Short –term Predication of Risk Management Integrating Artificial Neural Network (ANN)

Malaya Nayak, Tariq Abdullah

***Abstract:** The IT industry has boomed in the past few years with an ever increasing number of risk management applications being developed. There are inherent risks in software development projects and failure to deliver software projects within deadline or failure to develop software according to specifications can be costly. The software risks may occur during the project process. The management process of software risks consists the risk refinement, risk identification, risk monitoring, risk maintenance, risk estimation and risk mitigation. Neural Network has ability to stimulate hidden pattern recognition skill. The primary study of this paper is to focus on various risk management models and how risk tools may help in mitigating software risks during the project development. With the application of Neural Network, We propose short term risk management model which can predict the risk involvement with the upcoming project risks, analyzing from the previous projects causing serious loss in the IT project in terms of values on certain risk factors. Neural Network model can also ability to evaluate the assessment of risks in software development and acts as an effective instrument in analysis and minimizing risks that enable continuous improvement in software processes and products.*

***Keywords**: Cost risk, Schedule risk, Performance risk, User risk, Complexity Risk, Artificial Neural Network (ANN).*

I. INTRODUCTION

The feasibility study of the risk management process comprises all the activities required to identify the risk that might have a potential impact on the IT project. To identify the risk involvement, various tools and techniques have been discovered, such as documentation review, information gathering, check list analysis, assumption analysis, diagramming techniques, and so on [9].The success of software and related product are of low margin in reality thus highlighting challenges during development of software products are one the primary reasons for the failure of the software products. The software development projects involve special attention especially when parameters continue changing at different stages of the development cycle. The main primarily parameters of project risks which affects the project process such as performance risk, schedule risk, and cost risk [9,11]. One more parameter user risk also plays a vital role in the calculation of project risk [12]. Taking inspiration from [2,6,9,10,11] the new model is proposed and implemented to show that the proposed model gives 92% of accuracy to evaluate the next immediate project risk.

A. Cost Risk

Cost Risk arises when project cost exceeds the budgeted cost. It is the probability of loss due to cost over-run. Cost risk leads to chain of other risks leads to chain of other risks in a software project. The following example of sides that causes leading to cost risks in a soft project [4]:

- Incorrect budget estimation
- Cost overruns caused by underutilization of resources
- Sudden expansion of project scope
- Poor tracking of finances
- Unfeasible schedule
- Hardware failure
- Lack of testing and monitoring
- Technology change
- Management change
- Malfunctioning of tools

B. Schedule Risk

Schedule management lets in the processes needed to ensure well timed completion of the project. Risk and uncertainties affects project schedule thus causing schedule risk. Schedule risks are the reason for delay in the projects which again may lead to other risks. It is the likelihood of failing to meet scheduled plans. A schedule management risk refers to an overly optimistic task scheduling due to deficit of monitoring metrics and poor usage of historical estimates or records. Poor planning and control often leads to uncontrolled schedules and budgets and a lack of flow in the project development work. The following example of risks that causes scheduling risk in software project [4]:

- Extension and change in requirements
- Insufficient knowledge about tools and techniques
- Technology change
- Improper training to employees
- Human errors
- Slow management cycles
- Environmental changes
- Insufficient budget
- Lack of enough skills, goals specification and tools

Revised Manuscript Received on February 05, 2020.

* Correspondence Author
Malaya Nayak*, University of Derby, Derby, United Kingdom. Email: m.nayak1@unimail.derby.ac.uk
Tariq Abdullah, University of Derby, Derby, United Kingdom. Email: t.abdullah@derby.ac.uk

Retrieval Number: C5974029320/2020©BEIESP
DOI: 10.35940/ijeat.C5974.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

C. Performance Risk

Performance risk is the risk which a project meets when it fails to fulfill the end result to be achieved by it or fails to meet the project requirement that justifies it.

Performance risk leads to schedule risk and cost risk if there is a major technological problem increasing the project's duration or expenses. The inability of the project to give the required output leads to performance risk. A performance management risk is an outcome of a poor quality project execution which again occurs due to multiple factors like error or bugs, time delay and over budgeting issues. The following example of risks that causes performance risk in software project [4]:

- Lack of project standard
- Lack of design documentation
- Unrealistic schedule
- Lack of skill
- Inadequate knowledge about tools
- Technology changes
- Environmental changes
- Extension of requirements change
- Insufficient documents

D. User Risk

Lack of involvement of the user during the development of the project is the most cited risk. If there is no interest of the user towards the project development or no efforts are seen from the users' side, the project may be a failure. This user risk arises out of the users' attitude and behavior during the system development phase of the project. The following example of risks that causes user risk in software project [12,21,22]:

- Lack of users' cooperation
- Users resistance to change
- Users not committed to the project
- Lack of user participation
- Conflict between users
- Users negative attitude
- Users negative perception towards the project
- Lack of users knowledge

E. Complexity Risk

The inherent complexity of a software project, in terms of the difficulty of the project being undertaken, represents another dimension of software project risk. Every project passes through complexities like usage of higher version of technologies, less user-friendly techniques, new system of work which needs training and changed working methods. The following example of risks that causes complexity risk in software project [12, 15,22]:

- Project involves use of new versions of technology not being used earlier
- Large number of links to other systems
- High level of technical complexity
- Immature technology
- High complex task being automated
- One of the largest project undertaken by the organisation

II. REVIEW ON RISK MANAGEMENT APPLICATIONS

The software risks occurring during the various phases of life cycle of development of software have been seen to be top most causes for the less rates of success of application software delivered and hence it is significant to handle the risks prior to them being detrimental. Thus, the management of software risks has preventive measures to evaluate risks before starting off new projects and to increase their success rates. In order to deliver high-quality software products to the customers within the deadline set, it is important to use computer-based tools for risk assessment to help the mangers make a decision. Benefits acquired from a Risk management are essentially of two types: (a) Direct which benefits cater to cost, people and product. (b) Indirect or secondary which benefits cater to process management, optimization and decision making.

A. Analysis of SEI Continuous Risk Management (SEICRM) Model

We studied the SEI Continuous Risk Management (SEICRM) model which is an engineering practice with method, processes, and tools defined to handle various risk factor in a development of software process. The model helps in proactive decision making and creates a disciplined environment; it continuously assesses the risks involved; which risks pose greater threat and find out strategies to deal such risks. The SEICRM model is generally segregated into six sections [3]:

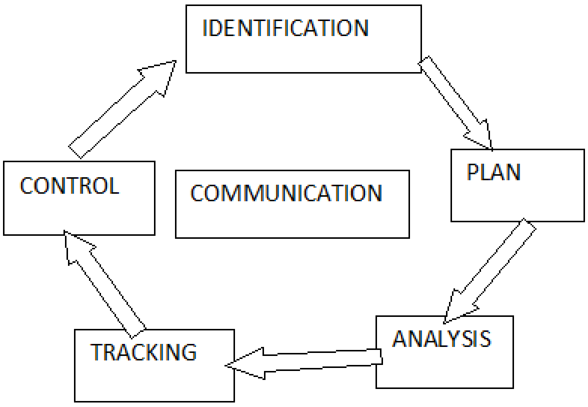


Figure 1: The performance of SEICRM model

1. **Risk Identification Phase:** The phase involves activities to discover potential risk elements before they become a threat to the project delivery. The project team are responsible to identify new and continuous risks during the entire lifecycle of the application software in development.
2. **Risk Plan Phase:** With the prioritized list of risks as input, this phase decides what exactly is to be done with the risks listed and prioritized. Decisions and mitigation strategies are chalked out during this phase entirely based upon the current knowledge about the project risks.

Retrieval Number: C5974029320/2020©BEIESP
DOI: 10.35940/ijeat.C5974.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3. **Risk Analysis Phase:** Purpose of this phase is to convert data elicited on risks and convert them into decision making information. The entire process of risk analysis has 3 steps: evaluation of attributes of a risk by estimating the probability and impact on a scale of 5, ensuring that the attributes of risks are accurate and prioritizing risks.
4. **Risk Tracking Phase:** Risk Tracking is a subjective practical application that defines as collection and documentation of all the different risk factors in a project by the members of team responsible for the job. The indicators of performance are showcases to decision makers in a documentation tracking format to help them analyse and pass on decision for the risk involved.
5. **Risk Control Phase:** Objective of this phase is to check for deviations from the risk mitigation plans. In addition to monitoring the already elicited risks, the team needs to keep track of any new risks that have cropped up as the development phase proceeds. The process is composed of the steps: identifying new risks in project, risk proposal submission, confirmation of risk, assigning resource of the newly identified risk and periodically keeping a check on new project risks.
6. **Communication Phase:** The purpose of this final phase is to communicate about the risk activities of all the current and emerging risks to the management. For effective risk management, continuous and open communication is necessary. Communication may occur both formally or informally by means of presentations or workshops of risk management approaches for the project. All documentations regarding the risks must be well documented and available for access for the management.

III. ARTIFICIAL NEURAL NETWORK (ANN)

Neural network is the brain child of Walter Pitts and Warren McCollum at University of Chicago who were this idea of a neural network in 1994. Later in 1957, The Perceptron was the first trainable neural network designed by Frank Rosenblatt. But it was consist of only one layer. A perceptron can learn by reinforcing weights that lead to correct behavior during training.

Neural networks algorithms are based on the way human brain works. It builds predictive models by learning the patterns in historical data. In a multi-layer perceptron the nodes are organized in layers. The first layer is called the input layer, the outermost layer is named as the output layer. In between these two or more internal layers present, called as hidden layers. The input layer takes the values of the independent variables as input. The nodes of the hidden layer take their inputs from the input layer process it and pass it on to the nodes of the output layer[17]. This way it works within a node. Particularly a node receives information from multiple nodes of the previous layer. These are multiplied with unique weights and added together with a small value called a bias. The total is processed by a function called the activation function and leads the node as output. The process proceeds till information reaches the output layer and leads it as a prediction for the dependent variable. Then the network compares the prediction with the real l value of the dependent variable. In case it doesn't match it adjusts all the weights in the network and repeats the process, these iterations process repeated till the neural network is able to produce accurate predictions for most of the observations once this is achieved, and we are left with a neural network model that can be applied to a new set of data for predictions.

Neural network is a kind of black box which takes one or multiple inputs processing them into one of multiple outputs. The neural network itself consists of many small units called neurons. These neurons are grouped into several layers. Units of one layer interpose of the neurons of the future layer through weighted connections, which actually adjust connections with a real-valued number affiliated to them.

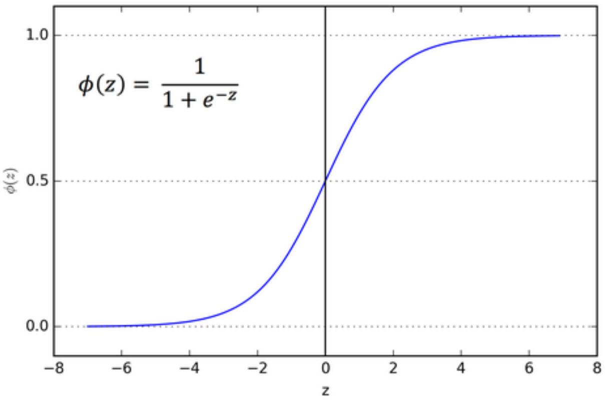
Neuron takes value of a connected neuron and it multiplies with their connections weight. Sum of all connected neurons and, the neurons bias value is put into the activation function, which elementary mathematically transforms the value before it finally be passed on to the next neuron. This manner the inputs are propagated through the whole network that's pretty much all the network does but the real deal behind neural networks is to find the correct weights in order to get the correct results. This is done through a wide range of techniques such as backpropagation.

Activation Function

Activation functions are an extremely important feature of the ANN. They basically decide whether a neuron should be activated or not. It actually limits the output signal to a finite value. Bias is the information which can impact output without being dependent on any feature. The activation function does the non-linear transformation to the input, making it capable to learn and perform more complex relationship. Without an activation function, the neural network is just a linear regression model[18, 19].

We can represent any kind of function with neural network. Hence neural network consider as UNIVERSAL FUNCTION APPROXIMATORS, means they can compute any function or any process. The main purpose of activation function is to introduce non-linearity in the network so it would be capable of learning more complex pattern. There are different types of activation functions named as: Sigmoid, TanH (Hyperbolic Tangent Activation Function), ReLu (rectified linear unit), Softmax. We have used Sigmoid Activation function [8,19,20] in our stock market price prediction.

Sigmoid Activation Function



The main reason why we use sigmoid activation function because it exists between (0 to 1) which help us to apply on the Gaussian data set.

Therefore, it specially used for models in which, we have to predict the probability as output. Because probability exists only in the range of 0 and 1, sigmoid is the proper choice.

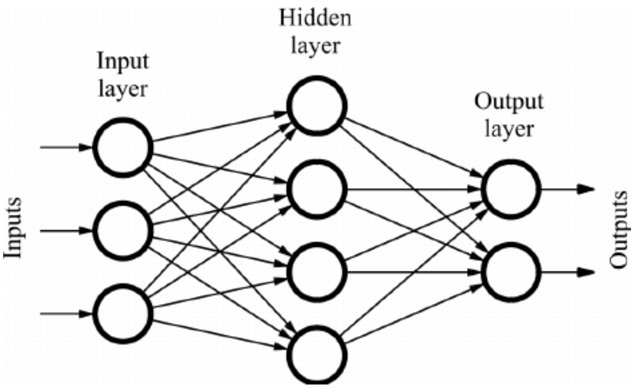
Artificial Neuron

A neural network is an interesting area of machine learning. It's simultaneously one of the oldest and one of the newest areas. A neural network is made up of artificial neurons[16]. One artificial neurons has many inputs. Each input has an associated weight. All inputs are summed giving the 'activation'. If the activation is bigger than a threshold the neuron outputs a signal. Neural network is used model complex relationships between inputs and outputs. The neural network is built in layers where nodes receive input from the layer below and send input to the layer above. Node will do a computation with the values (weights) it receives and if it reaches a threshold it will send the result value to layer above. At the start of training the weights are randomly assigned.

Activation Function

The activation is not necessary linear. An activation function is typically used to translate the activation into a threshold value.

Activation function are needed so we can break the linearity and represent more complicated relationships. Moreover, activation functions are required to in order to stack the layers. Activation function transforms inputs into outputs of a different kind.



In the following diagram the output of every neuron in the input layer feed into every neuron in the next hidden layer. The output of every neuron from the hidden layer is feed into the input of every neuron in the output layer.

Forward Propagation

Forward propagation is the process of pushing inputs through the net. At the end of each epoch, the obtained outputs are compared to targets to form the errors

Learning/Training

In order to learn there are various algorithms that can be used. Simply, it boils down to finding the right weights for the different inputs of the neurons.

Backpropagation

Backpropagation of errors is an **algorithm** for neural networks using gradient descent. It consists of calculating the contribution of each **parameter** to the errors. We back propagate the **errors** through the net and **update** the parameters (weights and biases) accordingly.

A. Forecasting With Neural Network: Advantages & Weaknesses

To summarize neural nets are data driven prediction algorithms that can be used for forecasting. They can be automated so that the software searches over a range of networks and chooses the best one or it averages across many networks. Neural nets can be used to forecast numerical series as well as binary series.

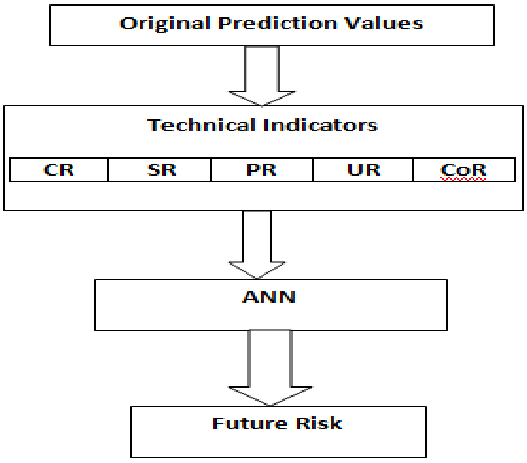
In terms of pre-processing there are different ways to setup a neural network and there is a disagreement on what's the best pre-processing approach. So you have to try different setups in any case it's useful to detect and address outliers and unusual periods before you run the neural net. Because of its data-driven nature and because it fits a very complex relationship, the neural net requires a sufficiently large number of observations in the training period. That's why it's typically used in high frequency applications such as trading and energy usage.

Neural nets are considered a black box because unlike regression we don't get coefficients that quantify the effect of different components such as seasonality and trend. However in forecasting that's really not a serious problem and it's not our main goal. Performance evaluation is extremely important with neural nets because they fit a complex relationship. They're easy to over fit. It's always good to compare the training and validation performance to make sure that you didn't over fit to your data. Finally neural nets are heavier to run in some applications this is acceptable while in others it really isn't.

B. Proposed Forecasting ANN Model for Risk Management

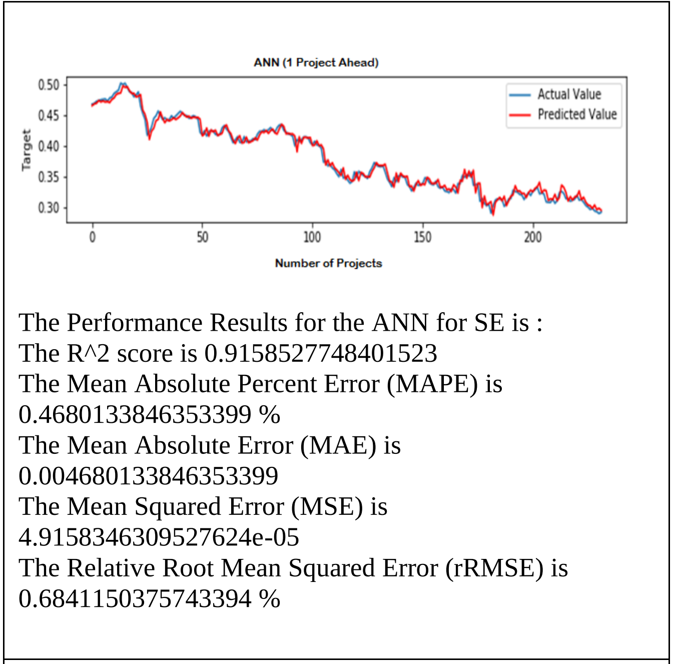
This section we briefly explain the procedure and architecture with respect to ANN and the **Technical indicators.**

The Model

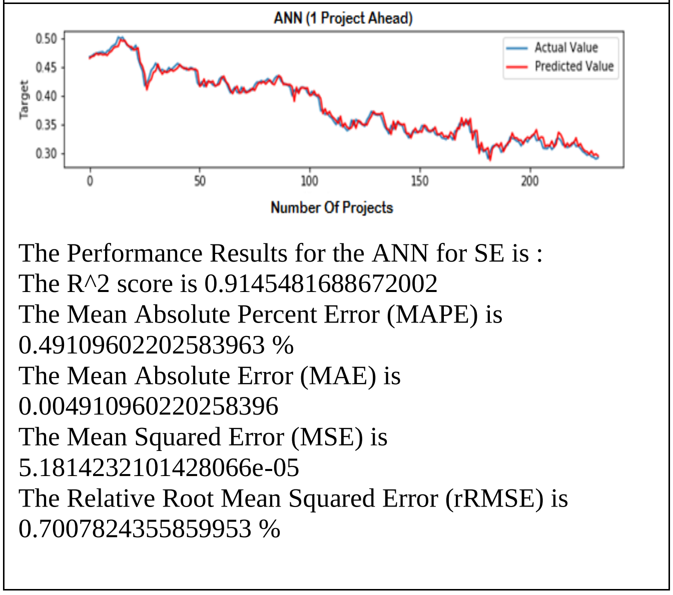


We have consider technical indicators are CR(Cost risk**), SR(**Schedule risk**), PR(**Performance risk**), UR(**User risk**) and CoR (**Complexity Risk**).**

In the process we have created data with giving random values to different indicators and feed to neural network as input and also targeted to random risk values. In the very first step we created random data for the five technical indicators. At the same time we created random risk values associated with the technical indicator respectively. All random values were taken in between 0 and 1. Where zero stands for highly risk project and 1 stands for with zero causality in the project.



```
The Performance Results for the ANN for SE is :
The R^2 score is 0.9158527748401523
The Mean Absolute Percent Error (MAPE) is
0.4680133846353399 %
The Mean Absolute Error (MAE) is
0.004680133846353399
The Mean Squared Error (MSE) is
4.9158346309527624e-05
The Relative Root Mean Squared Error (rRMSE) is
0.6841150375743394 %
```



```
The Performance Results for the ANN for SE is :
The R^2 score is 0.9145481688672002
The Mean Absolute Percent Error (MAPE) is
0.49109602202583963 %
The Mean Absolute Error (MAE) is
0.004910960220258396
The Mean Squared Error (MSE) is
5.1814232101428066e-05
The Relative Root Mean Squared Error (rRMSE) is
0.7007824355859953 %
```

In our experiment for each data set we ran the ANN algorithm for ten times and took the average for prediction (see Table I).

Table– I: Result.

No. of Project	R^2	MAPE	MAE	MSE	rRMSE
250	0.916	0.468	0.005	4.916	0.684
250	0.9155	0.491	0.0059	5.1811	0.701

Here in this work we feed 5 technical variables to ANN algorithm with two intermediate Layer targeting the single out put risk involved in the project. Many simulations we tested and presented two of them and show that our algorithm predicts 92 % accuracy with a multiple run. The mathematical formulas used to calculate the values in the ANN as follows.

The determination $R^2(y,\hat{y}) = 1-\frac{\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}_i\right)^2}$

The Mean absolute error $MAE = \frac{\sum_{i=1}^{n}|y_i-x_i|}{n}$

The Mean Squared error

$$MSE(y,\hat{y}) = \frac{1}{n_{sample}}\sum_{i=0}^{n_{sample}-1}(y_i-\hat{y}_i)^2$$

IV. CONCLUSION

In this paper, ANN algorithm specially designed to run on technical variables in the purpose of predicting risk involved to undertake a project. Integration of ANN in the field of risk management is the new idea and methodology. It shows better percentage rate of risk prediction compare to other methods so far this this domain. It opens for future work with respect to other Machine learning algorithms such as Support Sector Machine (SVM), Genetic Algorithm, Independent Component Analysis (ICA) and others. Experiment may be conducted to see its practical application on Genetic Algorithms. One can extent this work for long term prediction with rigorous experiment on benchmark datasets.

REFERENCES

1. B. W. Boehm, “Software risk management: principles and practices,” *IEEE software*, vol. 8, no. 1, 1991, pp. 32-41.
2. A. Cochocki, R. Unbehauen, “Neural networks for optimization and signal processing,” John Wiley & Sons, Inc., 1993.
3. A. J. Dorofee, J. A. Walker, C. J. Alberts, R. P. Higuera, R. L Murphy. “Continuous Risk Management Guidebook,” CARNEGIE-MELLON UNIV PITTSBURGH PA, 1996.
4. H. Hoodat, H. Rashidi, “Classification and analysis of risks in software engineering,” *World Academy of Science, Engineering and Technology*, vol. 56, no 32, 2009, pp. 446-452.
5. K. J. Kim, I. Han, “Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index,” Expert *systems with Applications*, vol. 19, no. 2, 2000, pp. 125-132.
6. B. Kröse, P. van der Smagt, *An introduction to neural networks*. Amsterdam: University of Amsterdam, 1997.
7. T. G. Kirner, L. E. Gonçalves, “Software Risk Management: a Process Model and a Tool,” in *Software Engineering Techniques: Design for Quality*. Springer, Boston, MA, 2006, pp. 149-154.
8. T. M. Mitchell, “Machine learning. 1997,” in *Burr Ridge*, vol. 45, no. 37, IL: McGraw Hill, 1997, pp. 870-877.
9. M. K. Nayak, S. Mohanty, “Schedule Risk Analysis of ICT Infrastructure Projects,” *International Journal of Computer Applications (IJCA)*, vol. 38, no. 5, 2012, pp. 1-5.
10. L. Wallace, M. Keil, A. Rai, “How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model,” *Decision sciences*, vol. 35, no. 2, 2004, 289-321.
11. M. K. Nayak, S. Mohanty, “Optimization Model in Human Resource Management for Job Allocation in ICT Project”, *International Journal of the Computer, the Internet and Management (IJCIM)*, vol. 19, no. 3, September-December 2011, pp. 21 -27.

12. M. K. Nayak, S. Mohanty, Evaluation of Frameworks for Risk Scheduling and Requirement Volatility with Quality Standards in ICT Project", *International Journal of Computer Applications (IJCA)*, vol. 29, no. 2, September 2011, pp. 1-6.
13. M. K. Nayak, S. Mohanty, "Predictive Analytical Model for Requirement Volatility in ICT Projects", *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 9, no. 8, August 2011, pp. 96-104.
14. M. K. Nayak, S. Mohanty, R. Soni, "People and Skills: A Potential Risk in ICT Infrastructure Management", *International Journal of Software Engineering and Its Applications (IJSEIA)*, vol. 5, no. 3, July 2011, pp. 135-142.
15. S. Liu, J. Zhang, M. Keil and T. Chen, "Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study", 2010.
16. H. El-Amir and M. Hamdy, "Deep Learning Pipeline | SpringerLink", Link.springer.com, 2020. [Online]. Available: https://link.springer.com/book/10.1007%2F978-1-4842-5349-6. [Accessed: 01- Feb- 2020].
17. T. Abar, A. Ben Letaifa and S. El Asmi, "Machine learning based QoE prediction in SDN networks - IEEE Conference Publication", Ieeexplore.ieee.org, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7986488. [Accessed: 10- Dec- 2019].
18. D. Gupta, "Fundamentals of Deep Learning - Activation Functions and their use", Analytics Vidhya, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/. [Accessed: 02- Feb- 2020].
19. N. Omkar, "Activation Functions with Derivative and Python code: Sigmoid vs Tanh Vs Relu", Medium, 2019. [Online]. Available: https://medium.com/@omkar.nallagoni/activation-functions-with-derivative-and-python-code-sigmoid-vs-tanh-vs-relu-44d23915c1f4. [Accessed: 18- Dec- 2019].
20. Vidisha De, T. T. Teo, W. L. Woo, T. Logenthiran,"Photovoltaic Power Forecasting using LSTM on Limited Dataset - IEEE Conference Publication", Ieeexplore.ieee.org, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8467934. [Accessed: 20- Dec- 2019].
21. S. Mittal,"Risk Analysis and Mitigation Steps in Different Phases of Software Development", B.K. Birla Institute of Engineering & Technology, vol. 2, no. 6, 2013, pp. 241-243.
22. L. Marziana Abdullah and J. M. Verner, "Outsourced strategic IT systems development risk - IEEE Conference Publication", Ieeexplore.ieee.org, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/5089291?cv=1. [Accessed: 01- Feb- 2020].

AUTHORS PROFILE

Malaya Nayak is currently studying Computing and IT at the University of Derby UK. His field of interests includes software engineering, security, project management and risk management.

Tariq Abdullah is an Academic Lead- Computing & IT and a research fellow at the University of Derby UK. He completed his PhD from Delft University of Technology (TUDelft). He has vast experience of leading the design, development and implementation of R&D projects from both industry and academia. His research interests include video analytics, big data analysis, machine learning and distributed systems.