patterns and rules are discovered from raw data. Data mining techniques can mainly be grouped into two categories, namely, supervised and unsupervised learning. In supervised learning, a model is built based on mapping input patterns to a desired output. Data consists of a set of features or variables as well as output (desired) variables. The task of any supervised learner is typically to build a model that both best fits a number of training examples (training data, which can include secondary training data for recalibrating a neural network once initially trained) and predicts the class into which new or unseen data falls, where the class information of the predicted instances may not be known (test data). On the other hand, in unsupervised learning, data without any class information is grouped in such a way that instances present in each cluster has a maximum similarity and retain a maximum dissimilarity among different clusters.

M. Núñez et al. (Eds.): ICCCI 2015, Part I, LNAI 9329, pp. 316–327, 2015.
DOI: 10.1007/978-3-319-24069-5_30

The effectiveness of supervised learning algorithms has been well demonstrated in the literature [1-3]. These algorithms perform well given reasonably large sets of training data. However, the predictive capabilities of supervised learning algorithms deteriorate substantially when there is small pool of training data. In these situations, generated models suffer from data overfitting. Models generated by considering only fewer training samples hold poor generalization capabilities (high degree of specialization on seen data), hence they demonstrate higher accuracy on training data and poor predictive accuracy on unseen or test data. There are a number of approaches to deal with this data ovefitting such as regularization, early-stopping and different forms of cross-validation [4, 5]. In this paper, we are proposing a novel approach to deal with data overfitting that is inspired by the Natural Immune System (NIS) and more specifically from the process of vaccination.

Artificial Immune System (AIS) is a computing paradigm inspired by the processes and metaphors of the Natural Immune System (NIS). In recent years researchers have developed many computational algorithms by borrowing concepts of Natural Immune Systems [6-11]. NIS protects us from various harmful diseases throughout our life span. The increasing interest of computational community to explore NIS is due to its adaptation and generalization capabilities. NIS learns from captured harmful instances in such a way that it provides lifelong immunity as well as provides protection against any mutated and stronger versions of the pathogens that can cause deadly diseases. Vaccination is an example of such processes of NIS that has adaptation and generalization capabilities. The underpinning principle of vaccination is to learn from weaker strands of viruses and generate a pool of antibodies and memory cells so that the human body is well equipped to handle a stronger version of the virus.

In NIS, memory cells are solutions for keeping a history of past pathogens and viral attacks. On the other hand, antibodies, which are cloned and mutated copies of stimulated B-cells, provide protection against variants of invading pathogens. Memory cells and antibodies are regarded as specialized and generalized cells, respectively. This process of generation of generalized antibodies provides a way for our NIS to avoid overfitting. In this paper, we are proposing that one way of reducing overfitting is through the generation of antibody equivalents as part of modelling.

All AIS approaches presented in literature produce memory cells that are a summarized form of training data. Memory cells can be defined as given a data consisting of n instances, find g number of instances that provide minimum information loss ($g < n$). These memory cells are proven to have efficient specialization capabilities and results in better model accuracy. This data specialization leads to better generalization capabilities on unseen data, given an adequate amount of training data. The contribution of memory cells is well demonstrated in AIS literature, but less explored is the role of antibodies in AIS approaches. Antibodies are cloned and mutated copies of existing data samples. The population of antibodies are known to have great generalization capabilities, but little research work has been undertaken to prove this claim. The work presented in this paper is based on the generation of antibodies to achieve robust and generalized classification models.

In [12] it was demonstrated that concepts of vaccination can be applied to computing algorithms to obtain robustness and an increasing degree of generalization.

A hybrid architecture was proposed where AIS algorithm was used to generate vaccine (in the form of memory cells or antibodies) and Artificial Neural Network (ANN) was used as a learning system. ANN was primed (trained) on vaccine before injecting and retraining on original data. In experimental results, it was successfully demonstrated that primed ANN was well equipped to mount a faster and more effective response when original data was introduced to the learning system. This means ANN required fewer learning cycles to converge to a local solution and often results in better models (fewer learning errors on training data). That work concluded that memory cells and antibodies have excellent data summarization capabilities and can be used as vaccination agents for other learning systems.

The proposed architecture in [12] was tested and showed superior classification results only against training data. No efforts were made to explore the generalization capabilities of such models on unseen or test data. In this paper, we are extending earlier work done in [12] to demonstrate that models generated using antibodies have good generalization capabilities and provide superior predictive results on unseen data. In the original published work [12], random mutation was used to create antibodies. Our initial experiments demonstrated that random mutation introduced a high degree of noise and resulted in poor predictive results. Therefore, this paper proposes a directed mutation approach based on the local and global structural information present in the training data.

2 Literature Review

In the last decade, AIS has been a focus of many computational researchers and many novel machine learning algorithms have been proposed. The main idea is to represent data instances as antigens and antibodies or B-cells as clusters. One of the first AIS algorithms was proposed by De Castro and Zuben [13] called Artificial Immune Network (aiNet). It utilizes the concepts of memory cells, antibodies, clonal selection and hypermutation. The proposed algorithm was used to perform data summarization by producing memory cells and to find natural clusters in the data. The AIS algorithms in the area of unsupervised learning can be found [9, 14-16]. The most significant work in the AIS supervised learning domain is proposed by Watkins *et al* [6] that used the concept of artificial recognition balls (ARBs), resource limitation, memory cell and hypermutation. The population of memory cells compete for survival and principles of natural selection are used to evolve the population of memory cells. Finally, a population of evolved memory cells is obtained, which are used to predict class membership on unseen data using a KNN algorithm. HAIS [17] is another AIS based classifier for supervised learning and is inspired by the humoral mediated response triggered by an adaptive immune system. HAIS uses core immune system concepts such as memory cells, plasma cells, Igs, antibodies and B-cells as well as parameters such as negative clonal selection and affinity thresholds.

In the field of optimization, Woldemariam [18] introduced the notion of vaccination in an AIS algorithm to solve function optimization problems. Vaccine is extracted by dividing the search space into equal subspaces. These vaccines are then intro-

duced into the algorithm to enhance the exploration of global and local solutions. The work done in the field of optimization using AIS algorithms can be found in [19-21]. In [12], concepts of vaccination were used to improve the accuracy and learning capabilities of artificial neural networks. A hybrid architecture was proposed based on AIS and ANN. The ANN was introduced to mimic the behavior of a learning system. AIS algorithm was used to extract memory cells or antibodies (called vaccine) from the data set. This vaccine was later introduced into the learning system (ANN) to demonstrate that the vaccine primed the learning system to handle a stronger version of the viral attack and as a consequence learning on data set not only became much faster but also provided better classification accuracy.

3 Proposed Approach

The components of NIS such as B-cells, memory cells and antibodies play an important role in keeping us alive and eliminating dangerous invaders. This process of antibody generation forms the basis of the computing algorithm proposed in this paper. In earlier published work [12], the robustness of the model built using memory cells and antibodies was measured using a three step methodology. In the first step, memory cells or antibodies were obtained from full labelled data using AIS algorithm. In step 2, ANN classifier was used to train on memory cells/antibodies and stopped once convergence was achieved on error sum of square (ESS). Finally, ANN was further trained on original full data until convergence was achieved. This three step methodology demonstrated that the final model had superior classification results on full (seen) data than the model built directly on full data.

According to this methodology, classification results are derived only from training data (model fitting) and no unseen data was used to evaluate the model's predictive capabilities. Generally, algorithms and data models are compared based on their predictive accuracy on unseen data. The common practice is to divide the data into training data (including recalibration data) and genuine, unseen test data. Models are built by single-shot or repeated exposure on training data while its accuracy and robustness are observed and measured on test data. In order to validate the contribution of vaccination concepts in the field of machine learning, the model's robustness or generalization capabilities must be evaluated on test data and not just on training data. In this paper, full data is split into training data and test data, and the same three step approach is followed. In step 1, antibodies are generated using only training data. In step 2 and 3, a classifier is used to build a model with these antibodies and the model is evaluated on full data. The rational of using full data and not just test data for the final evaluation is to incorporate any training errors obtained during the process of antibody generation. This process is shown in the form of following three steps.

$$Full\ data = training\ data + test\ data$$
$$\textbf{\textit{Step}}\ \mathbf{1}\text{:}\ \ antibodies = AIS\ (training\ data)$$
$$\textbf{\textit{Step}}\ \mathbf{2}\text{:}\ \ Model = \ Classifier\ (antibodies)$$
$$\textbf{\textit{Step}}\ \mathbf{3}\text{:}\ Evaluate = Model\ (Full\ data)$$

The results obtained from Step 3 are compared against step B, where the model is built on training data (Step A) and then evaluated on full data (Step B). The following section will explain the algorithm for generating antibodies.

$$\textbf{\textit{Step A}}: \ Model = Classifier\ (training\ data)$$
$$\textbf{\textit{Step B}}: \ Evaluation = Model\ (Full\ data)$$

3.1 Generation of Antibodies

The generation of antibodies is an important component of the proposed approach. The complete process is divided into three main stages to be described in more detail below:

$\textbf{\textit{Stage 1}}$: *Generate antibodies based on localised knowledge (Local expansion)*
$\textbf{\textit{Stage 2}}$: *Generate antobodies based on global knowledge (Global Expansion)*
$\textbf{\textit{Stage 3}}$: *Perform negative clonal selection*

In stage 1, a clustering algorithm is used to group each class of data samples into predefined number of clusters. This step is followed by generating antibodies based on each obtained cluster (performing local expansion on each cluster). In stage 2, new data instances are generated based on the structure of data present in each class. Finally, in stage 3, the data obtained in earlier two stages is aggregated and negative clonal selection to prune similar data or clusters [17] is performed to obtain a final set of antibodies (new data set). The antibodies generated by this approach are used to build a classification model. Table 1 states the parameters used in the proposed algorithm. Stages 1 and 2 are important to capture the structural composition of each class, separately.

Table 1. Description of parameters used in the proposed algorithm

Parameters	Description
n	Number of antibodies produced against each data sample
k	Number of clusters for each class
w	Localized cluster size threshold
globalExp	Number of antibodies generated by Global Expansion Step
glVariation	Controls each class's covariance matrix values
singlePointVariation	Controls variance values on non-cluster data instances
T	Negative Clonal Threshold

Stage 1: Local Expansion
The aim of local expansion is to generate new data samples (Fig. 1) against the training data of each class (generating localized antibodies). K-means clustering algorithm is used to find natural groupings within each class. Here, each cluster can be seen as an activated B-cell that produces antibodies based on its localized state. A Matlab function of multivariate normal random numbers generator (*mvnrnd*) is used to generate new data samples (antibodies) against each cluster (Fig. 1). The function *mvnrnd* takes the mean and covariance of data and returns a matrix of random numbers chosen from a normal distribution. However, if the size of cluster is less than 3, then

normrnd function is used to generate data samples. The function *normrnd* takes into account the mean and variance of data and returns a matrix of random numbers chosen from normal distribution.

```
Input: data: k, w, n, singlePointVariation, glVariation
Output: newData1
Kmeans Clustering (data, k)
For each data cluster x
   If size(x) > w
        newData1 = mvnrnd(mu(x), cov(x)*lmVariation, size(x)*n)
   Else
        newData1 = normrnd(mu(x), singlePointVariation, n)
   End If
End For
```

Fig. 1. Pseudo-code for generating data samples based on local expansion

Stage 2: Global Expansion

The aim of global expansion is to generate data for each labelled class separately. The Matlab function of multivariate normal random number generator (*mvnrnd*) is used for global expansion (Fig. 2). The data from stage 1 and 2 (local and global expansion) are aggregated to obtain a pool of antibodies (Eq. 1).

$$newData = newData1 \cup newData2 \tag{1}$$

```
Input: data, gmVariation, globalExp
Output: newData2
For each class
   newData2 = mvnrnd(mu(data), cov(data)*gmVariation, size(data)*
globalExp)
```

Fig. 2. Pseudo-code for generating data samples based on global expansion

Stage 3: Negative Clonal Selection

Negative clonal selection is performed on all newly generated data instances of each class, separately. This stage starts by assigning full labelled data to *self-data* and all instances of *newData* (Eq. 1) are compared one-by-one. If any instance of *newData* is too similar to existing *self-data*, it is removed, otherwise added into pool of *self-data*. After the end of this process, the final pool of *self-data* is the final set of antibodies used to build classification model. The similarity measure is calculated using pairwise Euclidean distance measure.

3.2 Example

A simulated data is used to demonstrate the proposed approach. The simulated data contains a total of 30 instances, where 12 instances belong to class 1 and 18 to class 2 (Fig. 3-A). Local and Global expansions are performed in the proposed approach using parameter values of k = [5, 5], n = [5, 5], w = 3, *globalExp* = [2, 2], *glVariation* = [1, 1] and T = [0.25, 0.25] and can be seen in Fig. 4-B and 4-C, respectively. The final pool of antibodies (*newData*) is shown in Fig. 3-D that is aggregation of local and global expansion followed by performing negative clonal selection. Local expansion is responsible for the generation of data instances based on internal clusters found in each class, whereas, global expansion is driven by the structure (mean and covariance) of the respective classes. In this paper, our emphasis is to demonstrate with empirical findings that models obtained using data instances of Fig. 3-D are more robust and generalized than model obtained using data of Fig. 3-A.

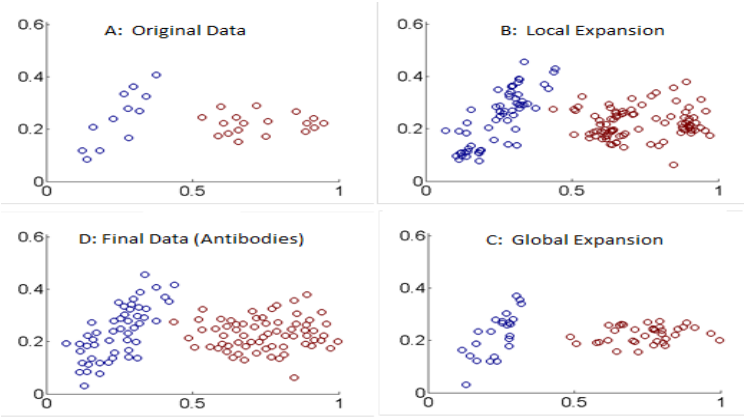

Fig. 3. Simulated 2-dimesional data highlighting the steps of proposed approach

4 Experimental Results

Four well-known and researched datasets have been selected to demonstrate the effectiveness of the proposed approach. The details of these datasets are provided in Table 2. The datasets are divided into training and test data. For experimental purposes, only those training and test data partitioning are selected which have the highest number of test errors while generating a classification model on training data. The antibodies are generated only from training data and then these antibodies are used to generate classification models. The model evaluation is conducted by obtaining classification accuracy on full data that is the aggregation of training data and test data. As stated previously, the rational of this arrangement is to avoid better classification accuracy on the test data at the cost of higher training error. Two classifiers, namely, artificial neural networks and C4.5 (J48) are used in these experiments. These experiments are conducted in Weka-3.6 [22]. The experimental methodology can be highlighted in following equations.

$$full\ data = training\ data\ + test\ data$$
$$antibodies = AIS\ (training\ data)$$
$$Model = ANN\ (antibodies) \quad OR \quad J48\ (antibodies)$$
$$Evaluation = Model\ (Full\ data)$$

The contribution of ANN in the field of machine learning and pattern recognition is well established [23, 24]. Artificial neural networks (ANNs) are computational models inspired by the functionality of biological neuron [25]. In the paper, Backpropagation algorithm is used in ANN. In Backpropagation, neurons are arranged in layers and send their signals forward and then error is calculated at the output layer that is propagated backward. Another classifier used here is J48 or C4.5 that is a decision tree based classification algorithm [26]. J48 builds the decision tree on the training data using the concept of information entropy.

All four data sets, namely, Heart, Breast cancer, Gene expression and Iris were used to demonstrate the applicability of the proposed approach. These datasets contain varying degree of complexity in terms of the number of features, instances and classes. These datasets (except gene expression dataset) were taken from UCI Machine Learning repository and further detail regarding these datasets can be found at [27]. Gene expression dataset originally had 254 features, excluding class labels. A Feature selection algorithm was used to select 14 features. The datasets were divided into training and test data. It can be seen from Table 2 that after the split the training data has two to three times fewer instances than test data. Approximately 30% of the total data was used to build a classification model and the remaining data was used to evaluate the classification accuracy. The rationale of using this arrangement is to demonstrate that given fewer training samples, the proposed approach can provide good classification models. The number of features stated in the Table 2 are the number of attributes without class label. The values of parameters used are described in Table 3. Various runs are performed to finally select these parameter values. Multiple values for each parameter describe values based on each class.

Table 2. Datasets information explaining training and test data split used in this paper

Datasets	Training data	Test data	Full data	Features	Classes
Heart	80	187	267	44	2
Breast Cancer	58	136	194	34	2
Gene Expression	31	71	102	14	2
Iris	45	105	150	4	3

The empirical results of all four datasets are presented in Table 4 using ANN classifier. The second row in Table 4 represents ANN classifier results when trained on training data alone and evaluated on full data. Row 3 onwards represents ANN classifier results when antibodies are used to train ANN and full data is used for evaluation. On Iris dataset, 14 classification errors were found on full data when ANN was trained only on training data. The selected training data was used to obtain 50 separate pools of antibodies. The results obtained from each pool of antibodies are presented in Table 4. The results are summarized and compared using mean, standard

deviation, maximum and minimum classification errors obtained. The maximum and minimum classification errors found for Iris data were 15 and 6, respectively, whereas mean and standard deviation were 10.2 and 2.6, respectively. The minimum classification errors obtained on Iris data is lower when antibodies were used to build the model than when the model was built only using the training data. The higher fluctuation in the classification errors of the Iris data is due to the stochastic nature of the generation of antibodies. Similar empirical results can be seen on other datasets. The best results were obtained on the Heart dataset, where the classification accuracy was improved by more than 50%. The classification errors decreased from 119 to 64. Another noticeable result was the maximum errors obtained on all datasets that were close to the errors obtained when only training data was used to build the ANN classifier.

Table 3. Parameter settings used for each dataset

Parameters	Heart	Breast Cancer	Gene Expression	Iris
K	9, 9	10, 10	5, 5	5, 5, 5
n	10, 7	15, 10	12, 12	5, 5, 5
w	3	3	3	3
globalExp	10, 5	5, 0	2, 2	2, 2, 2
glVariation	1, 0.5	0.5, 0.25	0.65, 0.65	0.5, 0.5, 0.5
singlePointVariation	5%, 2.5%	5%,2.5%	5%, 5%	5%, 5%, 5%
T	0.35, 0.25	0.35, 0.35	0.75, 0.75	0.05, 0.05, 0.05

Table 4. Classification results (errors) obtained using ANN on all datasets

	Iris	**Heart**	**Breast cancer**	**Gene expression**
ANN on Training data	14	119	49	12
Mean	10.2	85.6	39.4	10.1
STD	2.6	10.1	3.2	1.7
Maximum	15	109	47	14
Minimum	6	64	31	6

The classification errors on Iris data using J48 classifier are shown in Table 5. The classifier produced 16 classification errors on full data when the decision tree was built on training data (Row 2, Table 5). Row 3 onwards represents J48 classifier results when antibodies were used to build classification tree and full data was used for evaluation. Mean, standard deviation, maximum and minimum classification errors over 50 iterations were 9.8, 2.76, 18 and 4, respectively. These results are consistent with earlier results of the ANN classifier. The number of leaves and tree sizes increased when the model was built using antibodies. Antibodies add complexity to the training data and therefore the resulting tree will have more leaves and a larger tree size than trees generated by training data only. Similar empirical results can be observed on Heart, Breast cancer and Gene expression datasets in Table 6.

There has been a significant variance in the classification results obtained by both classifiers (J48 and ANN). This variance can be reduced using the ensemble method. The ensemble experiments were conducted on the Iris dataset. Fifty models using J48 algorithm are generated and a simple majority vote method is used to assign class membership to full data instances. Eight classification errors were recorded when the ensemble based J48 classifier was used. When the same ensemble principles were used on ANN, the classifier achieved nine classification errors. However, when an ensemble classifier was built on aggregating both ANN and J48 (100 populations), seven (7) classification errors were obtained (Fig. 4). The classification error value of seven is still higher than the minimum errors obtained in above experiments (4 and 6 errors for J48 and ANN, respectively). However, this ensemble method has removed a higher degree of variance in classification results.

Table 5. Classification errors obtained using J48 classifier on Iris data

	Errors	**Leaves**	**Tree Size**
Training data	16	3	5
Mean	9.18	6.04	11.08
STD	2.76	1.55	3.10
Maximum	18	9	17
Minimum	4	4	7

Table 6. Classification errors using J48 on Heart, Breast cancer & Gene expression data

	Heart Errors	**Cancer Errors**	**Gene Errors**
Training data	95	43	26
Mean	68.30	40.76	20.40
STD	11.55	6.03	5.50
Maximum	89	32	40
Minimum	25	56	10

5 Conclusion

In this paper, a novel computing algorithm inspired by the generation of generalized antibodies is proposed to introduce robustness and generalization in data. These antibodies have been able to reduce overfitting on the final model on four well-established datasets. Two classifiers, namely, ANN and C4.5, are successfully used to validate the effectiveness of the proposed approach. We have demonstrated with experimental results that both classifiers produced poor prediction accuracy on test data when a small number of training data was used. However, the pool of antibodies using the same classifiers leads to improved prediction accuracy and to more robust and generalized models.

In the future, a three way split (training, validation and test data split scheme) must be used to further evaluate the effectiveness of the proposed algorithm. The validation set can also be used to optimize parameter values. This proposed approach can form the basis of novel semi-supervised learning algorithm. Semi supervised learning can be an ideal platform to develop this approach further to evaluate its strengths.

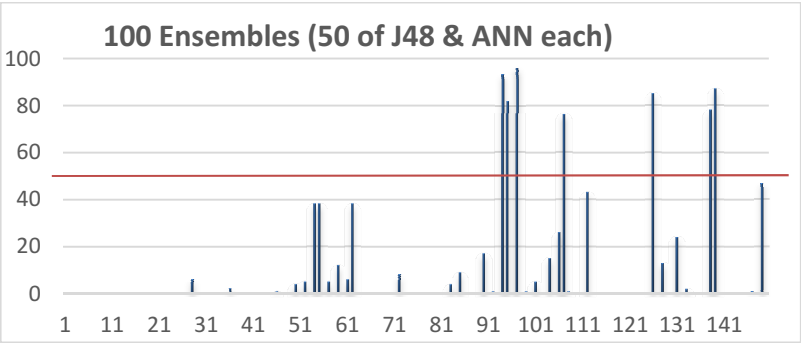

Fig. 4. Ensemble classifier obtained by using 100 models of J48 and ANN. Each horizontal bar represents counts of classification error obtain against each data instance across all models.

References

1. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. Informatica (Ljubljana) **31**, 249–268 (2007)
2. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery **2**(2), 121–167 (1998)
3. Murthy, S.K.: Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery **2**(4), 345–389 (1998)
4. Moore, A.W.: Cross-validation for detecting and preventing overfitting. School of Computer Science Carneigie Mellon University (2001)
5. Larose, D.T.: k-Nearest Neighbor Algorithm. Discovering Knowledge in Data: An Introduction to Data Mining, 90–106 (2005)
6. Watkins, A., Timmis, J., Boggess, L.: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. Genetic Programming and Evolvable Machines **5**(3), 291–317 (2004)
7. Ahmad, W., Narayanan, A.: Outlier detection using humoral-mediated clustering (HAIS). In: Proceedings of IEEE World Congress on Nature and Biologically Inspired Computing, NaBIC2010, pp. 45–52 (2010)
8. Castro, L.N.D., Zuben, J.: The clonal selection algorithm with engineering applications. In: Workshop Proceedings of GECCO, Workshop on Artificial Immune Systems and Their Applications, Las Vegas, pp. 36–37 (2000)
9. Li, X., et al.: ICAIS: a novel incremental clustering algorithm based on artificial immune system. In: International Conference on Internet Computing in Science and Engineering, pp. 85–90 (2008)
10. Liu, Z., et al.: FAISC: a fuzzy artificial immune system clustering algorithm. In: Third International Conference on Natural Computation (ICNC 2007) (2007)
11. Yap, F.W., Koh, S.P., Tiong, S.K.: Mathematical Function Optimization using AIS Antibody Remainder method. International Journal of Machine Learning and Computing **1**(1), 13–19 (2011)
12. Ahmad, W., Narayanan, A.: Principles and methods of artificial immune system vaccination of learning systems. In: Liò, P., Nicosia, G., Stibor, T. (eds.) ICARIS 2011. LNCS, vol. 6825, pp. 268–281. Springer, Heidelberg (2011)

13. Castro, L.N.D., Zuben, F.J.V.: aiNet: an artificial immune network for data analysis. In: Abbass, H., Sarker, R., Newton, C. (eds.) Data Mining: A Heuristic Approach, ch012, pp. 231–260. Idea Group Publishing (2002). doi:10.4018/978-1-930708-25-9
14. Younsi, R., Wang, W.: A new artificial immune system algorithm for clustering. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) IDEAL 2004. LNCS, vol. 3177, pp. 58–64. Springer, Heidelberg (2004)
15. Brownlee, J.: Clonal Selection Theory and CLONALG: The Clonal Selection Classification Algorithm (CSCA). Technical Report 2-02, CISCP, Swinburne University of Technology (2005)
16. Khaled, A., Abdul-Kader, H.M., Ismail, N.A.: Artificial Immune Clonal Selection Algorithm: A Comparative Study of CLONALG, opt-IA and BCA with Numerical Optimization Problems. International Journal of Computer Science and Network Security **10**(4), 24–30 (2010)
17. Ahmad, W., Narayanan, A.: Humoral artificial immune system (HAIS) for supervised learning. In: Proceedings of IEEE World Congress on Nature and Biologically Inspired Computing, NaBIC2010, pp. 37–44 (2010)
18. Woldemariam, K.M., Yen, G.G.: Vaccine-Enhanced Artificial Immune System for Multimodal Function Optimization. IEEE Transactions on Systems, Man and Cybernetics **40**(1), 218–228 (2010)
19. Castro, P.A.D., Von Zuben, F.J.: MOBAIS: a bayesian artificial immune system for multiobjective optimization. In: Bentley, P.J., Lee, D., Jung, S. (eds.) ICARIS 2008. LNCS, vol. 5132, pp. 48–59. Springer, Heidelberg (2008)
20. Castro, L.N.D., Timmis, J.: An artificial immune network for multimodal function optimisation. In: Proceedings of IEEE World Congress on Evolutionary Computation, pp. 669–674 (2002)
21. Kelsey, J., Timmis, J.: Immune inspired somatic contiguous hypermutation for function optimisation. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 207–218. Springer, Heidelberg (2003)
22. WEKA, Machine Learning Group at the University of Waikato. Weka 3: Data Mining Software in Java
23. Zhang, G.: Neural networks for classification: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C **30**(4), 451–462 (2000)
24. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial Neural Networks: A Tutorial. IEEE Computer, 31–44 (1996)
25. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
26. Quinlin, J.R.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)
27. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/