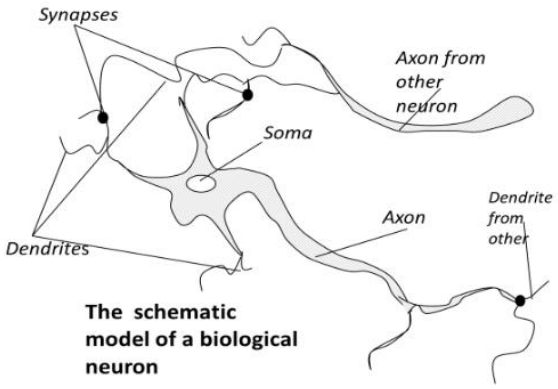
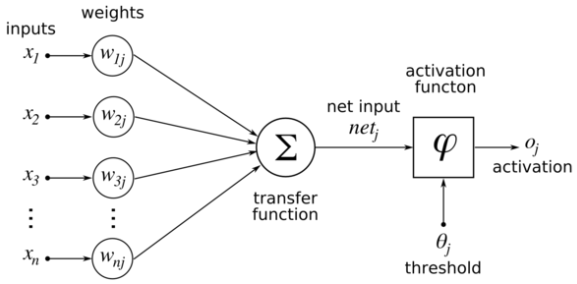
III. THE ANP

The first concept proposed in ANN is transforming the biological neuron into the artificial neuron by building the Perceptron, see figure 1(a).



(a)



(b)

FIGURE.1 (A) THE BIOLOGICAL NEURON (B) THE ARTIFICIAL NEURON

In figure 1 (b), a set of n synapses is associated to the inputs. Each of them is characterized by a weight. A signal at the i^{th} input is multiplied by its corresponding weight, and all the weighted input signals are summed. Thus, a linear combination of the input signals is obtained. A "free weight", or bias, is added to this linear combination and this forms a weighted sum, see equation 1. A nonlinear activation function φ is applied to the weighted sum, and eventually the value of the activation function is the neuron's output, see equation 2. The activation function could have many forms according to the neuron design (e.g. threshold function or a logistic function), see figure 2.

$$Z = \sum_{i=1}^{n} x_i w_i + x_0 w_0 \qquad \text{eqn. (1)}$$

$$y = \emptyset(Z) \qquad \text{eqn. (2)}$$

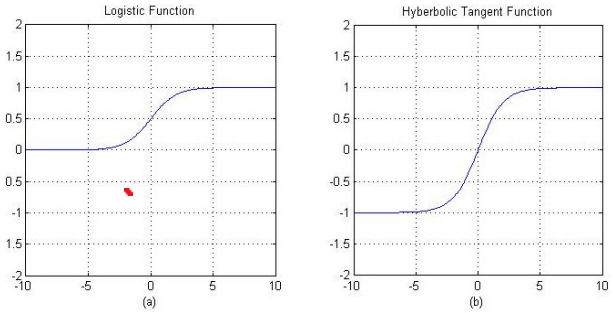


FIGURE.2 ACTIVATION FUNCTIONS:
(A) LOGISTIC FUNCTION (B) HYBERBOLIC TANGENT FUNCTION

During the training of the perceptron, the weight parameters are adjusted as the actual target is compared with the target output. If the actual output is similar to the target output, no change in weight is required. Otherwise, weight update is required according to the following equation, equation 3:

$$W_{k+1} = W_k + \lambda(o_i - t_i)x_{ij} \qquad \text{eqn. (3)}$$

Where λ is the learning rate, and W is the weight connecting neuron i and neuron j. In equation 3, it is evident that the required weight change is proportional to the error $(o_i - t_i)$. The learning process of the perceptron is governed by the following algorithm:

THE PERCEPTRON LEARNING ALGORITHM:

```
Initialize (weights);
For (t=0; t<Sizeof TrainingFile; t++)
{
Compute(Ynet);
ApplyThreshold(Ynet);
Compute(Yout);
If (Yout!=TargetValue)
{
UpdateWeights();
Continue;
}
Else
Break;
}
```

Having explained the basic concept behind the ANP, which is its basic building block, we can start analyzing the corresponding version intheartificial immune network. As the neuron is the basic building block of the nervous system, the B-Cell is the main building block of the Idiotopic immune network. In the following section, the concept of the AIP will be discussed and gradually built upon using concepts drawn from the biological immune system.

IV. THE PROPOSED AIP

According to Decastro and Timmis and Harmer et al, to apply an immunity based model in a specific domain, we should follow a series of problem solving stages to find a solution for a particular problem [13, 14].In this research design, attention is

focused on the Biological B-cell and its corresponding mathematical model that could be utilized to build the AIP, see figure 3.Within the B-Cell, two stages are represented: affinity measurement and affinity maturation, see figure 4. In affinity measurement the process of Ag-Ab matching is done, and matching pairs are produced: (P, Ep). As for the affinity maturation stage, only the matching pairs with high affinity (greater than a certain threshold) are going to be cloned and circulated in the artificial immune network.

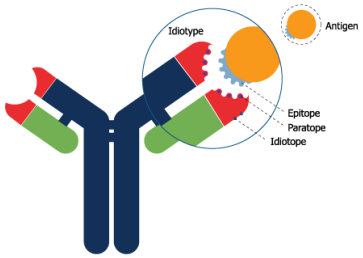


FIGURE.3 COMPLEMENTARITY BETWEEN THE BINDING REGION OF A RECEPTOR AND AN ANTIGEN EPITOPE [15]

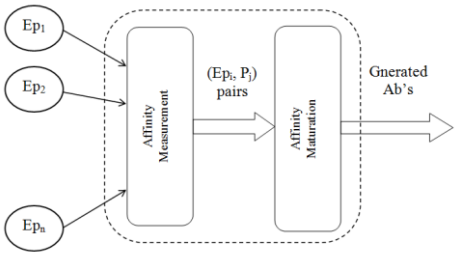


FIGURE.4 THE BASIC AIP

Starting with the affinity measurement stage, the immune system antigen recognition is based on the concept of complementarity between the binding region of the receptor P and the Ag epitope Ep. When complementarity is satisfied, then the B-Cell knows that this Ab locks to that Ag. This process is called *affinity*. There are many equations to measure the degree of affinity. Among which we will focus on the bit-wise affinity, where the string matching rule depends on the representation scheme and type of data. Therefore, that rule the hamming distance between two strings (P,Ep) could defined using equation 4:

$$h(P, Ep) = \sum_{i=1}^{N} P \oplus Ep \quad \text{eqn. (4)}$$

$h(P, Ep)$ is defined as the number of different bits between the two strings P and Ep. For exmple, if P=[1 0 0 1] and Ep=[0 1 1 0] then we expect $h(P, Ep)$ to be equale to 4, which is the length of the string P or Ep. That case is considered to have full complementarity where all the corresponding bits differ and as a result we have: $h(P, Ep) = N$, where N is the string length. However, as shown in equation 5, the calculated affinity $\mathcal{A}_{(\mathcal{P}_i,\mathcal{E}p)}$ in general must exceed θ (threshold) and not necessarily satisfy full complementarity, and a_i in that case measures the complementarity level.

$$\mathcal{A}_{(\mathcal{P}_i,\mathcal{E}p)} = \begin{cases} 0, & h(P, Ep) < \theta \\ a_i, & h(P, Ep) \geq \theta \end{cases} \quad \text{eqn. (5)}$$

Figure 5 illustrates the interaction that is taking place within the affinity measure stage starting from measuring the hamming distance between each P and Ep and ending up with deciding on the affinity levels.

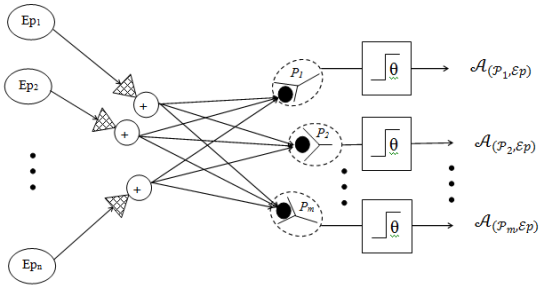


FIGURE.5 THE AFFINITY MEASUREMENT STAGE

All the aforementioned AIP design is based on an artificial immune model that considers the interaction between P and Ep, see figure 6. The affinity maturation stage, in general, is expected to produce clones on high or low affinity. In this research, the colnal selection algorithm is going to be used to control the learning process with the AIP, see figure 7.

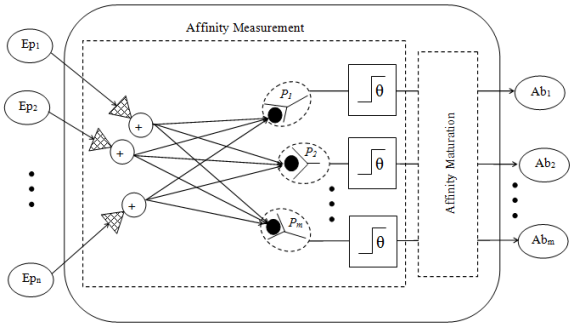


FIGURE.6 THE BASIC AIP INTERACTION

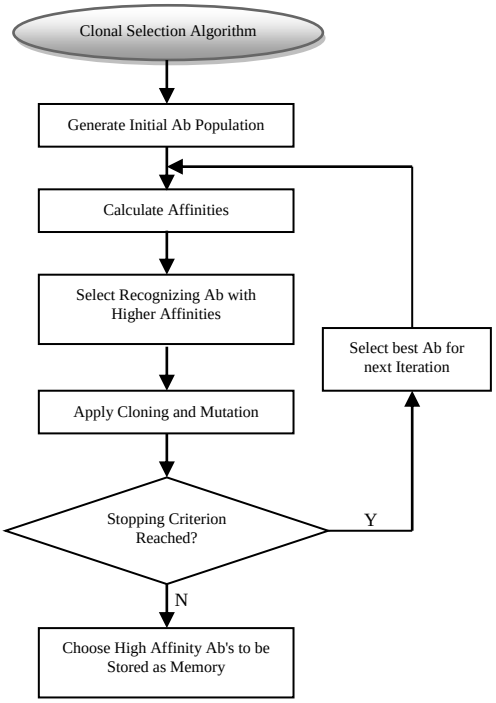


FIGURE.7 CLONAL BASED AIS SYSTEM

V. EXPERIMENTAL TESTS AND DISCUSSION OF RESULTS

Consider a simple ANP that has a single binary decision output unit. There are two input images that the perceptron need to recognize: a 3x4 image, S_1,representing number 1, and another 3x4 image, S_2, representing number 7, see figure 8. Pixels in images S_1 and S_2 are considered bipolar, which means it is either 1 or -1.

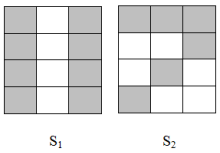


FIGURE.8 THE TWO INPUT PATTERNS

A. Character Recognition Using the ANP

Let us assume that there are a set of vectors in R^n where they could be classified into one of two classes S_1 and S_2.Each vector is in the format of $(x_1, x_2, ..., x_n)$, $n=12$. The output of the ANP is a linear combination of its weighted inputs, see figure 9, next page. The goal of the Perceptron algorithm is to find the values of those weights $w_1,w_2,...,w_n$that will enable the perceptron to differentiate between the two classes S_1 and S_2. The training of this proposed perceptron model is considered supervised. This means that the perceptron is expected to produce an output equals to 0 if class S_1 is recognized and produce an output equals to 1 if class S_2 is recognized. The ANP was trained for different epochs, and the recognition error values were recorded. Until epoch 50, the error was 0.9 and even after 500 iterations, where the error was almost 0.889. Figure 10, next page, shows only 50 iterations, because minor improvement was observed in the successive iterations.

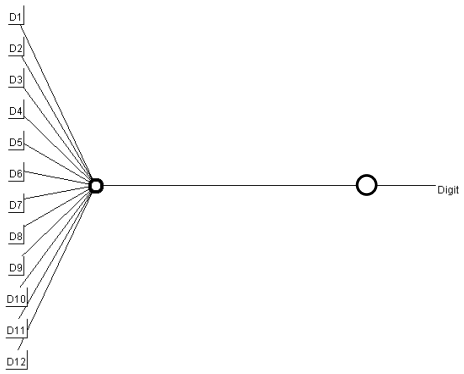


FIGURE.9 THE BASIC ANP MODEL

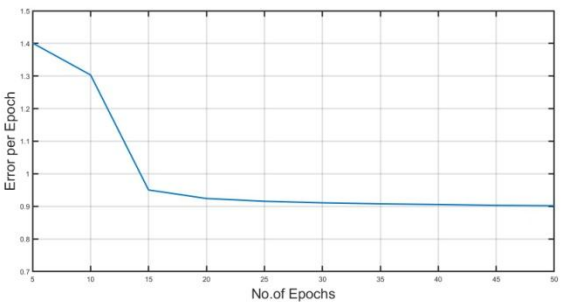


FIGURE.10 TRAINING OF THE BASIC ANP MODEL

B. Character Recognition Using the AIP

Based on the previous description of the AIS components we can describe their interaction. The clonal selection algorithm is one of the methods to model such interactions of the AIS with the external environments, or in other word, invading antigens.

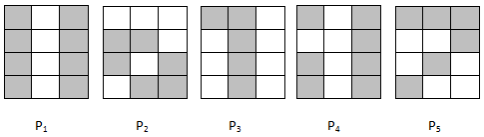


FIGURE.11 INITIAL POPULATION SAMPLE

Similar to the ANP algorithm, the clonal selection algorithm could be used to distinguish between two classes. In this simulation, One AIP unit is used to be compared to the basic ANP unit. In figure 11,$P_1, P_2, ..., P_5$*are the initial population samples, and they* are considered to be the antibodies, or antigen receptors, while S_1 and S_2 are considered antigens. It is assumed that the elements of S and P are represented in the same shape-space S^L. In the first iteration, after running the algorithm, we calculatedthe affinity for each antigen, M_1, with elements of P:

$$M_1 = \begin{bmatrix} 6 & 2 & 1 & 6 & 6 \\ 2 & 6 & 5 & 2 & 2 \end{bmatrix}$$

As mentioned earlier, the affinity values range between 0 and 12.According to the algorithm, the second step is to perform clonal selection and expansion. The highestn_1 affinity elements of P are chosen ($n_1=3$ in this experiment), and clones of these elements are generated in proportion to their affinity with the antigen. As the affinity increases, the number of copies increases. Next, Mutation occurs in proportion to the individual's affinity as well. As the affinity increases, the mutation rate decreases.

After Mutation, the mutated elements are added to the original population P. Then, the best individual is reselected to be kept as memory. As part of the meta-dynamics process, the lowest n_2 affinity elements of P are chosen ($n_2=2$ in this experiment). Then, they are replaced by a randomly generated new individuals. The whole process is repeated until a stopping criterion is met. In this experiment, the stopping criterion is met, when any affinity matching reaches 12. Figure 12 shows the affinity maturation for S_2, where a mutated version of P_2 at iteration 40 was able to recognize its pattern.

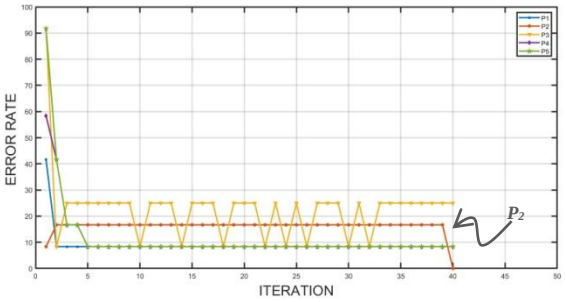


FIGURE.12TRAINING OF THE SINGLE AIP

C. Character Recognition Using the single-layer ANP

In the previous ANN experiment, the error showed no improvement less than 0.8, even after 500 iterations. It was imminent to increase the number of percrptrons per layer initially, in a single layer, to be able to judge on the network performance and eventually compare the findings with the AIP performance. a 7-1 single-layer perceptron was used to that end, see figure13.Not too much improvement has been observed compared to the previous single neuron experiment. After 50 iterations, a 0.80963 error was achieved and no noticeable progress took place even after 500 iteration; the error became 0.80492, see figure 14.

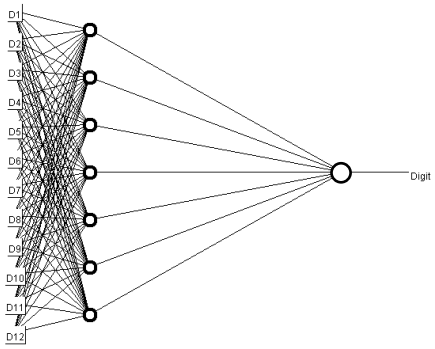


FIGURE.13 THE 7-1 SINGLE-LAYER ANPS

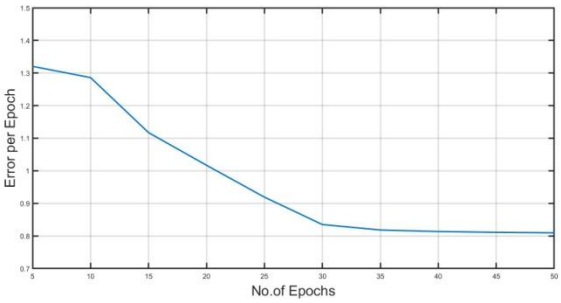


FIGURE.14 THE 7-1 TRAINING OF THE ANPS

D. Character Recognition Using the single-layer AIP

The previous experiment in section B was repeated using a single-layer of AIPs. Seven AIP's were used in a single-layer, and the simulation learning process was controlled by the clonal selection algorithm, see figure 15.

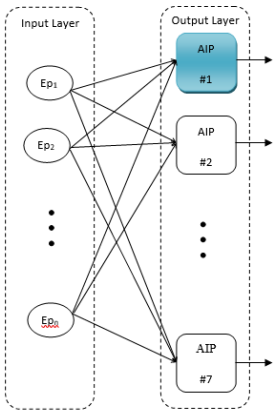


FIGURE 15. A SINGLE-LAYER AIP

In the depicted design, we have seven AIP's in one layer, which is considered the output layer. The AIP clonal selection algorithm is run for each AIP element in the layer, and the output of the winning element is chosen. The winning element is the one with the highest affinity. For each AIP we used a different input P vector(p_1, p_2, p_3, p_4, p_5).

After 14 iterations, for the S_1 pattern, P_2 vector was able to achieve full maturation before other recognizing vectors, which is an almost Zero Error recognition rate. Since the stopping criterion has been met, there was no need to further train the network using the other *P*vectors. At this point, the final mutated P_2 is the considered as the best recognizing vector (antibody). Then, eventually is reselected to be kept as memory, see figure 16.

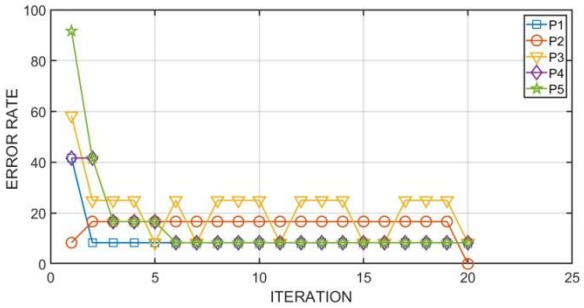


FIGURE.16TRAINING OF A SINGLE LAYER AIP

E. Character Recognition Using the multi-layer ANP

To be able to accurately judge the recognition ability of the AIP, it had to be compared with a full multi-layer artificial neural percptron model, and eventually compare the findings with the AIP performance. a 7-5-1 multi-layer ANP was used for that end, see figure 17. After training the network, a considerable improvement has been observed compared to the single-layer ANP. The recognition error reached 0.00982 at the 20th iteration, 0.000728 at the 200th iteration, and eventually 0.000191at the 500th iteration, which is almost Zero, see figure 18.

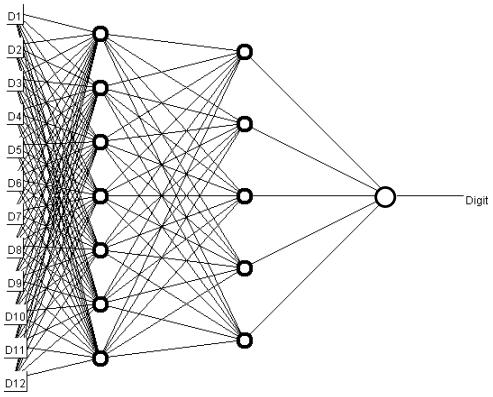


FIGURE.17 THE 7-5-1 MULTI-LAYER ANP

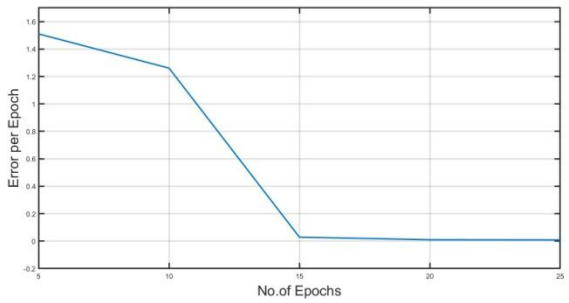


FIGURE.18 THE 7-5-1 ERROR PER EPOCH FOR THE SINGLE-LAYER ANP

F. Character Recognition Using the AIP

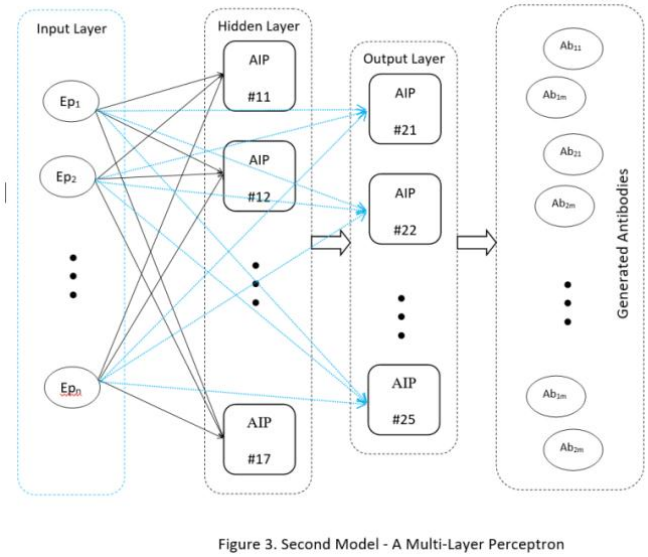


Figure 3. Second Model - A Multi-Layer Perceptron

FIGURE 19. A MULTI-LAYER AIP

experiment done at section D, stage-1, was continued to investigate the overall network performance by adding a second layer of 5 AIPs. as we finished stage-1, and the winner AIP was identified, see figure 19. That vector was less mutated and more cloned, while other AIP vectors, with less affinity were more mutated and then less cloned. From that pool, the inputs of the second artificial immune layer were chosen. It worth mentioning that that same S1and S2 from the original input layer are used as inputs to that second (output layer)as well. The clonal selection algorithm was run for each element in the output layer, and the output of the winning elementwas chosen. At this point, P2 is the considered as the best recognizing vector (antibody). Then, eventually is reselected to be kept as memory, see figure 20. Similar process was used to find the best recognizing vector for S2, and it was found that the mutated version of P3 can meet the condition for the stopping criterion after 9 iterations, see figure 21.

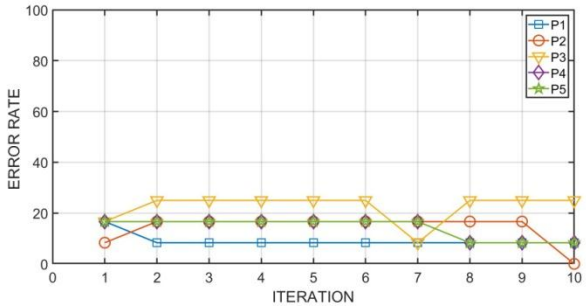


FIGURE.20 RECOGNITION OF S1 PATTERN-MULT LAYER AIP

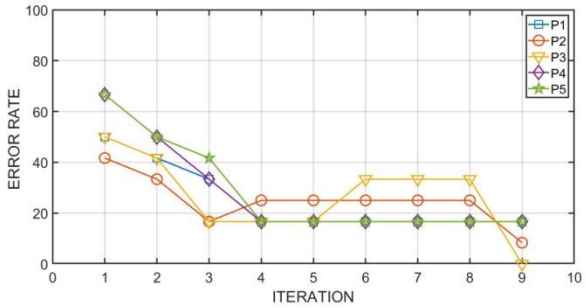


FIGURE.21 RECOGNITION OF S2 PATTERN-MULT LAYER AIP

VI. CONCLUSION AND FUTURE WORK

In this paper, simple AIP models are presented, and compared with the corresponding parallel models of the ANP models.Within the AIP models, the error rate decreased by 10% between the 1-1 and the 7-1 network structures. However, moving from the 7-1 to the 7-5-1 structure, the error rate noticeably improved from 0.8 to almost 0.01. That result is expected based on the 3-layer artificial neural network recognition capability. On the other hand, within the AIP model structures, the recognition performance increased by 67% between the 1-1 and the 7-1 network structures, while recognizing pattern S_1. As for pattern S_2, the recognition performance increased by 50% between the 1-1 and the 7-1 network structures. Moving to the multi-layer AIP, the recognition performance increased by 28.6% between the 7-1 and the 7-5-1 network structures for pattern S_1. As for pattern S_2, the recognition performance increased by 55% between the 7-1 and the 7-5-1 network structures.

The previous findings reflect that as the number of units in both networks increases, the recognition ability increases,and the error rate is reduced. However, it is imperative to judge on the performance of the AIP network by comparing it with the artificial neural network to evaluate the performance of the novel design. For the 1-1 network design, the AIP design outperformed the AIPsignificantly as it converged after 42 iterations for S_1 and 40 iterations for S_2comparedto an error rate of 0.9 even after 500 iterations. For the 7-1 design, as well,the artificial immune network model continued to outperform the artificial neural network model as it converges after 14 iterations for S_1 and 20 iterations for S_2compared to an error rate of 0.8 even at 500 iterations.As both networks have

more complex structures, 7-5-1, the performance gab was decreased, but the ANP network model still outperformed the ANP network model. The artificial immune network model converged after 10 iterations for S_1 and 9 iterations for S_2compared to an artificial neural network convergence after 20 iterations.Particularly, it was noticed that the AIP network-based model simulation examples show a better convergence and recognition abilities compared with the corresponding ANP network-based model examples.

As for the future research directions, the proposed artificial immune network could be improved bydesigning a reverse error calculation pass and eventuallycompare the overall AIP network-based model findings with a back propagation artificial neural network model. That research direction could open new avenues based on the new model. On the other hand, other artificial immune network training algorithms, such as the negative selection algorithm, could be also investigated to study its effect on the pattern recognition process, and perhaps eventually could be compared with the clonal selection algorithm.

REFERENCES

[1] Simon Hykins, 1999, Neural Networks: A Comprehensive Foundation, Prentice Hall.

[2] Dipankar Dasgupta, Luis F. Niño, 2009, Immunological Computation: Theory and application. NewYork, NY : CRC Press.

[3] Dipanker Dasgupta, 1998, “An Overview of Artificial Immune Systems and Their Applications,“ Artificial Immune Systems and their applications, by Dipankar Dasgupta, Springer.

[4] R. J. Kuo, Y. S. Tseng and Z. Chen, 2016, "Integration of fuzzy neural network and artificial immune system-based back-propagation neural network for sales forecasting using qualitative and quantitative data," Journal of Intelligent Manufacturing, vol. 27, (6), pp. 1191-1207.

[5] Huang Yue , Shenyang Ligong, Li Dan ; Gao Liqun, 2012, “ Power system short-term load forecasting based on neural network with artificial immune algorithm”, 24th Chinese Control and Decision Conference (CCDC), Taiyuan,China.

[6] Ruiying Zhou, Qiuhong Fan, Mingjun Wei, 2011, “Solving for multimodal function with high dimensions base on Hopfield Neural Network and immune algorithm”, International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), China.

[7] Waiel F. Abd El-Wahed, E. M. Zaki and A. M. El-Refaey, 2010, "Artificial immune system based neural networks for solving multi-objective programming problems," Egyptian Informatics Journal, vol. 11, (2), pp. 59-65.

[8] D. A. B. Fernandes et al, 2017, "Applications of artificial immune systems to computer security: A survey," Journal of Information Security and Applications, vol. 35, pp. 138-159.

[9] R. A. Zitar and A. Hamdan, 2013, "Genetic optimized artificial immune system in spam detection: a review and a model," *Artificial Intelligence Review,* vol. 40, *(3),* pp. 305-377.

[10] A. H. Mohammad and R. A. Zitar, 2011, "Application of genetic optimized artificial immune system and neural networks in spam detection," Applied Soft Computing Journal, vol. 11, (4), pp. 3827-3845.

[11] N. Bayar et al, 2015, "Fault detection, diagnosis and recovery using Artificial Immune Systems: A review," Engineering Applications of Artificial Intelligence, vol. 46, pp. 43-57.

[12] C. C. Kiang and R. Srinivasan, 2012, "An artificial immune system for adaptive fault detection, diagnosis and recovery," *International Journal of Advances in Engineering Sciences and Applied Mathematics,* vol. 4, *(1),* pp. 22-31.

[13] De Castro, L. N. and J. Timmis, 2002, Artifi cial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg.

[14] Harmer, P., G. Williams, P. D. Gnush and G. Lamont, 2002, An artificial immune system architecture for computer security applications. IEEE Transaction on Evolutionary Computation, 6(3), 252‑280.

[15] Anti-Idiotypic Antibody Production Service web page. Last access on Dec. 2018; URL https://rockland-inc.com/anti-idiotypic-antibody-production.aspx.