

An Application of Artificial Neural Network (ANN) Process to Assess Risk in Cement Industries in Bangladesh

Syimun Hasan Mehidi, Nayan Chakrabarty, H.M. Mohiuddin

Department of Industrial Engineering and Management (IEM), Khulna University of Engineering & Technology (KUET), Khulna- 9203, Bangladesh

Abstract

Today large and mid-size companies have more complex operations which involve different types of risk and are difficult to identify the risk. To extract these risks and analyze it scientifically, a professional risk assessment tool namely ANN is most appropriate and flexible network technology. This paper briefly presents Artificial Neural Networks (ANNs), universal and highly flexible functions, approximations for any data as an application to assess risk in cement industries for instance HOLLAND CEMENT BANGLADESH LIMITED has been selected to assess risks. To assess the risks a three-layer feed forward architecture of 10 risk factors which are independent and their strength of relationship was used as relative weight of input variables. Six neurons of hidden layer and two neurons of output layer with back propagation algorithm were designed using Neural Network Toolbox of MATLAB. The input data normalized to the range 0–0.9 and initial weights were randomly selected. The algorithm propagates the weights backwards and then controls the weights. The factors were inputted to a MATLAB-based application program and the numbers of iterations were set 9000. After implementation the overall process the adjustment weights were trained and by using adjusted weights actual output was found 0.63 in which predicted value was 0.66 with 95.45% prediction success, the results very promising.

Keywords: Artificial Neural Network (ANN); Risk assessment process; MATLAB; Risk factor

Introduction

Risk assessment provides a mechanism for identifying possible risks and opportunities which represent potential pitfalls that could affect the achievement of organizational objectives, positively or negatively. While organizations have been conducting risk assessments for years, many still find it difficult to remove their real value [1]. The linkage of risk assessment to drivers of shareholder value and key objectives has sometimes been lost. A large amount of the manufacturing industries in Bangladesh don't follow correct procedure to assess the potential risk. Most time they fail to identify the potential risk as well as their qualitative or quantitative analysis. As a result organization has been failed to meet its objectives within due time and it causes of losing big amount of money [2]. So it is important factor to identify the potential risk in the industry and assess their real value to avoid any unexpected event and to take necessary steps against the potential risk that can hamper to achieve organization's objectives. A robust risk assessment process, applied consistently throughout the organization, empowers management to better identify, evaluate, and exploit the right risks for their business, all while maintaining the appropriate controls to ensure effective and efficient operations and regulatory compliance [3]. In this paper work to assess the potential risk ANN process, which is called computational or innovative tool, most widely used in manufacturing industry.

In this paper we developed this method to study risk management process and Artificial Neural Networks (ANN) process as well as this proposed method also helps to figure out the risk variables and assess risk in cement industries in Bangladesh through applications of ANN.

In this paper first section represent literature review, second section represent a ANN tool for accessing risk, third section represent methodology, fourth section represent result, fifth section represent project success and failure, finally shows discussion, conclusion and references.

Literature Review

In this section describe risk, defining risk assessment, a process

for capturing and analyzing risks, purpose and applicability, neural network in risk assessment on concept of Risk Assessment, Artificial Neural Networks (ANN) for determining risk. Source of information were obtained from books, journals, reports. The history of neural networks started in mid twentieth century when simple neural network with limited capabilities were conceived. They never got into the main stream applications at that time due to poor generalization capabilities and lack of specificity with high memory loads. After two decades, the whole concept of neural networks changed when the multi-layer neural networks with back propagation learning algorithm was presented. From that time, many different researchers studied the area of artificial neural networks, which led to a vast range of different neural architectures applied to a plethora of different problems. At present; neural networks are used as principal solutions for various problems like grouping and classification, pattern recognition, approximation, prediction, cauterization and memory simulation. Neural networks may initially seem complex and computer intensive, but actually may integrate well with a clinical environment. It has been used widely in research and for practical applications since the early 80's. Most of the work used fixed ANN topologies and standard off the shelf learning algorithms like back propagation (BP). The learning algorithm generally only trains the connections weights and unit bias. The problem of designing a near optimal ANN architecture for an application is still largely done on a

***Corresponding author:** Syimun hasan Mehidi, Department of Industrial Engineering and Management (IEM), Khulna University of Engineering & Technology (KUET), Khulna-9203, Bangladesh, Tel: 880412870054; E-mail: Mehidiku09@gmail.com

Received May 05, 2014; **Accepted** October 01, 2014; **Published** October 08, 2014

Citation: Syimun Hasan Mehidi, Nayan Chakrabarty, H.M. Mohiuddin (2014) An Application of Artificial Neural Network (ANN) Process to Assess Risk in Cement Industries in Bangladesh. Ind Eng Manage 3: 138. doi: [10.4172/2169-0316.1000138](https://doi.org/10.4172/2169-0316.1000138)

Copyright: © 2014 Mehidi SH, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

trial and error basis. However it is an interesting issue there is strong biological evidence that the information processing capabilities of an ANN are determined by its architecture [2].

Although often considered a relatively new phenomenon initial attempts at artificial neural networks date back to the early 20th century. In 1943 Warren McCulloch and Walter Pitts co-authored a paper outlining a model of a simple neural network with electronic circuits. Later, as computers emerged in the 1950s, several researchers attempted to utilize the new technology to create better neural networks. Over the next decade or so these physiologists, psychologists and computer engineers contributed greatly to the development of artificial neural networks. One of these early pioneers was Frank Rosenblatt, a neurobiologist at Cornell that was researching vision in flies. The neural processing that occurred within the eye itself particularly intrigued Rosenblatt and formed the basis of his "Perceptron" neural network. The Perceptron and other models showed great promise with many initial successes.

Unfortunately, the early successes of artificial neural networks caused a great deal of hype within the media. This eventually led to disappointment as earlier claims were left unfulfilled. In part this was due to the very limited computing power available at the time. However, there were also conceptual limitations to progress. In 1969 Marvin Minsky and Seymour Papert published a book in which they discussed some of the limitations of the Perceptron model. The effect of these problems was to limit much of the funding available for research into artificial neural networks.

Although funding was minimal several scientists continued to develop neural network models. Paul Werbos worked to improve the earlier Perceptron model and created the now popular back-propagation network (though, it wasn't until it was rediscovered in 1986 by Rumelhart and McClelland that it became widely used). Other researchers such as Steve Grossberg, Teuvo Kohonen, and Henry Klopff also created new models. However, it was not until John Hopfield of Caltech presented a paper to the national Academy of Sciences that general interest in the field began to resurge. Hopfield's approach was not simply to create models but to develop technologies that could be applied to real life problems [4].

Ann tool for accessing risk

ANN can be used as a prediction tool to access risk for minimizing error where error is defined as the difference between desired value and output value. The process of minimizing error can be executed by back propagation algorithm which comprises the following steps [5].

A. Feed forward propagation: Feed forward propagation comprises the following steps:

- a. Evaluation of output of input layer to hidden layer
- b. Evaluation of output of hidden layer to output layer
- c. Evaluation of Error

a. Evaluation of output of input layer to hidden layer: To evaluate output of input layer at first the weight factor of each input variable is calculated. The total weight of input layer to hidden layer is equal to

$$W = (m + 1) \times n$$

Where, W=Total weight of input layer, m=Total no of input variable and n=Total no of hidden layer.

The initial value of weight factor is to be chosen randomly which

should be (-0.3 and +0.3). In the input layer, the state of each neuron is determined by the input variable.

$$net_j = \sum_{i=1}^{m+1} w_{ij} \times x_i$$

Where W_{ij} is the weight factor of input layer to hidden layer and X_i is the input variable. The activity of neuron is calculated by a sigmoid function:

$$O_j = 1 / (1 + \exp^{-net_j})$$

Where, O_j is the output of input layer.

Evaluation of output of hidden layer to output layer: The output of input layer is represented the input of hidden layer. The total weight of hidden layer to output layer is equal to $W = (n + 1) \times O$

Where, n=Total no of hidden layer, O =Total no of output layer. In hidden layer the state of each neuron is determined by:

$$net_k = \sum_{j=1}^{n+1} w_{jk} \times o_j$$

Where, W_{jk} is the weight factor of hidden layer to input layer and O_j is the output of input layer.

The activity of neuron in the hidden layer is calculated by the following function.

$$O_k = 1 / (1 + \exp^{-net_k})$$

Where, O_k is represented the output of output layer.

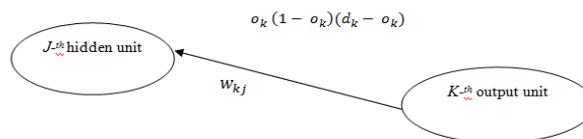
b. Evaluation of error: The error, E is measured from the desired value, d_k and the actual output value, O_k . The training, performed on a represented data set, runs until the sum of squared errors (SSE) is minimized:

$$SSE = \frac{1}{2} (d_k - O_k)^2$$

Where, SSE is the sum of squared errors. d_k =desired value and O_k =actual output value.

B. Back-propagation to the output layer

The back-propagation path from the output of the network up to the output unit k is shown as below:

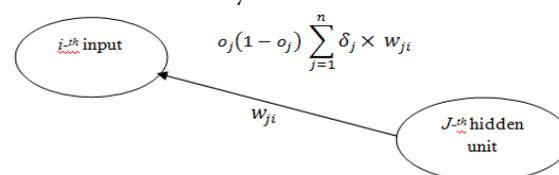


From this path we can collect by simple inspection all the multiplicative terms which define the back-propagated error δ_k . Therefore

$$\delta_k = O_k (1 - O_k) (d_k - O_k)$$

Where, d_k =desired value of Output layer, O_k =Output of Output layer value.

C. Back-propagation to the hidden layer: The back-propagation to the hidden layer shown as below



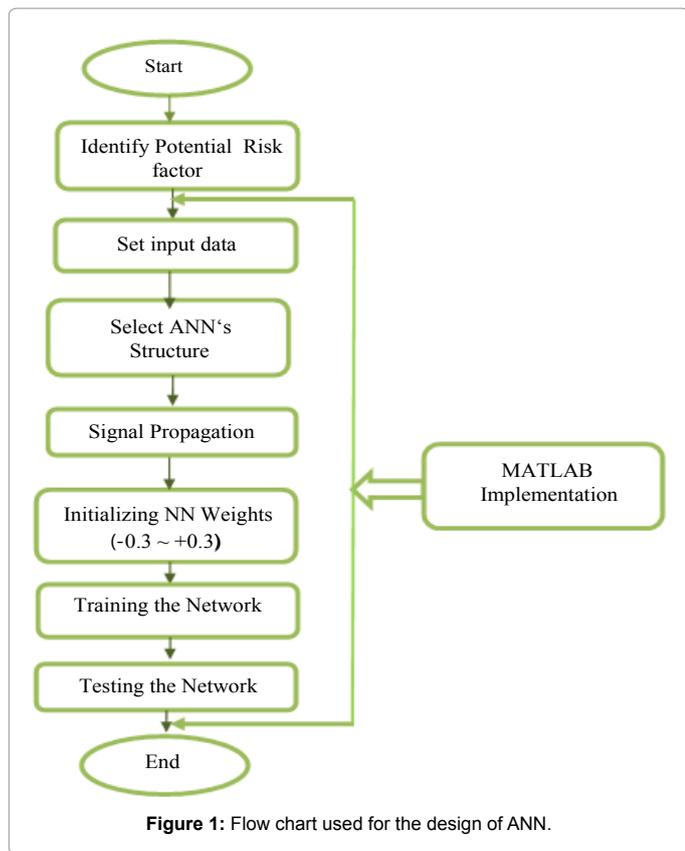


Figure 1: Flow chart used for the design of ANN.

Therefore the back-propagated error for hidden layer is δ_j and represent by

$$o_j(1-o_j) \sum_{j=1}^n \delta_j \times w_{ji}$$

Weight update: Adjustment of weight, layer by layer, is calculated from the output layer back to input layer. The correction is made by:

$$\Delta w_{ji} = LR \times \delta_j \times f'(net_j)$$

$$\Delta w_{kj} = LR \times \delta_k \times f'(net_k)$$

Where, Δw_{ji} is the adjustment of weight between neuron j and neuron i from the previous layer. $f'(net_j)$ is the output of neuron i and LR is the learning rate and δ_j depends on the layer [6].

Methodology

The whole process how artificial neural network works are used to assess in the cement industry: Holcim cement Bangladesh ltd. Flow chart used for the design of ANN (Figure 1).

A. Identify potential risk factor

Data Collection and Set Table 1 for training the neural network in the below input data table, data are collected with the help of experience persons in cement industries in Bangladesh and according to the previous work in the field. The "overall assessed risk" are calculated with discussing managerial persons in cement industries. Actually this value depends on people guessing power and they feel how much impact on the objectives for the corresponding risk factors, Table 2. Every risk factor treat as an independent variable and range is (0-9) where 0 means no risk and 9 means maximum risk. Risk value can be measured by qualitative analysis chart where range (0-3) indicates Low impact; range

(4-6) indicates Moderate impact and range (7-9) indicates High impact Figure 2.

Output of learning of network In MATLAB: From the input data set we used (12*11) input matrix for calculating overall assessed risk. Since ANN can't receive the value more than 1 so every data table set value has been divided by 9. Our assessment's risk value range is (0-9) and also added one additional column of complement of overall risk assessed.

Input Matrix: Table 3.

Result

According to the above algorithm (article 3.4) the graph shows

Risk factor	Symbol
Machine failure (Bearing failure, failure of fan impeller)	X1
Electrical failure (power supply)	X2
Uncertain occurrence of hazard	X3
Lack of standard operating procedure	X4
Improper chemical composition	X5
Wrong vendor selection	X6
Blockage	X7
Leakage	X8
Insufficient storage	X9
Lacking knowledge of machine labor	X10

Table 1: The risk factors which were collected from the above cement industry are given below.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Overall assessed risk
2	5	4	8	9	5	9	8	5	6	6
0	8	9	0	7	4	8	9	0	3	5
4	0	5	4	8	8	3	5	6	9	5.2
5	2	0	5	8	6	6	1	6	5	4.5
1	6	7	4	4	8	7	3	6	4	5
3	3	4	2	0	4	6	1	7	3	4
7	6	0	5	0	6	5	9	6	4	4.8
3	5	6	9	0	4	1	2	5	5	4
6	2	0	7	2	5	2	7	4	1	3.5
2	5	6	9	3	4	1	2	5	5	4.2
0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	0

Table 2: Different Risk Factors for Ten Years.

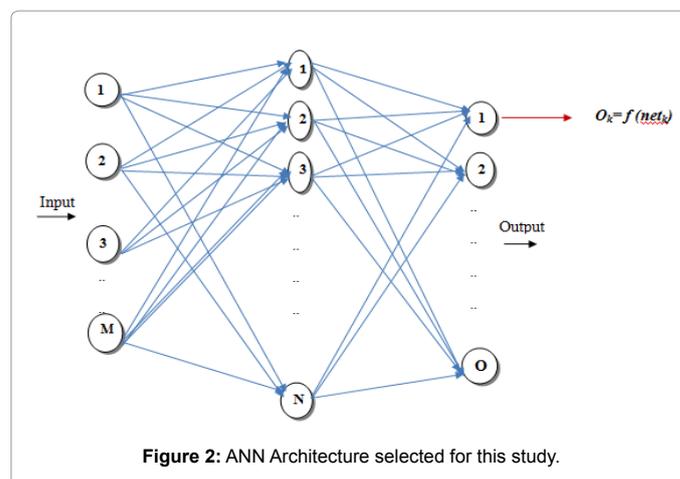


Figure 2: ANN Architecture selected for this study.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Overall assessed risk	Complement of Overall risk assessed
0.22	0.55	0.44	0.88	1	0.55	1	0.88	0.55	0.66	0.66	0.34
0	0.88	1	0	0.77	0.44	0.88	1	0	0.33	0.55	0.44
0.44	0	0.55	0.44	0.88	0.88	0.33	0.55	0.66	1	0.64	0.36
0.55	0.22	0	0.55	0.88	0.66	0.66	0.11	0.66	0.55	0.5	0.5
0.11	0.66	0.77	0.44	0.44	0.88	0.77	0.33	0.66	0.44	0.55	0.45
0.33	0.44	0.44	0.22	0	0.44	0.66	0.11	0.77	0.33	0.48	0.52
0.77	0.66	0	0.55	0	0.66	0.55	1	0.66	0.44	0.53	0.47
0.33	0.55	0.66	1	0	0.44	0.11	0.22	0.55	0.55	0.44	0.56
0.66	0.22	0.77	0.22	0.55	0.22	0.77	0.44	0.11	0.38	0.38	0.62
0.22	0.55	0.66	1	0.33	0.44	0.11	0.22	0.55	0.11	0.46	0.54
1	1	1	1	1	1	1	1	1	1	0	1

Table 3: Input Matrix of Artificial Neural Network.

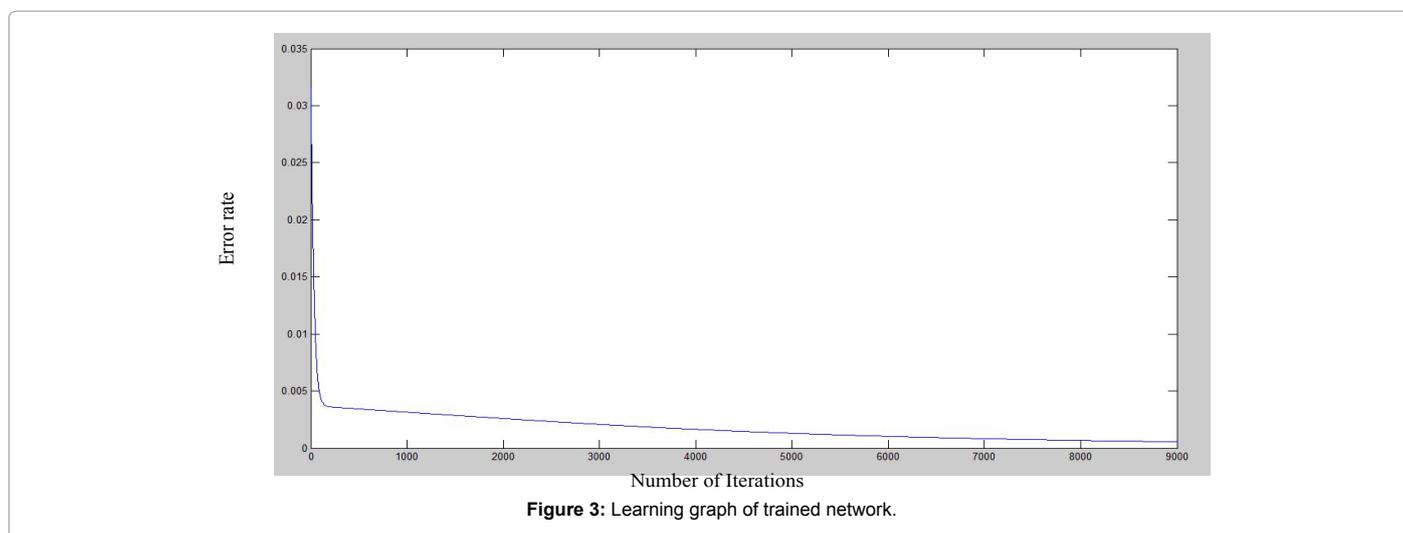


Figure 3: Learning graph of trained network.

.3158	.7620	-.7977	.7258	.4915	-.5284	-.4229	-.2853	1.0591	1.1836
-.8431	.9754	.9021	.0351	.6908	-.7786	-.1092	.7035	.4898	.9822
.3371	-.8625	.8272	.8225	.3676	.5237	.3307	-.3539	.1609	-.8542
.3115	-1.065	-.4413	-.6269	-.7731	.6594	.4260	-.4592	.7442	-.8528
-.1929	-.3175	.3970	.7494	-.1083	-.042	-.1198	.1171	.9259	.9560
-.1685	.8732	.5315	-.2101	-.9157	.1850	1.0156	-.3556	.1692	-.5154
.3561	-.3855	-.6804	-.2459	1.2578	.4759	-.366	-.722	-.2219	.4929
.3798	-1.098	.8697	.0619	.8585	-.300	-.6068	-.4978	.2321	-.0534

Table 4: Standard Weight Factors.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Overall assessed risk	Complement of Overall risk assessed
0.22	0.55	0.44	0.88	1	0.55	1	0.88	0.55	0.66	0.66	0.34

Table 5: Input Table.

that learning of the design network is correct because error rate is continuously decreasing and its lowest value trend is 0.002. So it can be expected that the trained network can predict a good assessed value of risk. Figure 3 shows the learning graph of the trained network.

- **Standard weight factor table**

The standard weight factors found from the MATLAB program are given in Table 4.

- **Standard weight factors Table 4.**
- **Sample calculation:** Here only one input matrix (1*12) is used

to calculate the hidden layer input Table 5. Calculation of input value for j -th neuron (net_j)

$$net_j = \sum_{i=1}^{m+1} w_{ji} \times x_i$$

Where, net_j is the net input of neuron j ; x_i is the output value of neuron i of previous layer; w_{ji} is the weight factor of the connection between neuron i and j .

A. Input layer to hidden layer

For hidden layer 1 Evaluation the state of the signal from the input layer

	Hidden 1	Hidden 2	Hidden 3	Hidden 4	Hidden 5	Hidden 6
net_j	1.33	1.525	-.1696	-.1344	1.355	.4587
activity of neuron, $f(net)$.790	.821	.45	.46	.79	.612

Table 6: Output of Hidden Layer.

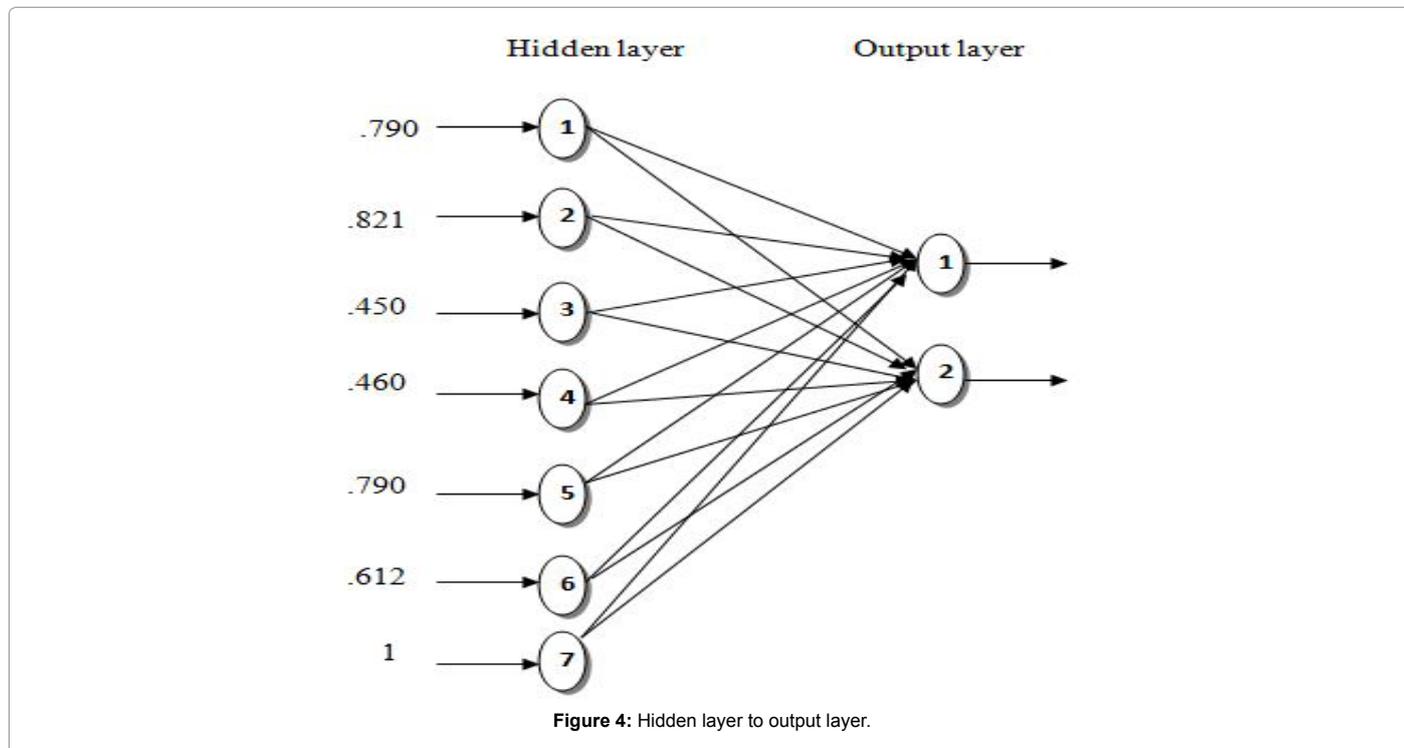


Figure 4: Hidden layer to output layer.

	Output 1	Output 2
net_k	.51	-.59
$f(net_k)$.63	.37

Table 7: Output of Output Layer.

	Desired value	Output value
Overall assessed risk	0.66	0.34
Complement of overall assessed risk	0.63	0.37
% of Correct Prediction	95.45%	
Error	0.00045	

Table 8: Comparison of Prediction Accuracy between Desired and Output Value.

$$net_1 = 22 \times .3158 + .55 \times .7620 + .44 \times (-.7977) + .88 \times .7258 + .495 + .528 \times .55 - 4229 + .88 \times .2853 + .55 \times 1.0591 + 1.1836 \times .66 - .8431 = 1.33$$

Where, net_1 = net input of neuron 1

Now the activity of neuron is determined by using sigmoid functions i.e.

$$f(net_j) = 1 / (1 + e^{-net_j})$$

$$f(net_1) = 1 / (1 + e^{-1.33}) = 1 / (1 + e^{-1.33}) = 0.79$$

According to the above calculation we get overall signal $net_1, net_2, \dots, net_6$ and activity neuron $f(net_1), f(net_2), \dots, f(net_6)$ are shown in Table 6. The Figure 4 shows in below the hidden layer to output layer. Here .790, .821, .450, .460, .790, .612, 1 are the input of hidden layer respectively. After iteration we get the desired output which is shown in Table 6 output of output layer shown in Table 7. Comparison of prediction accuracy between desired and output value shown in Table 8.

Project Success and Failure

The prediction percentage for iterations are represented in the following Figure 5 Performance percentage Vs Iterations as a function of the training. From the Figure 5 we can see that initially the performance percentage increase very quickly where for 500 and 1000 iterations the performance percentage is respectively 76.59% and 86.42%. The performance percentage became fluctuate from 1500 to 6500 iterations. After that the performance percentage became stable and the highest performance percentage was found for 9000 iterations which is 95.45% [6].

Discussion, Conclusion and Limitations

In this paper a new approach for evaluating risk based risk factors derived from an ANN model was proposed. The risk factor to assess risk was collected from a renewed company *Holcim cement Bangladesh limited*. The ANN model was implemented by MATLAB based on the risk factors and achieved a good performance. Here the learning graph of trained network showed that the error was minimized with respect to iterations and also showed the lowest possible error was 0.00045. The performance percentage graph showed the highest performance percentage obtained after 9000 iterations and the performance percentage was 95.45%. The good performance of ANN suggests that ANN is most flexible and useful tool for predicting risk in any industry.

Today large and mid-size companies have more complex operations which involve different types of risk and are difficult to identify the risk.

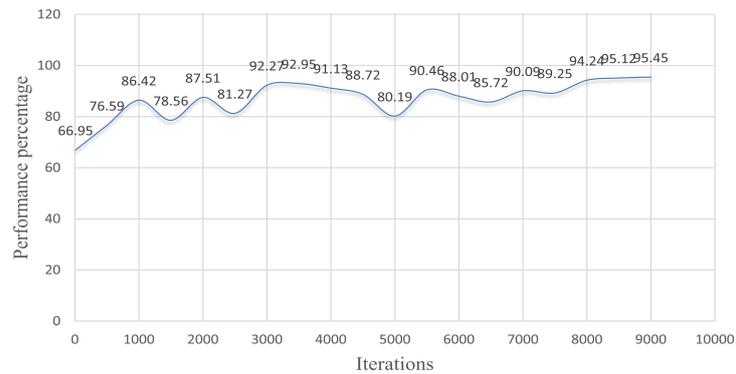


Figure 5: Performance percentage Vs Iterations as a function of the training.

To extract these risks and analyze it scientifically, a professional risk assessment tool namely ANN is most appropriate and flexible network technology.

References

1. Atkinson J, Jourdan C (2008) How principles-based risk assessment enables Organizations to take the right risks, PricewaterhouseCoopers LLP.
2. Lee Y (2007) A neural network face detector design using bit-width reduced fpu in fpga, University of Saskatchewan, Saskatoon.
3. Sarcia S A, Cantone G, Basili V (2007) A Statistical Neural Network Framework for Risk Management Process, Vol 6: 46-51.
4. Introduction to Neural Networks (2014) Marcus Bros.
5. Rojas R (1996) Neural Networks, Springer-Verlag, Berlin 165-169.
6. Lee S, HyungRyu J, Sun Won J, Park H J (2004) Determination and application of the weights for landslide susceptibility mapping using an artificial neural network, *Engineering Geology*, vol 71 : 289-302.