Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 1-8
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# Using Neural Networks to Predict Secondary Structure for Protein Folding

**Ali Abdulhafidh Ibrahim[1], Ibrahim Sabah Yasseen[2]**

[1]College of Science, Al-Nahrain University, Baghdad, Iraq
[2]College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
Email: Ibrahim.altaai@gmail.com

**How to cite this paper:** Ibrahim, A.A. and Yasseen, I.S. (2017) Using Neural Networks to Predict Secondary Structure for Protein Folding. *Journal of Computer and Communications*, **5**, 1-8.
http://dx.doi.org/10.4236/jcc.2017.51001

**Received:** November 7, 2016
**Accepted:** December 26, 2016
**Published:** December 29, 2016





## Abstract

Protein Secondary Structure Prediction (PSSP) is considered as one of the major challenging tasks in bioinformatics, so many solutions have been proposed to solve that problem via trying to achieve more accurate prediction results. The goal of this paper is to develop and implement an intelligent based system to predict secondary structure of a protein from its primary amino acid sequence by using five models of Neural Network (NN). These models are Feed Forward Neural Network (FNN), Learning Vector Quantization (LVQ), Probabilistic Neural Network (PNN), Convolutional Neural Network (CNN), and CNN Fine Tuning for PSSP. To evaluate our approaches two datasets have been used. The first one contains 114 protein samples, and the second one contains 1845 protein samples.

## Keywords

Protein Secondary Structure Prediction (PSSP), Neural Network (NN), $\alpha$-Helix (H), $\beta$-Sheet (E), Coil (C), Feed Forward Neural Network (FNN), Learning Vector Quantization (LVQ), Probabilistic Neural Network (PNN), Convolutional Neural Network (CNN)

## 1. Introduction

Bioinformatics involves the technology that uses computers for storage, retrieval, manipulation, and distribution of information related to biological macromolecules such as DNA, RNA, and proteins. The use of computers is absolutely essential in mining genomes for information gathering and knowledge building [1].

Protein structure prediction methods are categorized under bioinformatics which is a broad field that combines many other fields and disciplines like biology, biochemistry, information technology, statistics, and mathematics [2].

There are four different structure types of proteins, namely the Primary, Secondary, Tertiary and Quaternary structures. The Primary structure contains a sequence of 20 different types of amino acids. It provides the foundation of all the other types of structures. The Secondary structure refers to the arrangement of connections within the amino acid groups to form three different structured classes (H, E, and C) [3].

PSSP provides a significant first step toward the tertiary structure prediction, as well as offering information about protein activity, relationship, and function. Protein folding, or the prediction of the tertiary structure from linear sequence, is an unsolved and ubiquitous problem that invites research from many fields of study, including computer science, molecular biology, biochemistry and other. Protein secondary structure is also used in a variety of scientific areas, including proteome and gene annotation. Therefore, PSSP remains as an active area of research, and an integral part of protein analysis [4].

In this research, the authors have proposed five models of NN that has been used, including FNN, LVQ, PNN, CNN and CNN Fine tuning for PSSP. The main objective of this work is to gain an improvement of prediction accuracy ($Q_3$) so that the implementation results show that the proposed model (CNN Fine Tuning) performs better than the other models and looks promising for problems with characteristics similar to that problem (PSSP) by achieving prediction accuracy with $Q_3$ = 90.31%.

## 2. Related Materials

In this section, we will introduce dataset description, measures of prediction accuracy.

### 2.1. Data Set Description

The first dataset is obtained from matlab math work [5] and from thesis [3], it is contain 114 protein samples which are divided into training dataset with 75 protein samples and testing dataset with 44 protein sample. It is contains 28.3% $\alpha$-helix (H), 21.3% $\beta$-sheet (E) and 50.4% coil (C). The second dataset is formed of proteins from four different classes. We used this dataset from bioinformatics information Lab from University of Missouri, United States of America, it is contains 1854 protein. The first dataset are used for (FNN, LVQ and PNN) the second one is used for CNN and CNN fine tuning.

### 2.2. Measures of Prediction Accuracy

We have used one measuring method to evaluate the prediction accuracy of implemented models of NN. The three state accuracy ( $Q_3$ ) is defined as the percent of residues that have been predicted correctly:

$$Q_3 = \frac{N_H + N_E + N_C}{N_T} \quad (1)$$

where $N_H$, $N_E$, $N_C$ are the number of correctly predicted residues of type $H$,

$E$ and $C$, respectively and $N_T$ is the total number of residues in dataset. $Q_3$ concise as useful measure to compare different prediction methods [6].

## 3. Methodology

In our work, we have used five different structures of Neural Networks including (Feed Forward NN, Learning vector quantization NN, Probabilistic NN). We used a sliding window of size 17 for each structure of NN that which moves through the protein sequence and the output of the network is attained for the residue in the middle of the window; so as a result, the input layer includes 17 × 20 = 340 neurons and output layer contain 3 neurons for each NN structure. During training, it receives the input vectors along with the expected output vectors. When making predictions, it returns output vectors representing the likelihood of each residue being in (H, E or C). **Figure 1** illustrates a general structure of NN classifier (for only FNN,PNN and LVQ) that is receiving several input vectors and returning the predicted output vectors, comparing it with what could be the correct (expected output) classification.

### 3.1. Feed Forward Neural Network

The first structure of NN used are (Feed Forward NN), by using one input layer and two hidden layer with 10 neuron for each layer and one output layer as shown in **Figure 2** that illustrates implemented FNN structure using Matlab Version (R2015a).

In FNN the processing units in each hidden layer are fully connected to units in previous layer but not connected to units in the same layer. Only the outputs of the unit are connected to the units of next layer. Therefore there is no feedback in the system [7].

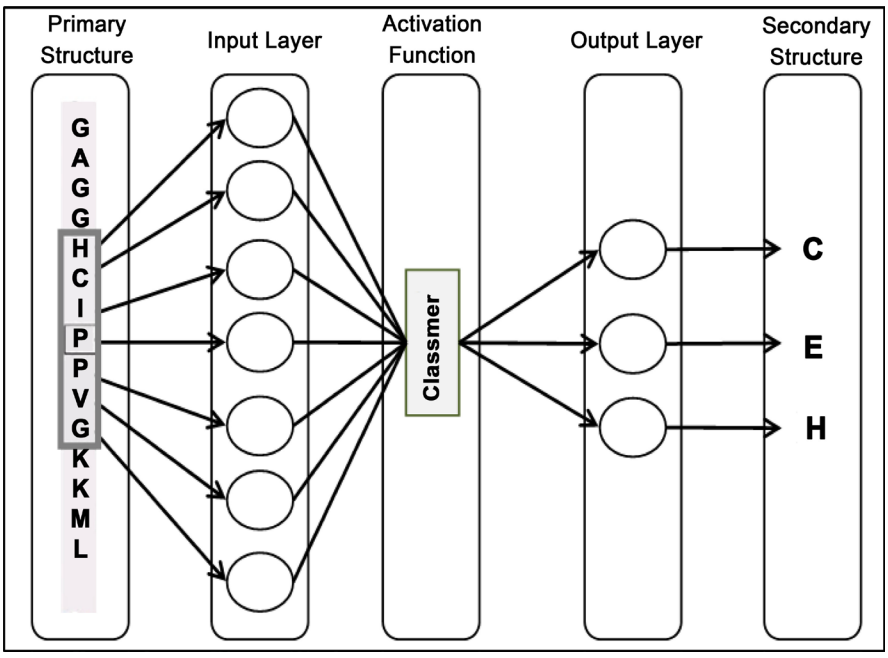


**Figure 1.** Classifier (NN) general structure.

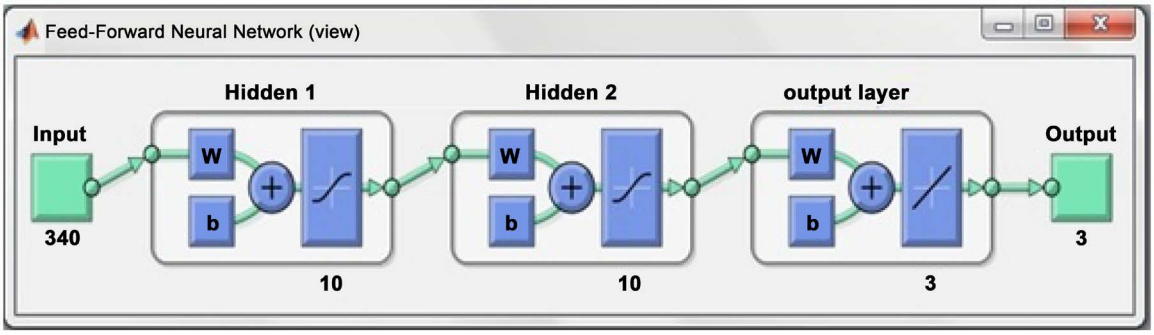


Figure 2. Feed forward neural network architecture.

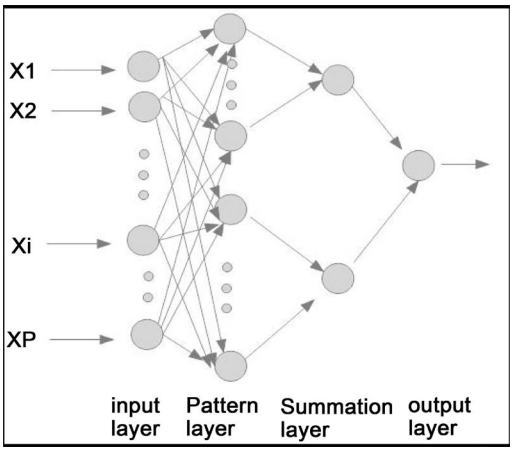


Figure 3. Architecture of PNN [10].

### 3.2. Probabilistic Neural Networks (PNN)

The second structures of NN used is (Probabilistic NN), PNN is defined as an implementation of statistical algorithm called Kernel discriminate analysis in which the operations are organized into multilayered feed forward network with four layers: input layer, pattern layer, summation layer and output layer [8], as shown in Figure 3.

It is usually much faster to train a PNN network than multilayer Perceptron Network but one of the disadvantages of PNN models compared to other networks is that PNN models has a large number of neurons in hidden nodes (pattern layer) due to the fact that there is one neuron for each training line [9]. Our implemented structure for PNN including one input layer with 340 neurons, and pattern layer with 14,151 neurons (one neuron for each amino acid), and three neurons for both summation and output layers as shown in Figure 4 that illustrate implemented PNN structure using Matlab (R2015a).

### 3.3. Learning Vector Quantization (LVQ)

The Third implemented structure of NN is LVQ; its structure has two layers is a competitive layer and linear layer [11], as shown in Figure 5.

LVQ is a method for training competitive layers in a supervised manner. The competitive layer learns to classify input vectors in much the same way as the competitive layers of Self-Organizing Feature Maps. The linear layer transforms

the competitive layer's classes into target classifications defined by the user. The classes learned by the competitive layer are referred to as subclasses and the classes of the linear layer as target classes [10]. Our implemented structure for LVQ includes one input layer with 340 neurons, and competitive layer with 10 neurons, and three neurons for both linear and output layers, as shown in Figure 6 that illustrates implemented LVQ structure using Matlab (R2015a).

### 3.4. Convolutional Neural Network (CNN)

CNN is a multilayer perceptron designed specifically to recognize two dimen-

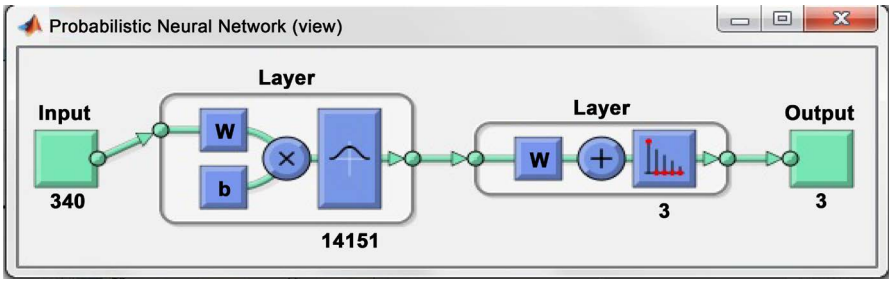


Figure 4. Implemented Architecture of PNN.

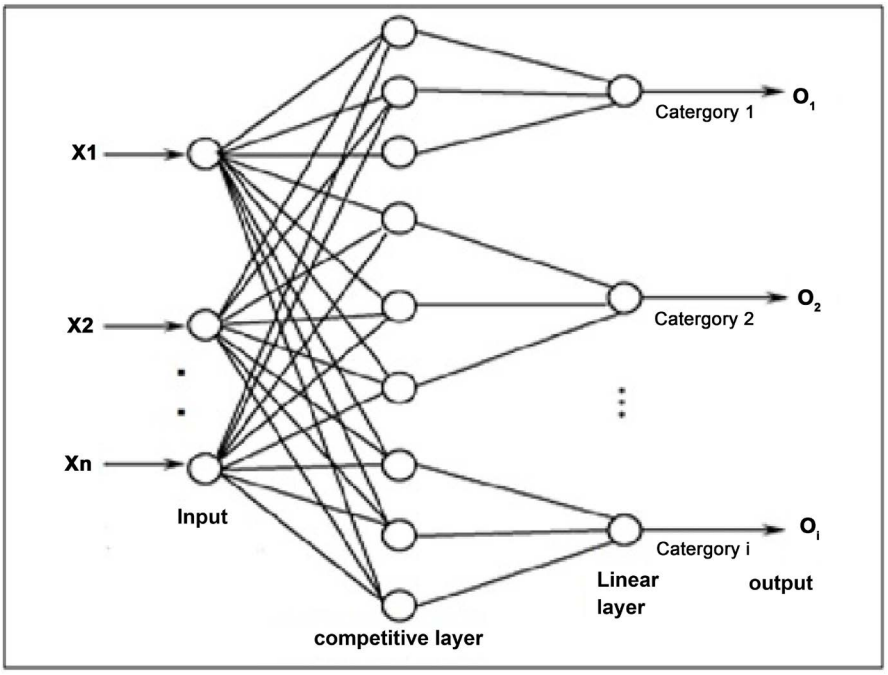


Figure 5. Architecture of LVQ [11].

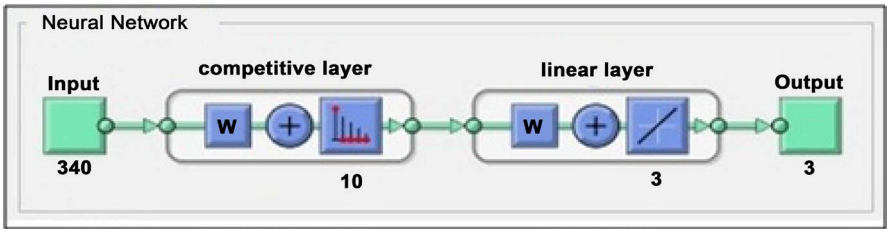


Figure 6. Implemented Architecture of LVQ.

sional shapes with a high degree of invariance to translation, scaling, skewing, and other forms of distortion. Figure 7 shows the architectural layout of convolutional network made up of an input layer, four hidden layers, and an output layer. This network is designed to perform image processing (e.g., recognition of handwritten characters) [12].

In our work we implement two different structures of CNN; the first implemented structure of CNN has six layers. The first layer is the input layer (input: 21 × 15), the second layer is the filter layer (Filter: 21 × 4 × 30), the third layer is the convolutional layer (Convolution: 30 × 12), the forth layer is the pooling layer (Pooling 30 × 6), and the fifth layer is the classifier layer which is (softmax classification). Finally, the last layer is the output layer. Figure 8 describes CNN classifier general structure that is receiving several input vectors and returning predicted output vectors.

The second implemented structure of CNN we used is a CNN fine tuning approach to tune the parameter of the whole model as step to increase the accuracy and find more accurate prediction for the secondary structure of the protein by replace the softmax activation function that is found in Figure 8 of the CNN first structure used by sigmoid activation function and do the back propagation approach for each epoch to tune the parameter.

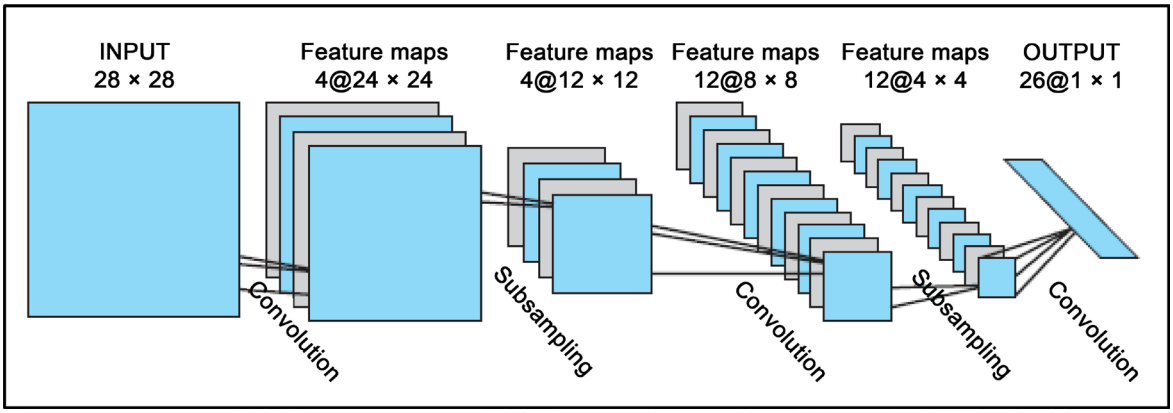


Figure 7. CNN for image processing such as handwriting recognition [12].

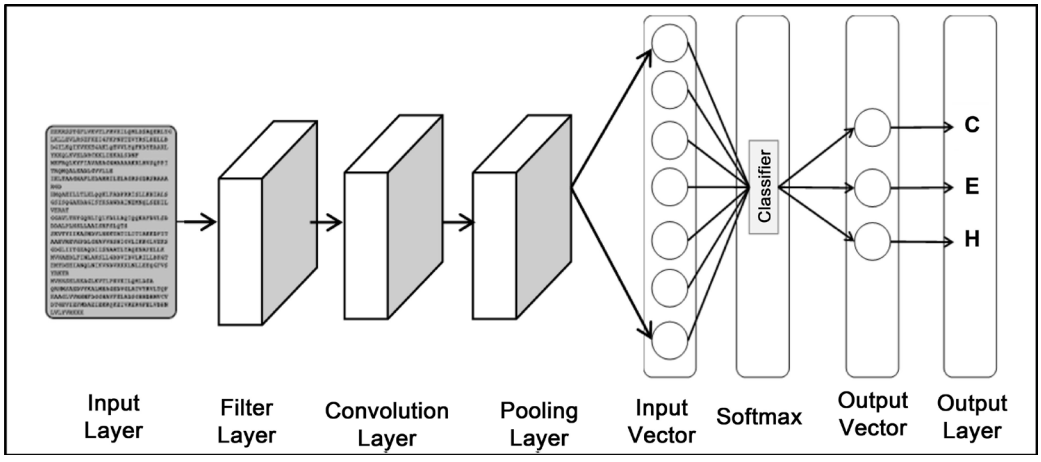


Figure 8. CNN classifier general structure [12].

## 4. Results and Discussion

In this section we will display the result of five implemented models of NN, in addition previously are mentioned $Q_3$ as three state accuracy measures will use $Q_H$, $Q_E$ and $Q_C$ are the percentage of correctly predicted residues observed in class E, H and C, respectively, as shown in Figure 9.

Figure 9 visualizes comparison between the five implemented models of NN. This figure compares the average of $Q_H$, $Q_E$, $Q_C$ and $Q_3$ of these three structures and shows that in Feed Forward NN give higher prediction accuracy $Q_3$ than PNN and LVQ and more balanced prediction of three secondary structures and less difference between prediction accuracy of H, E and C. In PNN structure there is high difference between prediction accuracy of (H, E) and C in other side. This is because of class imbalance problem in protein secondary structure datasets which causes that the classifiers give more importance to majority class (C). In LVQ show that this structure can be train and predict only coil (C) so it has higher prediction accuracy (100%) and completely predicted correctly and other classes (H, E) completely predicted wrongly (0%). Finally it has been proven that CNN Non Fine Tuning and CNN Fine Tuning can obtain higher prediction accuracy than all other implemented structures by achieving prediction accuracy (61.694%) for Non Fine Tuning and (90.31%) for Fine Tuning model.

## 5. Conclusion

In this work, we presented five structures of NN, including FNN, PNN, LVQ, CNN and CNN fine tuning for prediction of protein secondary structures. The results show that CNN fine tuning network can achieve better performance and can improve prediction accuracy when compared to other structures by achieving prediction accuracy with (90.31%). CNN and CNN Fine Tuning need a

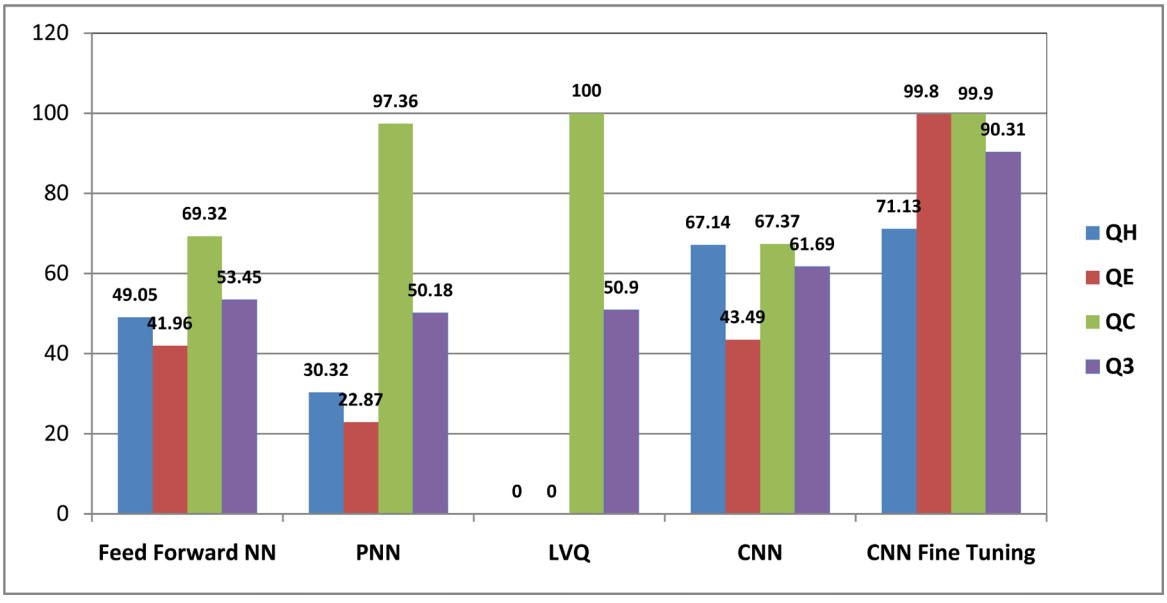


Figure 9. Final comparison of prediction accuracy for all models.

large amount of data in the dataset to maximize their effectiveness, approximately more than 500 protein samples in the dataset. The results also show that FNN network can achieve better prediction accuracy when compared with only PNN and LVQ. For PNN structure, there is high difference between prediction accuracy of (H, E) and C in other side, and PNN has faster training time and large number of neurons in hidden nodes in comparison with another implemented structures. For LVQ, it has the slowest training time in comparison with other structures and this structure can be trained and predicted only in coil (C), so it has higher prediction accuracy (100%) and completely predicts correctly, and other classes (H, E) completely predict wrongly (0%) due to its limited capability to classify complex problem as our problem(PSSP).

## References

[1] Xiong, J. (2006) Essential Bioinformatics. Cambridge University Press, Cambridge.

[2] Buatan, K. (2007) Protein Secondary Structure Prediction from Amino Acid Sequence Using Artificial Intelligence Technique.

[3] Tsilo, L.C. (2008) Protein Secondary Structure Prediction Using Neural Networks. Doctoral Dissertation, Rhodes University, Grahamstown.

[4] Pollastri, G., Martin, A.J., Mooney, C. and Vullo, A. (2007) Accurate Prediction of Protein Secondary Structure and Solvent Accessibility by Consensus Combiners of Sequence and Structure Information. *BMC Bioinformatics*, **8**, 1. https://doi.org/10.1186/1471-2105-8-201

[5] Mathwork. Last Visited 10 June 2016. http://www.mathworks.com/help/bioinfo/examples/predicting-protein-secondary-structure-using-a-neural-network.html

[6] Singh, M. (2006) Predicting Protein Secondary and Super Secondary Structure.

[7] Schmidt, W.F., Kraaijveld, M.A. and Duin, R.P. (1992) Feedforward Neural Networks with Random Weights. *Conference B: Pattern Recognition Methodology and Systems*, *Proceedings* of 11*th IAPR International Conference on Pattern Recognition*, **2**, 1-4. https://doi.org/10.1109/icpr.1992.201708

[8] Rao, P.N., Devi, T.U., Kaladhar, D., Sridhar, G. and Rao, A.A. (2009) A Probabilistic Neural Network Approach for Protein Superfamily Classification. *Journal of Theoretical and Applied Information Technology*, **6**, 101-105.

[9] Sawant, S.S. (2015) Introduction to Probabilistic Neural Network—Used For Image Classifications. College of Engineering and Research, Pune.

[10] Naoum, R.S. and Al-Sultani, Z.N. (2013) Hibrid System of Learning Vector Quantization and Enhanced Resilient Backpropagation Artificial Neural Network for Intrusion Classification. *International Journal of Research and Reviews in Applied Sciences* (*IJRRAS*), **14**, 2.

[11] Soleiman, E.M. (2014) Intrusion Detection System Using Supervised Learning Vector Quantization. Maleke Ashtar University of Technology, Tehran.

[12] Haykin, S.S., Haykin, S.S., Haykin, S.S. and Haykin, S.S. (2009) Neural Networks and Learning Machines. Vol. 3, Pearson, Upper Saddle River.