iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 Issue: III Month of publication: March 2015

DOI:

www.ijraset.com Volume 3 Issue III, March 2015
IC Value: 13.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Neural Network Approaches for Software Development Time Estimation: A Review

Vidisha Agrawal[1], Vishal Shrivastava[2]
Department of Computer Science and Engineering, ARYA college of Engineering and IT, Jaipur, Rajasthan, India

Abstract: Software effort estimation calculates the effort necessary to complete the project, in term of scheduling, acquiring resources, and meeting costs. The aims of researchers have been: first is to determine which technique has the greatest effort prediction accuracy and secondly to propose new or combined techniques that could provide better estimates. Most of the research has focused on the construction of formal models and the early models were typically statistical regression-based. This paper presents an Adaptive Neuro-Fuzzy Approach for Software Development Time Estimation. This proposed technique is aimed at building and evaluating a Neuro - fuzzy model for software project development time. The forty one modules were used as a data set. Our proposed approach is software development time estimation method show the proposed ANFIS model gives new approach and ideas as compared to different types of neural network models. In the proposed method accurate estimation of software development time will be done and the results of Neuro Fuzzy approach will be compared with different types of neural network models based upon various parameters such as Root Mean Squared Error (RMSE), Relative Standard Deviation (RSD), Magnitude of Relative Error (MRE), Mean Magnitude of Relative Error (MMRE), Balanced Relative Error (BRE) and Prediction (Pred).

Keywords: Soft computing, Adaptive, Neuro-Fuzzy Inference System (ANFIS), RMSE MRE, BRE, RSD, MMRE, BRE, Prediction.

I. INTRODUCTION

Software Engineering (SE) [1] is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software and it provides the fundamentals, principles and skills needed to develop and maintain high quality software products. Some of the areas of SE are: Requirement, design, construction, testing, and management.

Software Engineering Management includes planning and measurement of SE; in this context, the topic software development effort estimation (SDEE) [2] is presented. Development effort estimates [14] are required by project managers for planning and for controlling the process of software development projects. So, Software projects development can be considered as the most uncertain and complex project when compared to other types of engineering projects, because their activities involve an intangible product and these are continually changing in response to customers' requirements and the development of new technology. Based on the results of several investigations of software development projects, the main areas responsible for project failure were found to be as follows: project goal setting, project scheduling, project staffing (availability and capabilities), customer requirements, unmanaged risks, improper project execution, stakeholder politics, and commercial pressures. The success of a software project will be highly dependent on the Project Planning Phase, which involves those activities that determine a project's scope, scheduling, cost, resources, and risks. Therefore, while this phase is critical in all cases of project management, so it is a special phase in a software development project, but the main activities in the software project planning phase are effort estimation and risk management. These two activities together with quality estimation become the major issues in the success of software development project and the accuracy of the results will provide the great support in project execution phase.

Software development estimation techniques can be classified into three general categories:

A. *Expert judgment*

The term expert estimation [3] is not clearly defined and covers a wide range of estimation approaches; a common characteristic is, however, that intuitive processes constitute a determinant part of the estimation. This technique implies a lack of analytical argumentation and by the frequent use of phrases such as "I think that . . ." and "I feel that . . ." and it aims to derive estimates based on an experience of experts on similar projects. The means of deriving an estimate are not explicit and therefore not repeatable.

B. Algorithmic models

To date is the most popular technique [4] in the literature. It attempts to represent the relationship between effort and one or more characteristic of a project; the main cost driver in a model is usually taken to be some notion of software size (e.g. the number of lines of source code). Its general form is a linear regression equation, or by a group of non-linear regression equations as those used by Boehm.

C. Machine learning

In recent years machine learning techniques have been used as a complement or alternative to the previous two techniques. Fuzzy logic models are included in this category as well as neural networks, genetic programming, regression trees and case-based reasoning. Software cost and schedule estimation supports the planning and tracking of software projects. Effectively controlling the expensive investment of software development is of paramount importance.

II. ARTIFICIAL NEURAL NETWORK (ANN)

The feed forward multi-layer network with back propagation learning is the most commonly used structure in the field of software cost estimation [16]. The network contains neurons arranged in layers with each neuron is connected to every neuron of the lower layer forming a complete graph. The cost drivers or project attributes are fed as inputs at the input layer which propagates across subsequent layers of processing elements known as neurons and generates effort estimation in terms of Person-Months (PM) at the output layer. ANN [5] follows a two-step process. In step 1 threefold validation is employed for the training of the non-linear adjustment (ANN) [28]. This is followed by predicting stage in step 2. At this stage, a new project is presented to the trained system. The training process of an ANN [17] is a non-linear and non-constrained optimization problem, where a search takes place for a minimum of the error function between the network output and the desired output. This cost function traditionally is the mean square error (MSE).

A. Fuzzy Logic

The three main steps to apply fuzzy logic for effort prediction are:
Step 1: Fuzzification: It converts crisp input to fuzzy output.
Step 2: Fuzzy Rule Based System: Fuzzy logic systems use fuzzy IFTHEN rules. Once all crisp input values are fuzzified into their respective Linguistic values, the fuzzy inference engine accesses the fuzzy rule base to derive.
Step 3: Defuzzification: It converts fuzzy output into crisp output.
An adaptive software effort estimation model incorporating different fuzzy logic system is developed to handle imprecision and uncertainty in software attributes of COCOMO-II model. Ahmed’s Type-2 Fuzzy logic System (FLS) which evaluates the performance of a prediction system developed using the framework for handling imprecision and uncertainty when size is provided as a precise but uncertain input is an another example of fuzzy system software cost estimation. The prediction system consists of two stages: nominal effort prediction and EAF (Effort Adjustment Factor) prediction. The outputs of both the stages are merged (multiplied) to produce the actual effort.

B. Adaptive Neuro Fuzzy Approach

Neuro-fuzzy was proposed by J. S. R. Jang. Adaptive Neuro fuzzy [6] is a kind of neural network that is based on Takagi–Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions. The Neuro-fuzzy [12] in fuzzy modeling research field is divided into two areas: linguistic fuzzy modeling that is focused on interpretability, mainly the Mamdani model; and precise fuzzy modeling that is focused on accuracy, mainly the Takagi-Sugeno-Kang (TSK) model.

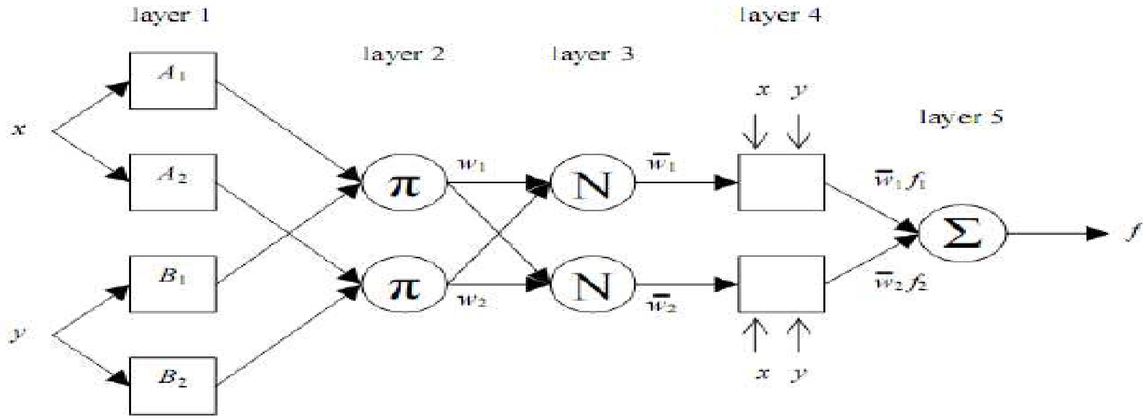


Figure1: ANFIS Architecture

The proposed method estimates the software development time accurately by proposed Adaptive Neuro Fuzzy Inference System (ANFIS) as it is a combination of Fuzzy Logic and Neural Network so ANFIS takes advantages from fuzzy logic and neural network. This ANFIS constructs a Fuzzy inference system by using given training data set whose membership function parameters are adjusted by back propagation algorithm or in combination with least square type of method. Figure.2 shows a high level diagram of the proposed ANFIS. Inputs and their membership functions appear to the left of the ANFIS structural characteristics, while outputs and their membership functions appear on the right.

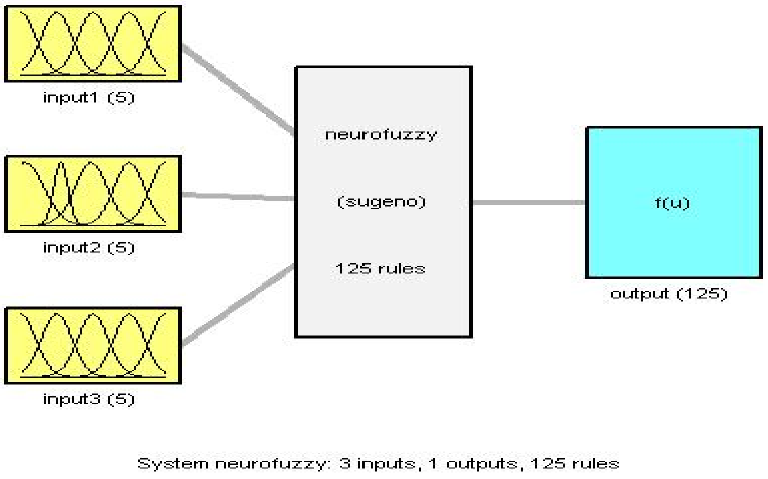


Figure 2: Diagram of proposed ANFIS

III. LITERATURE SURVEY

In this paper, I have made a review on my topic Adaptive Neuro Fuzzy model for software time estimation by reading different kinds of papers and analyzing different techniques which are being used in these papers published by authors which are discussed as follows:

Nassif et al. [7] “Towards an early software estimation using log-linear regression and a multilayer perceptron model” (2013) have proposed a novel log-linear regression model based on the use case point model (UCP) to calculate the software effort based on use case diagrams. A fuzzy logic approach is used to calibrate the productivity factor in the regression model. Moreover, a multilayer perceptron (MLP) neural network model was developed to predict software effort based on the software size and team productivity.

The proposed approach outperforms the original UCP model. Furthermore, a comparison between the MLP and log-linear regression models was conducted based on the size of the projects. Results demonstrate that the MLP model can surpass the regression model when small projects are used, but the log-linear regression model gives better results when estimating larger projects.

Rama Sree P,Prasad Reddy,Sudha K.R [8] "Hybrid Neuro-Fuzzy Systems for Software Development Effort Estimation" (2012) this paper represent is an attempt to experiment different types of Neuro-Fuzzy Models. Two case studies are used for this purpose. The first is based on NASA-93 dataset and the other is based on Maxwell-62 dataset. They have analyzed case studies using six different criterions like Variance Accounted For (VAF), Mean Absolute Relative Error (MARE), Variance Absolute Relative Error (VARE), Mean Balance Relative Error (Mean BRE), Mean Magnitude Relative Error (MMRE) and Prediction.

Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra [9], "Software Development Effort and Cost Estimation: Neuro-Fuzzy Model" (2012) discussed Neuro-Fuzzy model. Neuro-Fuzzy models are the combination of Artificial Neural Network and Fuzzy Logic. Artificial Neural Network has the ability to learn from previous data. It model complex relationships between both independent variables (cost drivers) and dependent variables (effort). Fuzzy logic simulates the human behavior and reasoning. Fuzzy logic is basically used in situation where decision making is very difficult and conditions are not clearly defined. Facts that may be dismissed are focused in this technique.

Urvashi Rahul Saxena, S.P.Singh [10], "Software Effort Estimation Using Neuro-Fuzzy Approach" (2012) in this paper they explore Neuro-fuzzy techniques to design a suitable model to utilize improved estimation of software effort for NASA software projects. Comparative Analysis between Neuro-fuzzy model and the traditional software model(s) such as Halstead, Walston-Felix, Bailey-Basili and Doty models is provided. The evaluation criteria are based upon MMRE (Mean Magnitude of Relative Error) and RMSE (Root mean Square Error). Integration of neural networks, fuzzy logic and algorithmic models into one scheme has resulted in providing robustness to imprecise and uncertain inputs.

Vachik S. Dave, Kamlesh Dutta [11], "Comparison of Regression model, Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort Estimation" (2011) they have compared Neural Network models and regression model for software Development effort estimation. The comparison reveals that the Neural Network (NN) is better for effort prediction compared to regression analysis model. Further, we have compared two Neural Network models- Feed-Forward Neural Network (FFNN) and Radial Basis Neural Network (RBNN). The evaluation of the models is based on Mean Magnitude Relative Error (MMRE).

IV. PROPOSED METHOD

The proposed ANFIS model [12] is a first-order Sugeno type fuzzy inference system with 3-inputs and 1-output. Each input has generalized bell type membership functions and the output has a constant membership function. The standard dataset as proposed by Lopez-Martin et.al. has been used for the experimentation purposes. They used the sets of system development projects, where the Development Time (DT), Dhama Coupling (DC), McCabe Complexity (MC) and the Lines of Code (LOC) metrices were registered for 41 modules. Since all the programs were written in Pascal, the module categories mostly belong to procedures and functions. The development time of each of the forty-one modules were registered including five phases: requirements understanding, algorithm design, coding, compiling and testing. To estimate the time accurately, first we select the modules from the Lopez Martin data set which have least variations and then train our ANFIS model with the training data as shown in Table 1.

Table1: Training Data

INPUTS			OUTPUT
MC	DC	LOC	
2	0.083	10	15
2	0.125	9	15
2	0.125	9	16
2	0.125	14	16
2	0.167	7	16

2	0.167	8	18
2	0.167	10	15
2	0.167	10	15
2	0.167	10	18
2	0.2	10	14
2	0.2	10	15
3	0.083	17	22
3	0.125	11	19
3	0.125	15	18
3	0.125	15	19
3	0.143	13	21
3	0.143	14	20
3	0.143	14	21
3	0.143	15	19
3	0.143	15	20
3	0.167	13	15
3	0.167	14	13
3	0.2	18	19
5	0.143	22	24.5
5	0.143	22	24.5
4	0.077	16	21
4	0.077	31	21

Therefore, the number of rows of the training data is equal to the number of training data pairs, and, since there is only one output, the number of columns training data is equal to the number of inputs plus one.

A summary of several important soft computing based effort estimation models along with their approach is as shown in table 2.

Table 2

Soft computing based effort estimation models

Authors	**Title & Journal**	**Year**	**Technique**	**Data Set**	**Metrics**
C.J. Burgess, M. Lefley	Can genetic programming improve software effort estimation? A comparative evaluation *Information and Software Technology*	2001	Artificial Neural Network	Desharnais	MMRE, BMMRE, AMSE,Pred$_{25}$
Kirsopp,C. and Shepperd,M.J	Making Inferences With Small Numbers Of Training Sets *6th International Conference on Empirical Assessment & Evaluation in Software Engineering*	2002	Case Based Reasoning	Desharnais	MMRE

Xishi Huang, Luiz F. Capretz, Jing Ren, Danny Ho	A Neuro-Fuzzy Model for Software Cost Estimation *Third International Conference On Quality Software*	2003	Neuro-Fuzzy	Cocomo81	RE PERC IMPR
Harsh Kumar Verma, Vishal Sharma	Handling Imprecision in Inputs using Fuzzy Logic to Predict Effort in Software Development *IEEE 2nd International Advance Computing Conference*	2010	Fuzzy Logic	Cocomo81	$Pred_{25}$ MMRE
Iman Attarzadeh, Siew Hock Ow	Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model *IEEE International Conference on Fuzzy Systems*	2011	Fuzzy Logic	Cocomo81 NASA	MMRE $Pred_{25}$
Prasad Reddy P.V.G.D , Sudha K.R,Rama Sree P	Application Of Fuzzy Logic Approach To Software Effort Estimation *International Journal Of Advanced Computer Science And Applications*	2011	Fuzzy Logic	NASA	VAF,MARE VARE,MEAN BRE,MMRE $PRED_{30}$

V. CONCLUSION

To increase the performance of software effort estimation models under the quickly changing computing environment. If the dataset has more linear data points, it is easier to estimate. The Neuro-Fuzzy Model which uses more number of rules would give correct results as practically all the combinations of inputs are being represented as rules for the fuzzy inference system. Software Development Time Estimation results show that the proposed ANFIS model gives outstanding results as compared to different types of neural network models. This model is suitable for various areas of software development such as Effort estimation, Cost Estimation.

REFERENCES

[1] S.Basha and P.Dhavachelvan, “Analysis of Empirical Software Effort Estimation Models” in International Journal of Computer Science and Information Security (IJCSIS), Vol. 7, 2010.

[2] B.Hughes and M.Cotterell, “Software Project Management”, Tata McGraw-Hill, 2006.

[3] T.Gruschke, “Empirical Studies of Software Cost Estimation: Training of EffortEstimation Uncertainty Assessment Skills”, 11th IEEE International Software Metrics Symposium, IEEE, 2005.

[4] Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee, “Comparing Some Neural Network Models for Software Development Effort Prediction”, IEEE conference , 2011.

[5] J. Jantzen, “Neuro- Fuzzy modeling”, Reort no 98-H-874, 1998.

[6] Xishi Huang et al., “A Neuro-Fuzzy Model for Software Cost Estimation”, Proceedings of the Third International Conference on Quality Software (QSIC’03), IEEE, 2003.

[7] Nassif Bou Ali, Ho Danny, Capretz Fernando Luiz, “Towards An Early Software Estimation Using Log-Linear Regression and A Multilayer Perceptron Model”, 2013, pp. 144-160.

[8] Rama Sree P,Prasad Reddy and Sudha K.R., “Hybrid Neuro-Fuzzy Systems for Software Development Effort Estimation”, International Journal on Computer Science and Engineering, Vol. 4, pp.1924-1932, Dec 2012.

[9] Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra[29], “Software Development Effort and Cost Estimation: Neuro-Fuzzy Model”, IOSR Journal of Computer

Engineering (IOSRJCE), Vol. 2, pp. 12-14, July-Aug. 2012.

[10] Urvashi Rahul Saxena, S.P.Singh, “Software Effort Estimation Using Neuro-Fuzzy Approach”, CSI 6th International Conference, pp.1-6, Sept.2012.

[11] Vachik S. Dave, Kamlesh Dutta, “Neural Network based Software Effort Estimation & Evaluation criterion MMRE”, International Conference on Computer & Communication Technology (ICCCT), pp.347-351, 2011.

[12] N. Sharma1, A. Bajpai and M. R. Litoriya, “A Comparison of Software Cost Estimation Methods: A Survey”, The International Journal of Computer Science and Applications (TIJCSA), Vol.1, no. 3, ISSN – 2278 – 1080, May 2012.

[13] T. R. Benala, S. Dehuri and R. Mall, “Computational Intelligence in Software Cost Estimation: An Emerging Paradigm”, ACM SIGSOFT Software Engineering Notes Page, Vol. 37, no.3, 2012.

[14] J.S. Pahariya, V. Ravi and M. Carr, “Software Cost Estimation using Computational Intelligence Techniques”, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) IEEE, 2009.

[15] C.J. Burgess, M. Lefley, ” Can genetic programming improve software effort estimation? A comparative evaluation”, Information and Software Technology, pp.863–873, 2001.

[16] Anish Mittal, Kamal Parkash and Harish Mittal “Software Cost Estimation Using Fuzzy Logic”, ACM SIGSOFT Software Engineering Notes Page , Vol 35 no. 1, November 2010.

[17] Harsh Kumar Verma, Vishal Sharma, “Handling Imprecision in Inputs using Fuzzy Logicto Predict Effort in Software Development” , IEEE conference, 2010.

[18] G. D. Boetticher, “An assessment of metric contribution in the construction of a neural network-based effort estimator”, Proceedings of Second International Workshop on Soft Computing Applied to Software Engineering, 2001.

[19] A.R.Venkatachalam, “Software cost estimation using artificial neural networks”, in: Proceedings of the International Joint Conference on Neural Networks, pp. 987–990, 1993.

[20] J. Ryder, “Fuzzy modeling of software effort prediction”, Proceedings of IEEE Information Technology Conference, Syracuse, NY, 1998.

[21] Allidri et al., “COCOMO Cost Model Using Fuzzy Logic”, 7th International Conference On RFuzzy Theory & Technology Atlantic City, New Jersey, February27-March3, 2000.

[22] Heejum Park and Seung Baek, “An Empirical validation of a neural network model for software effort estimation”, Expert System with Application,Elsevier, pp.929-937, 2007.

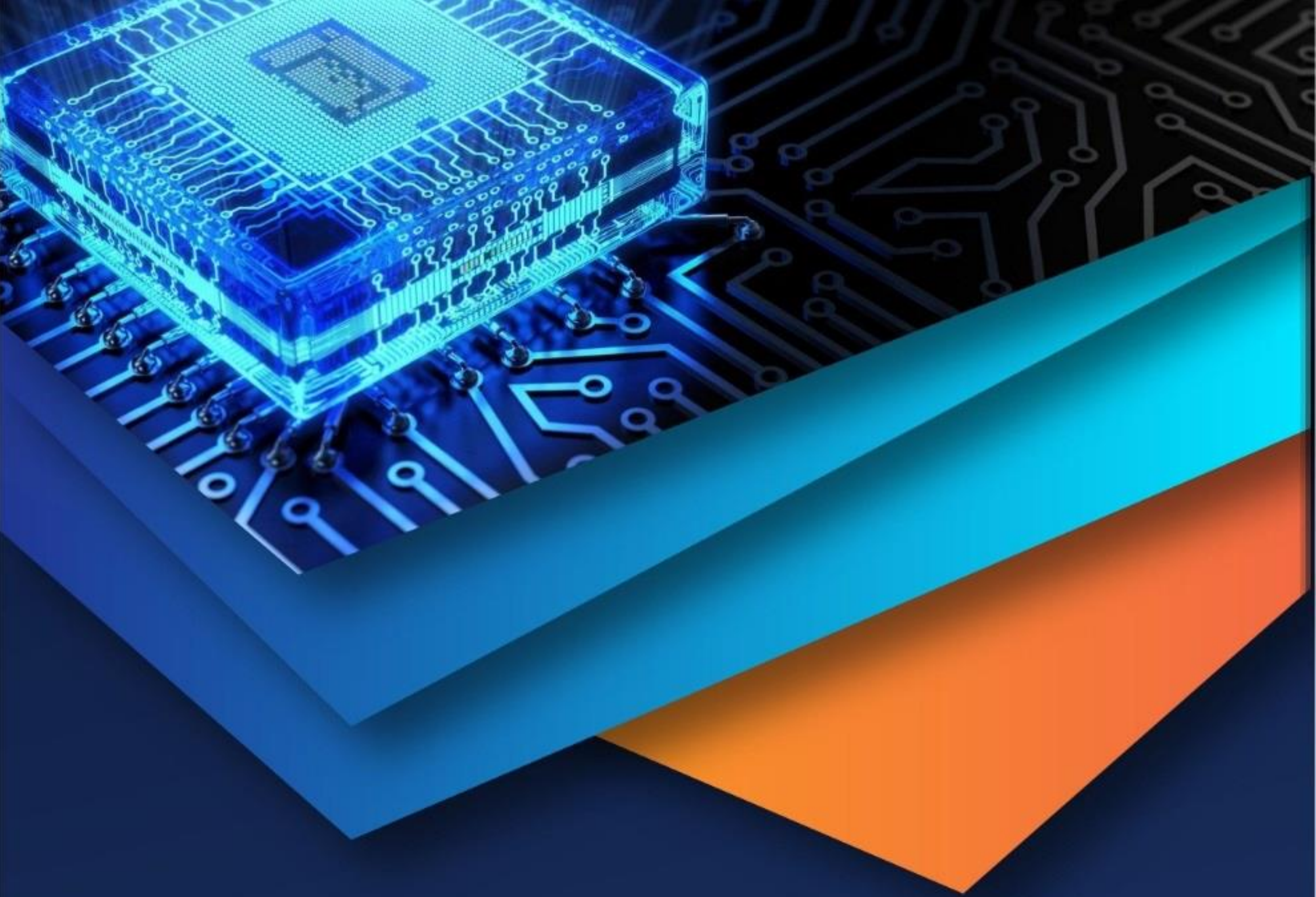
[23] Dinesh C.S. Bisht and Ashok Jangid, “Discharge Modelling using Adaptive Neuro-Fuzzy Inference System”, International Journal of Advanced Science and Technology,Vol. 31,pp.99-114, June 2011.

[24] Mrinal Kanti Ghose, Roheet Bhatnagar and Vandana Bhattacharjee, “Software Development Effort Estimatiom- Neural Network Vs. Regression Modelling Approach”, International Journal of Engineering and Technology, Vol. 2(7), pp. 2950-2956, 2010.

[25] Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra, “Software Development Effort and Cost Estimation: Neuro-Fuzzy Model”, IOSR Journal of Computer Engineering (IOSRJCE), Vol. 2, pp. 12-14, July-Aug. 2012.

[26] M. Chemuturi, “Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimator” , 2009.

[27] Vachik S. Dave, Kamlesh Dutta, “Comparison of Regression model, Feed-forward Neural Network and Radial Basis Neural Network for Software Development Effort Estimation”, ACM SIGSOFT Software Engineering, Vol. 36, No.5, Sept. 2011.



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY