PREDICTION OF IT JOBS USING NEURAL NETWORK TECHNIQUE

Dr. Nedhal A. Al-Saiyd[*], Amjad S. Al-Takrouri[**]

[*] Computer Science Department
Faculty of Information Technology,
Applied Science University, Amman-Jordan
nedhal_alsaiyd@yahoo.com

[**] Software Engineering Department
Faculty of Information Technology,
Applied Science University, Amman-Jordan
amjad9791@gmail.com

ABSTRACT

The problems with interviewing and selection of graduated students to work and occupy the right job according to their qualifications, still presents a great challenge for employment organization, and IT companies. There are several methods by which one can predict the appropriate job that is qualified to person's skills, but none of them is quite accurate. ANN models are very effective in predicting. In order to test the effectiveness of the developed ANN models, coefficient of determination and likelihood function that minimizes the root mean squared error were used. This research investigates how IT jobs predicting changes with respect to the skills and experiences, knowledge of the graduated students' situations using back propagation artificial neural networks (ANN). To achieve the task result, data sets were taken from 50 graduated students. The data used in the multilayer feed-forward neural networks uses back-propagation algorithm models are arranged in a format of 35 input factors parameters for testing , and 35 input factors for training for 50 persons that cover IT job skills.

Keywords: Neural Network, IT Job Prediction, Back propagation, Feed-forward, Neurons

1 INTRODUCTION

In last decades, soft computing (SC) has been used largely in sciences and engineering disciplines. Artificial Neural Networks (ANNs) are different from conventional computing paradigms; they are suitable to solve non-linear problems with imprecision, approximation, uncertainty and partial truth [1].

Artificial Neural Networks is the system that consists of a number of processing units and operating parallel and is designed in a similar structure of human brain. One of the characteristics of ANN, it does not need detailed information about the system and is independent of statistical distribution of the data. Also, ANN can define the complicated interrelationships inside the system [2]. The ANN has the ability to establish the relationship between input-output data, make interconnections between input and output layers, and working in parallel to find the coefficients between the inputs and outputs of a system. All jobs are done in a learning system [3], [4].

Recently, ANNs have been successfully applied to solve a variety of problems that involves transactions on classifying, predicting, associative memory, control systems, optimization and decision making[5],[6]. Artificial Neural Networks modeling can well provide the correlation of nonlinear time series in delay state space and is considered as an effective analysis method for prediction. Therefore, ANN can achieve the purpose of prediction.

The rest of the paper is organized as follows: Section ‘2’ describes the motivational background of development methodologies and the related work to prediction model. We introduce some fundamental

analysis and technical analysis schemes and we point out the defects of the fundamental and technical analysis. Section '3' presents the notification of using back-propagation neural network in predicting Software engineering jobs. It shows the main features to solve the problem of accurate prediction and the results. Finally, section '4' contains our conclusions and outlines the future work.

2 MOTIVATION AND OBJECTIVES

Some companies search for who is the most suitable person for a certain job position, either in the interview sessions or reading large number of Curriculum Vitae (CVs); which takes a lot of time and efforts**.** Also, there are many persons wonder about what is the job that is qualified to their knowledge and skills in software engineering disciplines.

Neural network is particularly useful in solving such problems of considerable complexity that cannot be expressed as a series of steps and classifying and predicting the appropriate job depending on many objective factors as the input data. Their most important advantage is in solving problems that are too complex for conventional technologies - problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found [6].

A new model based on back-propagation artificial neural network has been developed to predict **IT jobs** from the analysis of the objective factors. The research explores a neural network model that predicts the software engineering jobs. The experimental results shows that the prediction model has good prediction effect.

3 RELATED WORK

Many researchers studied and applied ANN as function estimators in different domains, but to our knowledge none of them predict suitable jobs in software engineering disciplines. ANN has not been examined for this area.

Komar and etal applied ANN and fuzzy logic to predict the existence of a defect in file of software release. They approved that artificial neural network provides better results compared to fuzzy inference system; where the applicability of best method depends on the availability and the quantum of the data [1].

To predict the market, the analysts use many methods, which involve a fundamental analysis and a technical analysis. But results showed that none of them are completely efficient in forecasting the future of the stock price because different people have different views when they interpret the same fundamental and technical analysis reports. Also, people have achieved a prediction by using the advanced modeling methods like neural network techniques.

There were many available trials to build prediction models. Nearly all the available models were statistical correlation between properties and the classification data. In 1991 Barrick and Mount's seminal established a classification of personality characteristics; which is known as the "Big Five personality factors": *Extraversion*, *Emotional Stability*, *Agreeableness* (Satisfaction), *Conscientiousness (diligence*), and *Openness to Experience*. These factors are often studied in relation to different outcomes.

Researches during the past decades had found several numerical predictors that have positive influence on academic success in educational system. They worked on Agreeableness and *Conscientiousness*, Conscientiousness and *Openness* predictors. These predictors are studied in relation to various results that were used to predict academic performance, job performance, or college grades. While others used GPA and the grade received in a single college course to predict the academic success [8].

The hierarchical multiple regression analysis to predict academic performance when work drive was entered before the Big Five variables, the Big Five variables did not add significantly (either as a set or individually) to the prediction of course grade. They took in their consideration the cognitive ability. Work drive is "an enduring motivation to expend time and effort to finish projects, meet deadlines, be productive, and achieve success"[9].

Plant E. A., Ericsson K. A., Hill E., and Asberg K. shows that the amount of study by college students is a poor predictor and has negligible relationship to academic performance in a university setting. They predict statistically the relationship between the study behavior in college and cumulative GPA; which is potential measure of academic performance in college [10].

Previous research on the acquisition of expert performance has shown that the level of expertise in a domain is closely related to the amount of high quality, focus on practice, planned practice that individuals have collected during many years of committed training [11]

Also, there is increasing interest in the role of organizational structure to predict academic job satisfaction and job performance; which are related to self-efficacy and perceptions of social context. The hypothesized relationships among variables are supported by structural equation modeling. Job performance was positively predicted by job satisfaction [12], [13].

4 FEEDFORWARD NEURAL NETWORK TO PREDICT IT JOBS

With the advent of modern computer technology and information science, sophisticated information systems can be built and can make decisions or predictions based on information contained in available past data. Such systems are called learning systems and are currently used for the purpose of classification and prediction [14], [15].

In this paper, we have attempted to use the neural networks technique in **the predictions on IT job and Career** using Back Propagation Neural network (BPNN).

The BP algorithm is the most often chosen NN algorithm for the following conditions:

- It is not possible to discover a formula that relates the event with the causing variables;
- There are a lot of available data examples that discover the relationship between the event and the causing variables.
- The problem domain predicting IT job has some repeated features and rules hidden behind the data, which reflect mathematical relationship with jobs. The BP algorithm is easy to implement
- The user is required to "propagate" beyond the training data.
- The BP algorithm has the definite ability to estimate any function;
- The BP algorithm has excellent fault tolerance properties;

Neural networks (NNs) have proved to be efficient in a number of applications. NNs have powerful capabilities to solve a variety of prediction and forecasting problems. They have the ability to train, learn, and generalize from previous experiences and it is suited to predict models and give the correct response. NNs have been used in estimating the parameters of a formal model through its extremely parallel-distributed structure to emulate the model itself to predict future outcomes. The training process depends on few priori assumptions for the problem under study, and the training process is provided by training datasets. The learning process depends on a training process and it is used to find the functional relationships among the datasets even the underlying relationships are hard to describe or unknown, and adjust the network weights. After the training process is successfully finished, the neural network will identify a certain result when a new set of data is presented to its input layer. This process is called a generalization process. Moreover, the major task of the learning algorithm is to find the NNs weights, which is calculated from the minimum difference between the NNs output and the desired output result.

The most popular and successful learning algorithm for feed-forward Neural Networks utilizes the back-propagation multi-layered feed-forward Neural Networks with a sigmoid training function. 95% of business applications using neural networks apply multi-layer feed-forward with back propagation [16] .In this section, the BP learning algorithm is selected to develop a suitable model for IT jobs prediction problem. The NN architecture model contains of three main layers: input layer, hidden layer, and output layer. Figure 1 shows the structure model of multi-layer back-propagation neural network ML BPNN.

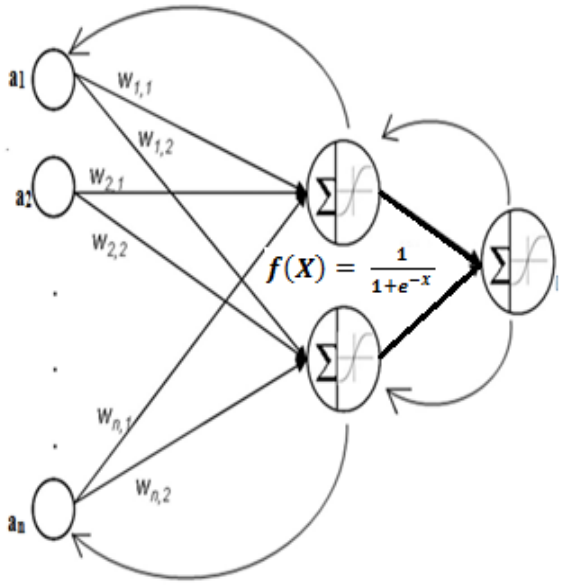

Figure 1: Back-Propagation Multi-Layer Neural Network

Each layer consists of a number of elements called neurons a_i. Typically, every neuron in one layer is only connected to neurons in the next layer. Each neuron in the input layer is assigned with independent factor value obtained from the training patterns. The input layer neurons do not do calculation; the input neuron merely sends a copy of its value to all the hidden layer neurons. Each neuron in a hidden layer receives a number of input values, multiplies each of them by the corresponding weight value. The weight **Wij** is an adaptive coefficient that determines the strength of the connection between two neurons. The result from the multiplication is summed and passes through a transfer function that maps the inputs into the (0, 1) range to produce a single output. The neuron output will not be activated unless the summation exceeds certain threshold. This operation can be represented by the following two equations:

$$\boldsymbol{u_j} = \sum_{i=1}^{n} \left(\boldsymbol{W_{ij}} * \boldsymbol{a_i} \right) \quad (1)$$

$$f(x) = \frac{1}{1+e^{x}} \quad 0 \ll f(x) \ll 1 \tag{2}$$

Where:

f(x) = neuron output,
$\mathbf{u_j}$ = input to the transfer function,
$\mathbf{a_j}$ = i^{th} input,
Wj = weight of connection i,
x = gain of sigmoid function, and
n = number of inputs to one neuron.

The learning process adjusts all the trainable weights through feeding back the error information through the system to reduce the overall error and let the output approach the desired output and close to expected output. The equation Eq. (3) calculates Current Gradient Descent Value using Eq. (4), and Eq. (5):

$$\frac{dE}{dW_{ik}} = dE * O_1 \tag{3}$$

Where:

$$dE = f_i^{\backslash}(\sum H_1) * \sum W_{ki} * \delta_1 \tag{4}$$

$$\delta_1 = -E * f_1^{\backslash}(\sum O) \tag{5}$$

$\mathbf{O_1}$ **=**Activation function value of output neuron.

After calculating the Gradient Descent Equation, we will actually update the weights, using the gradients onto Back-propagation equation, as shown in Eq. (6):

$$\Delta W_t = E \frac{dE}{dW_t} + \alpha \Delta W_{t-1} \tag{6}$$

Where:

α = Momentum which is, Back-propagation can easily get stuck at a local minimum.
$\Delta \mathbf{W_{t-1}}$ = the previous change from previous iteration.
E = the learning rate. This shows the degree to which the deltas will affect the current network.
$\frac{dE}{dW_t}$ = Current Gradient Descent Value.

4.1 The Structure of Multilayer Network

Selection of the most adequate NN is not a straight forward task, since it depends on several different factors such as the prior knowledge, the nonlinearity of the problem, the set of independent input parameter, the dependent output, the number of training data values, etc.

The input layer contains a number of neurons equal to the number of delayed measurements allowed to build the network model. In our case, there are thirty five inputs to the network. The hidden layer consists of linear hidden units. The output layer consists of one output neuron producing the estimated value of the fault. There is no direct connection between the network input and output. Connections occur only through the hidden layer. The hidden units are fully connected to both the input and output layers. The hidden and output layer nodes have linear activation functions. Usually the transfer function is a sigmoid function. Every neuron receives the weighted sum as its inputs from its previous layer, and generates an output based on the weighted sum, which, in turn, passes to the next layer. The purpose of this network is to minimize the Euclidean distance from the difference between the resulted output and the desired output by adjusting the set of weights at every epoch. When the network error falls within the pre-defined error bound, the BP training process stops.

4.2 Number of Neurons in Hidden Layers

It is difficult to set up the number of neurons needed in the hidden layer. Unfortunately, there is no formula to choose the right number of neurons for a given problem. Too few neurons may lead to an insufficient ability to learn, require more epochs to train, besides has difficulty to meet the performance criteria. Using too many neurons can increase the computational time it takes to train each epoch, and also over fits the data. Obviously, some compromise must be reached between too many and too few neurons in the hidden layers [17]. A general method is to choose the number of neurons that satisfy the problem's criteria, and then to add a few more neurons to optimize the convergence speed.
There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers. Some of them are summarized as follows [17].

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer.
- The number of hidden neurons should be less than twice the input layer size.

An arbitrary number of neurons in the hidden layer is chosen, it start with the number of hidden neurons equal to 2/3 of the input layer size plus the size of the output layer [17]. Then the optimum number is found using the "forward" and "backward" selection approaches and retraining the net. The optimum number of neurons in hidden layer was 24.

Table 1: Parameters Setting

Parameter	Value
Error Rate	0.01
Learning Rate	0.35
Momentum	0.45
Number of Neurons in Input layer	35
Number of Neurons in Hidden layer	24
Number of Neurons in output layer	1

4.3 NN Input Parameters

There are 35 of factors that affect

- GPA.
- Having oral and written English skills.
- Active in events workshops and contests.
- Extra lectures/Training Courses.
- Theory of Big Five Factors: Extraversion, Emotional Stability, Agreeableness, Conscientiousness, and Openness to job Experience.
- Imaginative and good at creative reasoning
- Able to direct the work of others.
- Willing to continuously update personal IT skills and knowledge.
- Able to accept responsibility.
- Familiar of fundamental knowledge of web development such as "PHP", "ASP", "XML" "HTML5", "HTML".
- Knowing Structural approach.
- Knowing OOP approach.
- Able to understand and solve complex problems.
- Good appearance.
- Good researcher.
- Committed to understanding new technology and upgrading skills.
- Having experience in Software Testing techniques and quality assurance.
- Knowing about establishing a detailed program specification.
- Knowing about Decision tree.
- Knowing about SQL, Oracle, JDeveloper, etc.
- Knowing Reusable Software Component.
- Knowing Software Engineering Management.
- Knowing Data Structure & Algorithms.
- Knowing more than one programming language skills and experience.
- Having knowledge about analysis modeling.
- Having knowledge about analysis design modeling.
- Having knowledge about analysis Design Methodologies & techniques
- Having knowledge about analysis Methodologies & techniques.
- Having oral and written communication skills.
- Having the ability to use the UML Case Tool.
- Programming skills and experience.
- Having Knowledge of object-oriented languages and tools such as C++ and Java.
- Able to think logically and analytically in a problem-solving environment.
- Able to work independently or as part of a team.
- Having knowledge about control concurrency problems and solutions.

4.4 Expected Output

- System analysis.
- Architecture Designer.
- Developer.
- Testing Engineer.
- Project Manager.
- Software Maintenance Engineer.
- Research Assistant.
- Computer Instructor.
- Database Administrator.
- Web Designers/Developers.
- Database Design Specialist.
- Software Designer.
- Database Programmer (Systems).
- Systems Software Programmer.

4.5 Experimental Data

The combinations of ANN parameters are clearly identified to provide a best result. The data were divided into training and testing groups of different sizes. The training and the test data of ANN are obtained from the experimental studies. Fourteen job characteristics of software engineering domain are identified and various job skills levels (Low, Medium, and Good) and they are 35 skills have been established. The fourteen job characteristics have been selected and they are used as the inputs to neural network model.

The data used in the multilayer feed-forward neural networks uses back-propagation algorithm models are arranged in a format of 35 skills input parameters for testing and 35 input for testing and training for 50 persons.

The combinations of ANN parameters are clearly identified. The experimental results show that the prediction model has good prediction effect.

5 EXPERIMENTAL RESULTS OF PREDICTION

In the following section, we will show the prediction of the IT Jobs Model, using NNs in the training and in the testing cases**.** The NN was trained with a different set of initial weights until the best sets of weights were calculated and the Root Mean Squared Error *(RMSE)* was reduced to a small value. We used the NNs weights developed from the

training case to test the NN performance. The NNs model has been tested with the rest of the collected data, which represents 100% of the collected data set.

Step1: Choosing the skill's levels manually for each job's factor; indicated in the user interface by "low", "Medium", and "Good". Or Loading 'csv' file each skill separated by "," with a 1, 2 or 3 values, The skill levels are represented internally in the function "INPUT[][]" as 1, 2, and 3 respectively, that perform this task.

Step 2: Inserting the "Error rate"; 0.01 is the default value.

Step3: Clicking the "Train" button to start the learning process. This will start showing the current iteration number and the resultant error. At the end of the learning process, the screen shows the over-all iteration numbers, "Epoch #", and iteration calculated error.

Step4: Clicking the "Test Network" button at the end of the execution, the system will test the "Data Set" by entering only input factors to predict the Output "Actual[][]"(Predicted) function and compare with the IDEAL[i][0] of the "Data Set" as shown in Figure 3 at "Neural Network Results" screen., the resultant window displays as a meaningful "Job Name" instead of numerical representation, the predicted value.

The evaluation criterion is used for the developed model to measure its performance. The criterion of evaluation (i.e. performance) was defined as the Root Mean of the Square Error (RMSE) The RMSE of the testing in NN case is given in Table 2.

6 CONCLUSION

The aim of this research paper is to analyze, formulate, construct, and evaluate back-propagation artificial neural network model in prediction of IT jobs with high accuracy. The developed model is well-suited for the analysis of many objective factors, for persons with qualified knowledge, and skills in software engineering disciplines. This model uses thirty five attributes in input layer, 24 neurons in hidden layers. The combinations of ANN parameters are clearly identified. The experimental results show that the proposed prediction model has good prediction effect and good performance.

Predicting IT jobs will help to make significant improvements in the experience and skills of users searching for jobs, and help employers and jobseekers to bring more transparency and to discover the market qualification of different job positions.

ACKNOWLEDGMENT

The authors are grateful to the applied science private university, Amman-Jordan, for the partial financial support granted to cover the publication fee of this research article.

7 REFERENCES

[1] Komar V., Sharma A., and Kumar R., Applying Soft Computing Approaches To Predict Defect Density In Software Product Releases: An Empirical Study, Computing and Informatics, Vol. 32, 2013, 203-224, 2013. Available at: http://web.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=4&sid=b331afee-f950-4ce2-8b61-bf41be3fe172%40sessionmgr115&hid=118.

[2]. Daunoras J., Gargasas V., Knys A., Narvydas G., "Application of Neural Classifier for Automated Detection of Extraneous Water in Milk", Elektronika ir Elektrotechnika (Electronics and Electrical Engineering), No. 9, pp. 59–62, 2011.

[3] Mellit A., Kalogirou S. A., "Artificial Intelligence Techniques for Photovoltaic Applications: A review", Progress in Energy and Combustion Science, Vol. 34, No. 5, pp. 574–632, 2008. Available at: http://dx.doi.org/10.1016/j.pecs.2008.01.001.

[4] Ciylan B., "Determination of Output Parameters of Thermoelectric Module using Artificial Neural Networks", Elektronika ir Elektrotechnika (Electronics and Electrical Engineering), No.10, pp. 63-66, 2011.

[5] Ball S., Güler N., Yakar N., "A Prediction of Unemployment Rates By Using Artificial Neural Networks via People's Educational Background", in Proc. of the International Conference on Modeling and Simulation, Konya, Turkey, pp. 947–951, 2006.

[6] Graupe D., Principles of Artificial Neural Networks, World Scientific Press, Singapore, p. 320, 2007.

[7] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/cs11/article1.html.

[8] Ridgell, S. D., Lounsbury, J. W, Predicting academic success: general intelligence, "Big Five" personality traits, and work drive, College Student Journal Publisher, 2004. Available at: http://www.freepatentsonline.com/article/College-Student-Journal/126386901.html.

[9] Lounsbury, J. W., Sundstrom, E., Loveland, J. M., & Gibson, L. W. Intelligence, ''Big Five'' Personality Traits, and Work Drive as Predictors of Course Grade, Personality and Individual Differences, No. 35, pp. 1231–1239, 2003. Available at: http://www.sciencedirect.com/science/article/pii/S0191886902003306

[10] https://castl.duq.edu/Conferences/Library03/PDF/GPA/Ashby_Plant_E.pdf.

[11] Plant E. A., Ericsson K. A., Hill E., Asberg K., Why Study Time Does Not Predict Grade Point Average Across College Students: Implications of Deliberate Practice for Academic Performance, Contemporary Educational Psychology 30, pp. 96–116, 2005.

[12] Borgogni, L., Russo S. D., Petitta L., and Vecchione M., Predicting Job Satisfaction and Job Performance in a Privatized Organization, International Public Management Journal, 2010.

[13] Lounsbury J. W., Moffitt, L., Gibson, L. W., Drost, A. W., & Stevenson, M. W., An Investigation of Personality Traits in Relation To The Job And Career Satisfaction of Information Technology Professionals, Journal of Information Technology, No. 22, pp. 174–183, 2007.

[14] Naik, B., & Ragothaman, S., Using Neural Networks to Predict MBA Student Success, College Student Journal, Vol. 38, No. 1, pp. 143-149, 2004. Available at: http://sunburst.usd.edu/~bnaik/research/NEURAL.pdf,

[15] D'Souza, K. A., Maheshwari S. K., Predicting And Monitoring Student Performance In The Introductory Management Science Course, Academy Of Educational Leadership Journal, Vol. 15, No. 2, 2011. Available at: http://alliedacademies.org/Publications/Papers/AELJ%20Vol%2015%20No%202%202011%20p%2069-80.pdf

[16] Smith, K.A., and Gupta, J.N.D., "Neural Networks in Business: Techniques and Applications for the Operations Researcher," Computers & Operations Research, Vol. 27, No. 11, pp. 1023-1044, 2000. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.5430...pdf

[17] Heaton J., Introduction to Neural Networks with Java, First Edition, Second printing, Heaton Research Inc., 2005.

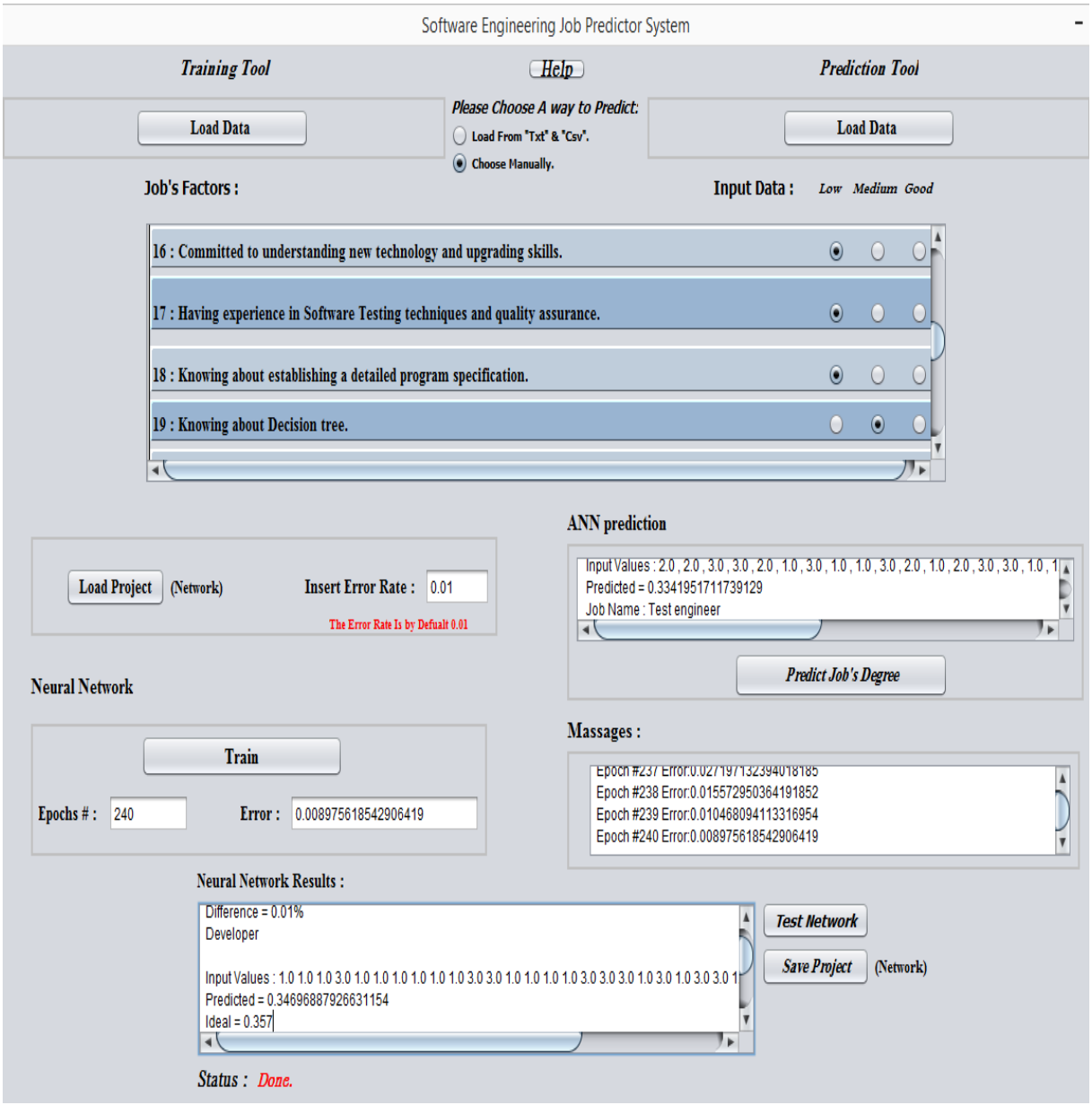

Figure 3: Testing Screen Shot.

Table 2: Test Table of variables

Variable Name	INPUT[][]	IDEAL[][]	Input Neurons	Hidden Neurons	Output Neuron	Error Rate	Learn Rate	Momentum
Value	{1,2,3}	0 to 1	35	24	1	0.01	0.35	0.45