



A Neural Network Classification Model Based on Covering Algorithm and Immune Clustering Algorithm

F. Zhao

Fang Zhao

School of Information Science and Technology,
Zhejiang Shuren University
Hangzhou 310015, China
fly_Sylvia@163.com

Abstract

Inspired by the information processing mechanism of the human brain, the artificial neural network (ANN) is a classic data mining method and a powerful soft computing technique. The ANN provides a valuable tool for information processing and pattern recognition, thanks to its advantages in distributed storage, parallel processing, fast problem-solving and adaptive learning. The constructive neural network (CNN) is a popular emerging neural network model suitable for processing largescale data. In this paper, a novel neural network classification model was established based on the covering algorithm (CA) and the immune clustering algorithm (ICA). The CA is highly comprehensible, and enjoys fast computing speed, and high recognition rate. However, the learning effect of this algorithm is rather poor, because the training set is randomly selected from the original data, and the number of nodes (covering number) and area being covered are greatly affected by the learning sequence. To solve the problem, the ICA was introduced to preprocess the data samples, and calculate the cluster centers based on the antibody-antigen affinity. The CA and the ICA work together to determine the covering center and radius automatically, and convert them into the weights and thresholds of the hidden layer of neural network. The number of hidden layer neurons equals the number of covering. In addition, the McCulloch-Pitts (M-P) neurons were adopted for the output layer. Based on the input feature of the hidden layer, the output feature completes the mapping from input to output, creating the final classes of the original data. The introduction of the ICA fully solves the defect of the CA. Finally, our neural network classification model was verified through experiments on real-world datasets.

Keywords: Artificial neural network (ANN), covering algorithm (CA), immune clustering algorithm (ICA), constructive neural network (CNN)

1 Introduction

The artificial neural network (ANN) is an intelligent computer system mimicking the biological neural network of animal brains. Being a soft computing technique, the ANN enjoys various advantages, ranging from distributed storage, parallel processing, fast problem-solving and adaptive learning. As a result, the ANN has been widely applied in many fields, especially in information processing and pattern recognition [1, 2, 3, 4, 11, 15, 17, 21].

Proposed by Zhang Ling and Zhang Bo, the constructive neural network (CNN) is grounded on the geometrical meaning of McCulloch-Pitts (M-P) neurons. The main ideal of the CNN is to transform the neural network design into a problem of domain coverage [24]. Based on feedforward CNN, the covering algorithm (CA) is highly comprehensible and easy to construct, without no false covering center and a few hidden network nodes. In addition, the algorithm also boasts a fast computing speed, a high recognition rate, and a quick identification of the covering center. However, the learning effect of this algorithm is rather poor. This is because the training set is randomly selected from the original data, and the number of nodes (covering number) and area being covered are greatly affected by the learning sequence.

The immune clustering algorithm (ICA) is an excellent way to process non-convex datasets, without needing the number of clusters [13]. The algorithm aims to build a memory set that can identify and represent the structure of the target dataset. When it is applied to 2D and 3D datasets, the ICA first obtains the memory network, then visualize the distribution structure of the dataset, and finally output desirable clustering results [5, 8].

In the light of the above, this paper designs a novel neural network classification model. The ICA was taken as the frontend processor of the neural network to cluster the data samples. Besides, the ICA was combined with the CA, eliminating the negative impact of learning sequence on efficiency. In this model, the center and radius of covering are determined automatically, and converted into the weights and thresholds of the hidden layer of the neural network. The number of hidden layer neurons equals the covering number, while the M-P neurons were selected for the output layer. The input-output mapping was completed by evaluating the input feature of the hidden layer. The effectiveness of our model was proved through experiments.

2 Covering algorithm and immune clustering algorithm

2.1 Covering algorithm

(1) Constructive neural networks (CNN)

The CNN was developed by Zhang Ling and Zhang Bo, drawing on the geometrical meaning of M-P neurons. The network mainly projects an n -th dimensional vector onto an $(n+1)$ -th dimensional hypersphere, turning the design of a neural network into a problem of domain covering [24]. The mapping process can be explained as follows: First, the bounded set D is transformed in n -th dimensional space and mapped to S_n , $T : D \rightarrow S_n$, where S_n is a hypersphere in the S_n -th dimensional space; through the transform, D is mapped one by one to a hypersphere with a radius of d , facilitating the data analysis and processing on the hypersphere.

For a given sample set, setting up a network is equivalent to covering a given sample set with a set of domains. In this way, the neural network construction is converted into a domain covering process. With clear geometrical meaning, the CNN can process large data problems intuitively with a small amount of calculation. Hence, this technique has attracted much attention in the academia [7, 9, 18, 19, 23].

(2) Covering algorithm

An M-P neuron is a processing unit that contains multiple inputs and a single output. The input-output relationship of M-P neurons can be expressed as a symbolic function:

$$y = \text{sgn}(\langle \omega, x \rangle - \theta) \quad (1)$$

where $x = (x_1, x_2, \dots, x_n)^T$ is an n -th dimensional input vector; $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$ is a weight vector; θ is a threshold. The classification boundary of M-P neurons is a hyperplane in n -th dimensional space:

$$\langle \omega, x \rangle - \theta > 0 \quad (2)$$

where ω is a unit vector and the normal vector of the hyperplane; $|\theta|$ is the distance between the origin of the coordinates and the hyperplane. In n -th dimensional space, Eq. (1) consists of two functions with $\langle \omega, x \rangle - \theta > 0$ represents the hyperplane P . If $\langle \omega, x \rangle - \theta > 0$, then the corresponding point

falls in the positive half space of the hyperplane and $sgn(\langle \omega, x \rangle - \theta) = 1$; if $\langle \omega, x \rangle - \theta < 0$, then the corresponding point falls in the negative half space of the hyperplane and $sgn(\langle \omega, x \rangle - \theta) = -1$.

The M-P neurons can be seen as recognizers that can identify the spatial locations partitioned by hyperplanes. However, the problem will become too complex to be represented intuitively by the hyperplane of the neurons, when there are too many dimensions and hyperplanes.

To solve the problem, some improvements were made by Zhang and Zhang [23]. It is assumed that the input vectors are of equal length, and limited to a sphere in the n -th dimensional space. Then, $\langle \omega, x \rangle - \theta > 0$ represents the part of the sphere P in the positive half of the space. This part is called a spherical field (Fig.1) on the sphere. If ω and x are equal, then ω is the center of this spherical field, and $r(\theta)$ is the radius. The radius decreases monotonically with the increase of θ . Taking Eq. (2) as the activation function $sgn(\langle \omega, x \rangle - \theta)$, the activation function is the characteristic function of the spherical domain on the corresponding sphere. In this way, the complex problem can be solved easily in the sphere:

$$sgn(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{3}$$

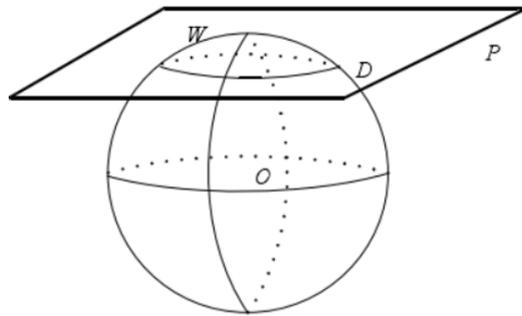


Figure 1: The spherical field

Through the above transform, the neural network architecture can be constructed by finding the optimal covering. Let the definition domain H of the input be the bounded set D in the n -th dimensional space, and S^n be the n -th dimensional hypersphere in the $n + 1$ -th dimensional space. If the inputs are unequal, the sample points can be mapped to the spherical surface S .

$$T : D \rightarrow S^n, T(x) = (x, \sqrt{r^2 - \|x\|^2}) \tag{4}$$

Eq. (4) can be understood as follows: Located inside S , D is an n -th dimensional hyperplane passing through the origin in the $(n + 1)$ -th dimensional space; then, each point on D is projected perpendicularly to S through transform T . The original points have a one-to-one correspondence with the projections on the hemispherical surface (Fig.2). To sum up, each neuron (ω, θ) is a characteristic function of the spherical domain with ω as the center on the hypersphere S^n and the radius of $r(\theta) = r(\cos(\theta/R))$.

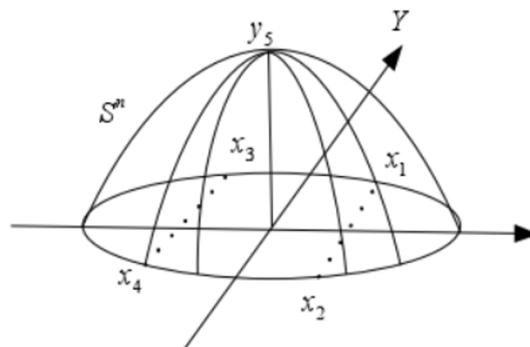


Figure 2: The transform $D \rightarrow S^n$

If the samples to be classified are unknown, then each sample is judged whether it belongs to a certain area, according to the class of the sample and the covering CNN, i.e. whether the domain $\langle \omega, x \rangle - \theta > 0$ exists. It is necessary to check if any field contains an unknown sample in the spherical sphere. Once such a field is determined, the unknown sample can be categorized correctly.

It is assumed that the learning sample has a total of N classes, $X = \{X_1, X_2, \dots, X_N\}$. The basic idea of the CA can be described as: setting up a spherical sphere for each sample point in each class, such that the final constructed field covers all the initial sample points. For the k -th learning sample X_k , the spherical domain can be learned and constructed by selecting the point a_i not covered in X_k , and computing $d(\omega)$ by:

$$\begin{aligned} d^1(\omega) &= \max_{x \notin X_k} \{ \langle a_i, x \rangle \} \\ d^2(\omega) &= \min_{x \in X_k} \{ \langle a_i, x \rangle \mid \langle a_i, x \rangle > d^1(\omega) \} \\ d(\omega) &= \frac{1}{2}(d^1(\omega) + d^2(\omega)) \end{aligned} \tag{5}$$

In addition, a covering $C(a_i)$, $\langle \omega, x \rangle - \theta > 0$ should be built with a_i as the center and $\theta = d(\omega)$ as the threshold.

Algorithm 1 CA for sample set X

Step 1: Initialize the class number $i = 1$, find the maximum modulus r of all samples in sample set X , and then project the point in X to the corresponding sphere. The center of the sphere is at the origin and the radius is $R(R > r)$.

Step 2: Construct a covering $C(i)$. If no point in X_i is covered, go to Step 7; otherwise, take any point a_i in X_i has not been covered.

Step 3: Perform calculation by Eq. (5), and make the covering $C(a_i)$ with a_i as the center and θ as the threshold.

Step 4: Find the center of gravity of all points covered by $C(a_i)$, map it to the sphere, and denote the projection as a'_i . Continue to calculate the threshold θ' by Eq. (5), and find the spherical sphere $C(a'_i)$.

Step 5: If $C(a'_i)$ is greater than the number of points covered by $C(a_i)$, such that $a'_i \rightarrow a_i$ and $\theta' \rightarrow \theta$, go back to Step 4; otherwise, go to Step 6.

Step 6: Calculate the translation point a''_i of a_i , and set up $C(a''_i)$. If $C(a''_i)$ is greater than the number of points covered by $C(a_i)$, go to Step 4; Otherwise, obtain a coverage $C(i)$.

Step 7: If class label i does not reach the maximum number of classes, then go back to Step 3; otherwise, terminate the algorithm.

2.2 Immune clustering algorithm

Immune algorithms are designed based on the immune process of the natural immune system [14]. Among them, the ICA designed by De Castro, named artificial immune network algorithm (aiNet) [13], has been widely used for data compression and clustering [6, 10, 12, 22]. The pseudo code of this ICA is presented below.

In the above algorithm, the differentiation between classes in affinity relies on affinity measurement. Let $x_1 = (x_{11}, x_{12}, \dots, x_{1m})$ and $x_2 = (x_{21}, x_{22}, \dots, x_{2m})$ be two sample points. Then, the affinity between them can be measured by the Euclidian distance $d_{1,2}$:

$$d_{1,2} = \sqrt{\sum_{j=1}^m (x_{1j} - x_{2j})^2} \tag{6}$$

So far, the basic ideas of the CA and ICA have been introduced, paving the way for model construction.

Algorithm 2 The aiNet

-
- 1: Initialization: randomly initialize the antibody Ab ;
 - 2: Set the main parameters: σ_d (death threshold), σ_s (suppression threshold), N (clone multiplier), ξ (affinity maturation rate), n (number of clones);
 - 3: **while** the termination condition is not satisfied **do**
 - 4: **for** an antigen $x_i \in Ag$ **do**
 - 5: Affinity calculation: Calculate the affinity between x_i and each Ab in the antibody network;
 - 6: Cloning: Reorder the Ab based on the affinity value, and select the n antibodies with the highest affinity to x_i for cloning. Note that the number of cloned antibodies is positively correlated with the affinity;
 - 7: Mutation: Perform mutation on the set of clones by $C = C - a * (C - x_i)$;
 - 8: Selection: Calculate the affinity between population c and x_i after mutation, and add the antibody with the highest affinity $\xi\%$ into memory cell M ;
 - 9: Cloning inhibition: Remove the antibody node with the antibody-antibody affinity less than σ_s in M ; remove the antibody node with the antibody-antigen affinity less than threshold σ_d in M ;
 - 10: **end for**
 - 11: Consolidate antibodies and memory cells: $Ab = [Ab; M]$;
 - 12: Network suppression: Remove antibody nodes whose affinity between Ab antibodies is less than the threshold σ_s ;
 - 13: Optimization: Randomly generate some new antibodies to replace the $r\%$ antibodies with the lowest affinity in the antibody network;
 - 14: **end while**
-

3 Model construction

3.1 ICA-based determination of covering center and radius

After the run of the aiNet algorithm, a simplified structure can be obtained to represent the original dataset, namely, the memory cell population [20] (Fig.3 (a)). On this basis, the minimal spanning tree (MST) of the antibody was obtained by Prim's algorithm (Fig.3 (b)). Then, the side length of the MST was depicted in a bar chart (Fig.3 (c)). For each edge, if the sum of its standard deviation and mean connection strength is smaller than the connection strength in the current iteration, then the edge should be considered an inconsistent edge. This ensures that the edges connected together can represent the same class [1, 16]. The process looks like this:

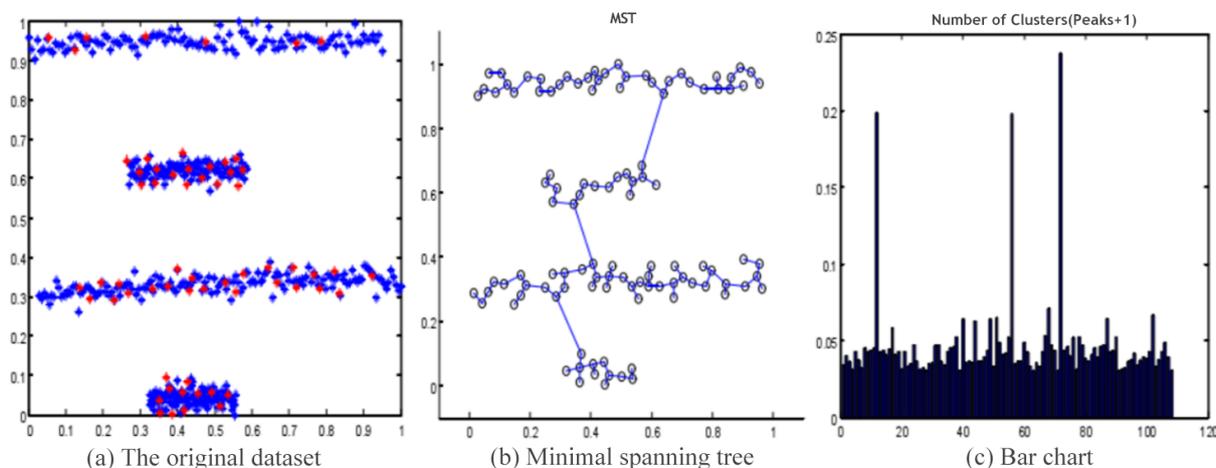


Figure 3: The MST and bar chart of the dataset to be clustered

As shown in Fig.3, three sides were much longer than the other sides. Four cluster classes and their center points can be obtained by cutting from the three sides. Next, the number, center and

radius of covering can be derived from the clustering results, and a covering set $C(a_1, a_2, \dots, a_p)$ can be obtained through the ICA, where p is equal to the cluster class.

3.2 Neural network model

Let $E = \{r^t = (X^t, Y^t), t = 0, 1, \dots, p\}$ be a sample set. Then, the memory of the neural network can be described as follows: The neural network remembers the memory relationship between input X^i and output Y^i . The association of the neural network can be expressed as: If the input of the neural network is $X^i + \Delta^i$, where Δ^i is interference or noise, the output of the neural network will remain as Y^i . Using a neuron to represent the space division of the sphere, the memory-association or classification function of the traditional neural network can be achieved through the space division.

Thus, neural network classification model can be defined by the following elements:

(1) Feature vector of sample point $A(a_1, a_2, \dots, a_n)$. This vector carries the features of the neuron. Each neuron has a unique feature vector, and determines if the class output should be activated based on this feature vector.

(2) Radius r . The radius defines the scope of influence of a neuron.

(3) Neuron class or function. This element defines the function of each neuron in the neural network, and marks the class of the sample. The output of the neuron is either activated or disabled.

(4) Input X . The input defines the input vector (x_1, x_2, \dots, x_n) of the data sample. If the input is within the scope of influence of a neuron, then the neuron output will be activated in the current iteration; otherwise, the neuron output will be disabled.

(5) Activation function:

$$y = \text{sgn}\left(\sum_{i=1}^n \|x_i - a_i\|^2 - r^2\right) \quad (7)$$

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (8)$$

So far, the authors have defined the neural network classification model, and introduced its parameters.

4 Model architecture and its algorithm

Unlike the backpropagation (BP) model, a neural network does not need weight adjustment in its construction. Instead, the structure is integrated into the learning process of the neural network. In this way, the established network can tolerate a high fault and recognize each sample.

4.1 Model architecture

The architecture of classification neural network model is shown in Fig.4.

As shown in Figure 4, the neural network consists of three layers, namely, the input layer, the classification layer and the M-P layer. The input layer mainly defines the input vector (x_1, x_2, \dots, x_n) of the data sample. The classification layer, containing M neurons, divides the sample space into M subspaces. The scope of influence of the M neurons equal the size of the covering, which covers all samples. Each neuron is mapped to the output layer, corresponding to a class. The M-P layer maps the neurons to the output layer based on their relevance to the input features, and outputs the classes of the neurons. The total number of classes of the original sample data is denoted as K .

4.2 Algorithm design

By the algorithm of our model, the test sample data inputted into the model are clustered and then converted into the hidden layer. Finally, the classification results of the inputs were outputted.

(1) Determination of classification neurons

(2) Determination of M-P neurons

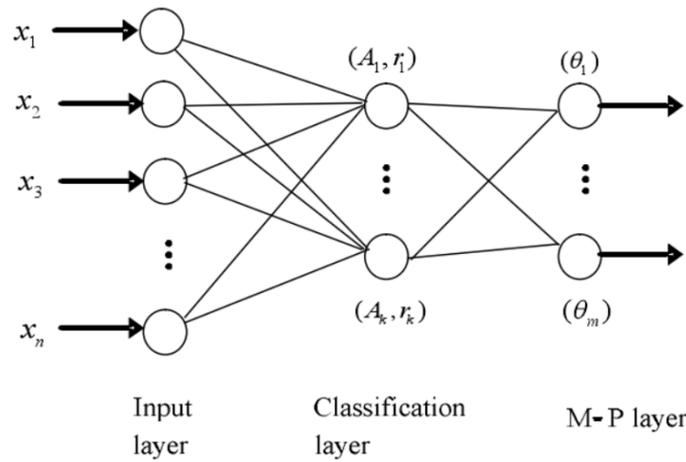


Figure 4: The architecture of neural network

Algorithm 3 Constructing classification neurons

Require: Sample set $R = \{(x^1, y^1), (x^2, y^2), \dots, (x^p, y^p)\}$

Ensure: Classification neurons

Step 1: Initialize the number of classification layer neurons $M = 0$.

Step 2: Process the sample set by the CA and the ICA to obtain the covering set: $C(a_1, a_2, \dots, a_p)$.

Step 3: For each covering in C , generate new classification neurons.

$$M = M + 1, A^M = x^k$$

The weights of M-P neurons can be determined based on the components of neuron clustering samples:

$$W_{mk} = -(y(A^m))_k, m = 1, 2, \dots, M, k = 1, 2, \dots, K \tag{9}$$

$$\theta_k = -\sum_m W_{mk} \tag{10}$$

This section fully explains the construction and algorithm of the neural network classification model.

5 Experiments and results analysis

To verify its effectiveness, the proposed neural network classification model was applied to two experiments on the data from three UCI standard databases, in comparison to the k-means clustering (KMC) and the CA. The classes, number of samples and number of conditional attributes of Iris, Wine, and Zoo databases are listed in Table 1 below.

Table 1: Parameters of the three datasets

Dataset	Number of classes	Number of samples	Number of conditional attributes
Iris	3	150	4
Wine	3	178	13
Zoo	7	214	15

The original data were divided into a training set and a testing set. The former was used to train the classification neural network, and the latter to test the accuracy of this network. The ICA was employed to determine the number and center of clusters, calculate the radius of the clusters, and convert them to the network parameters. Table 2 lists the classification accuracies of our model,

Table 2: The classification accuracy rate of our method

Method	Classification accuracy (%)		
	Our method	k-means	Covering
Iris	95.80	88.67	5.58
Wine	92.26	86.52	91.49
Zoo	88.14	84.31	89.87

the KMC and the CA. The accuracy is the mean of 50 tests. Overall, our model achieved better classification accuracy than the two contrastive methods.

The superiority of our model can be explained as follows: Our model consists of both classification layer neurons and M-P neurons. From the perspective of covering, the results of immune clustering are adopted to partition the sample space, giving a clear meaning to the hidden layer neurons and making it easy to extract the rule knowledge. The whole sample space is completely divided by all hidden layer neurons. The number of neurons equals the number of immune cluster centers. Each classification neuron is described by two parameters (A, r) and its class, where A and r are the feature vector and radius of hidden layer neurons, respectively. Furthermore, the M-P layer serves as the output layer, and the hidden layer neurons are mapped to the output layer according to the relationship between them and the input features. In this way, our model outputs the classes of the original data.

6 Conclusion

To improve the classification results of neural networks, this paper puts forward a novel neural network classification model and its algorithm based on the CA and the ICA. The data samples were preprocessed by the ICA to determine the covering center according to the antibody-antigen affinity. Then, the CA was introduced to automatically determine the center and radius of the covering. Then, these parameters were converted into the weights and thresholds of the hidden layer of the neural network. The number of hidden layer neurons equals the number of clusters. The introduction of the ICA solves the problem of the CA in the random selection of initial covering center. Experimental results show that the proposed neural network classification model outperformed classic classification methods. Further research will integrate the CA and individual neural networks, aiming to further enhance the learning and recognition accuracy of our model in the context of big data.

References

- [1] Al-Enezi, J. R., Abbod, M. F., Alsharhan, S. (2010); Artificial immune systems-models, algorithms and applications, *International Journal of Research & Reviews in Applied Sciences*, 3(2), 118-131, 2010.
- [2] Chen, G.C.; Yu, J.S. (2005); Particle swarm optimization neural network and its application in soft-sensing modeling, *Advances in Natural Computation*, 3611, 610-617, 2005.
- [3] Choubey, H.; Pandey, A. (2018); Classification of healthy, inter-ictal and seizure signal using various classification techniques, *Traitement du Signal*, 35(1), 75-84, 2018.
- [4] Gao, H.; Gao, L.; Zhou, C.; Yu, D. (2004); Particle swarm optimization based algorithm for neural network learning, *Chinese Journal of Electronics*, 32(9), 1572-1574, 2004.
- [5] Hruschka, E.R.; Campello, R.J.G.B.; Freitas, A.A. (2009); A survey of evolutionary algorithms for clustering, *IEEE Transactions on Systems Man & Cybernetics Part C*, 39(2), 133-155, 2009.
- [6] Huang, Z. (1998); Extension to the k-means algorithm for clustering large data sets with categorical values, *Data Mining and Knowledge Discovery*, 2(3), 283-304, 1998.

- [7] Kaufman, L.; Rousseeuw, P.J. (1990); *Finding Group in Data: An Introduction to Cluster Analysis*, New York: John Wiley & Sons, 1990.
- [8] Lai, J.Z.C.; Huang, T.J.; Liaw, Y.C. (2009); A fast k -means clustering algorithm using cluster center displacement, *Pattern Recognition*, 42(11), 2551-2556, 2009.
- [9] Li, H.; Ding, S.F. (2013); A novel neural network classification model based on covering and affinity propagation clustering algorithm, *Journal of Computational Information Systems*, 9(7), 2565-2573, 2013.
- [10] Malim, M.R.; Halim, F.A. (2013); Immunology and artificial immune systems, *International Journal on Artificial Intelligence Tools*, 21(06), 1250031-1-1250031-27, 2013.
- [11] Mostefa, T.; Tarak, B.; Hachemi, G. (2018); An automatic diagnosis method for an open switch fault in unified power quality conditioner based on artificial neural network, *Traitement du Signal*, 35(1), 7-21, 2018.
- [12] Ng, A.Y.; Jordan, M.I.; Weiss, Y. (2002); On spectral clustering: Analysis and an algorithm, *Proceedings of Advances in Neural Information Processing Systems*, 14, 849-856, 2002.
- [13] Nunes, L.; Jose, F.; Zuben, V. (2001); An artificial immune network for data analysis, In *Data Mining: A Heuristic Approach*, 2001.
- [14] Qian, X.; Wang, X. (2009); A New study of DSS based on neural network and data mining, *International Conference on E-Business and Information System Security*, 1-4, 2009.
- [15] Salajegheh, E.; Gholizadeh, S. (2005); Optimum design of structures by an improved genetic algorithm using neural networks, *Advances in Engineering Software*, 36(11-12), 757-767, 2005.
- [16] Strikwerda, C. (2008); The danger theory and its application to artificial immune systems, *University of Kent at Canterbury*, 114-148, 2008.
- [17] Tang, C.; Cao, X. (2001); The research development of evolutionary neural networks, *Systems Engineering and Electronics*, 23(10), 92-97, 2001.
- [18] Wang, L.W.; Tan, Y.; Zhang, L. (2005); Constructive fuzzy neural networks and its application, *Advance in Neural Network-ISNN2005*, 3497, 440-445, 2005.
- [19] Wang, L.W.; Wu, Y.H.; Tan, Y.; Zhang, L. (2006); A modified constructive Fuzzy Neural Networks for classification of large-scale and complicated data, *Advance in Neural Network-ISNN2006*, 3972, 14-19, 2006.
- [20] Watkins, A.; Timmis, J.; Boggess, L. (2004); Artificial immune recognition system (AIRS), an immune-inspired supervised learning algorithm, *Genetic Programming and Evolvable Machines*, 5(3), 291-317, 2004.
- [21] Yao, W.; Wang, Q.; Chen, Z.; Wang, J. (2004); The research overview of evolutionary neural network, *Computer Science*, 31(3), 125-129, 2004.
- [22] Ye, S.Z.; Zhang, B.; Wu, M.R.; Zheng, W.B. (2003); Fuzzy classifier based on the constructive covering approach in neural networks, *Journal of Software*, 14(3), 429-434, 2003.
- [23] Zhang, B.; Zhang, L. (1992); *Theory and Applications of Problem Solving*, North-Holland, Amsterdam, 1992.
- [24] Zhang, L.; Zhang, B. (1999); A geometrical representation of McCulloch-Pitts neural model and its applications, *IEEE Transactions on Neural Networks*, 10, 925-929, 1999.



Copyright ©2020 by the authors. Licensee Agora University, Oradea, Romania.

This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.

Journal's webpage: <http://univagora.ro/jour/index.php/ijccc/>



This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).

<https://publicationethics.org/members/international-journal-computers-communications-and-control>

Cite this paper as:

Zhao, F. (2020). A Neural Network Classification Model Based on Covering Algorithm and Immune Clustering Algorithm, *International Journal of Computers Communications & Control*, 15(1), 1008, 2020. <https://doi.org/10.15837/ijccc.2020.1.3739>