Int. J. Advanced Networking and Applications 1990
Volume: 05, Issue: 04, Pages:1990-1995 (2014) ISSN : 0975-0290

Neural Network based Software Effort Estimation: A Survey

Muhammad Waseem Khan
Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt
Email: Muhammad.wasim1@gmail.com
Imran Qureshi
Department of Computer Sciences, COMSATS Institute of Information Technology, Wah Cantt
Email: imarwat11@gmail.com

ABSTRACT

Software effort estimation is used to estimate how many resources and how many hours are required to develop a software project. The accurate and reliable prediction is the key to success of a project. There are numerous mechanisms in software effort estimation but accurate prediction is still a challenge for the researchers and software project managers. In this paper, the use of Neural Network techniques for Software Effort Estimation is discussed and evaluate on the basis of MMRE and Predicate. At the end, a specific Neural Network based Effort Estimation technique is proposed for further research.

Keywords - Effort Estimation, Fuzzy Logic, Genetic Programming, Particle Swarm Optimization, MMRE.

Date of Submission: February 06, 2014 Date of Acceptance: March 15, 2014

I. INTRODUCTION

One of the main challenges for software industry is to predict software development effort accurately. Software effort estimation is the part of software development life cycle for finding the software effort to estimate software costs required [1]. Many parameters, i.e., time, effort, and personnel used for the software development can be estimated using Effort estimation [2]. Software estimation consists of three stages, i.e., 1st stage is size estimation, 2nd stage is effort estimation, and time estimation, while 3rd stage consists of cost estimation and staffing estimation. Figure 1 shows the interaction among these stages.

II. MODELS USED FOR EFFORT ESTIMATION

Several techniques and methods have been proposed to improve the correctness of software development effort estimation. Currently used Effort estimation models are defined as under:

Fuzzy Logic

Fuzzy logic is used to solve difficult and compound problems. It can minimize the complexity of problem [3].

Linear Regression Trees

Linear Regression Trees (LRT) used independent variable to find the best fit. It helps in understanding that how dependent variable change its value as a result of change in independent variable [4].

Genetic Programming

The technique of GA was developed for handling general optimization problems with large search space [5]. They can generate solutions with optimal results [6].

Analogy Based Effort Estimation

It is used to make relationship and find similarity between different projects in order to find accurate results [7].

Soft Computing

It uses project features, built-in knowledge, and expert knowledge to get accurate prediction results[8].

Expert Judgment

An expert of software development processes estimates software development parameters. The accuracy of estimate is depend on project matches and similarity among projects [9],[10].

Data Mining Techniques

Data mining technique is used to transform large and complex data into meaningful patterns and rules [11]. Some basic operations of Data Mining are Regression and Classification [12].

Case Based Reasoning

CBR uses previous knowledge, similar past case, and reuse them to solve a new problem using CBR[13].

Algorithmic Estimation

It contains mathematical formula to relate independent variables (such as cost drivers) to dependent variable (such as effort, cost). One well known model for estimation is COCOMO model developed by Barry Boehm [14].

Rule Induction

The process of producing a set of rules is done randomly or sometime algorithmically and the subset of examples that are selected for this purpose are often referred to as the training set [15].

Feed Forward Neural Network

Many neurons are used in the construction of an

Volume: 05, Issue: 04, Pages:1990-1995 (2014) ISSN : 0975-0290

FFNN; these neurons are connected with each other through specific network architecture. The primary goal of the FFNN is to transform the inputs into meaningful outputs. There is no self-loop or backward feed in this network.

Radial Basis Neural Network

There is no main difference in the network architecture of RBNN compare to FFNN. The only difference is that the radial basis layer of RBNN is located like Hidden layer in FFNN architecture. Radial basis Layer contains different type of neurons, which contains Radial Basis Function (RBF) as an activation function[16].

Artificial Neural Network

Artificial Neural Network (ANN) uses machine learning and pattern recognition methodology [17] to find accurate estimates for software development effort. It is found that ANN improve performance of effort estimation on the basis of mean absolute error [18, 19].

Above all techniques, ANN (Artificial Neural Network) has the ability to discover relationships between the dependent and independent variables. It can also learn from previous data. ANN is the interconnection of artificial neurons. ANNs include the two fundamental components of biological neural networks are Neurons (nodes) and Synapses (weights).

The main focus of this paper is to discuss how we can predict the effort using Artificial Neural Network techniques and a comparative analysis of ANN techniques for software development effort estimation. Section 3 will cover different challenges related to ANN (Artificial Neural Network) based effort estimation, current trends including existing ANN based effort estimation techniques and comparison among these techniques. Section 4 contain summary of current work on Effort Estimation using ANN, and finally section 5 will presents the conclusion.

III. RELATED WORK

The early techniques of software development effort estimation try to solve the problem of accuracy in effort estimation, but none of them produce 100% accurate results.

Kumar, et al [20] used wavelet neural network (WNN) with four models, i.e., WNN-morelet, WNN-guassian, TAWNN-guassian, and TAWNN-morelet. A Threshold acceptance training algorithm is used for wavelet neural network, i.e., TAWNN. WNN-Morelet and WNN-Guassian outperformed all other techniques. Their approach has improve the efficiency reults.

Jaswinder Kaur, et al.[21] implemented a back-propagation ANN of 2-2-1 architecture on NASA dataset consist of 18 projects. Input was KDLOC and development methodology and effort was the output. He got result MMRE as 11.78.

B.Trimula Rao[22] proposed a FLANN for software effort prediction. It generates output (effort) and then processed final output layer. It has one drawback that its relation between inputs and outputs cannot be justified.

Ideri et al.[23] tried to converts hidden information stored in synaptic weights into fuzzy if-then rules. They extracted if-then fuzzy rules from the trained FFNN. These fuzzy rules then encode the neural network information accurately. In 2004, Idri et al. used RBNN for effort estimation. This time, they extracted fuzzy rules for trained RBNN network using Jang and Sun method (1992).

Pichai Jodpimai[24] proposed a model to measure the estimation accuracy. It consists of 3 steps, i.e., First step is to prepare data from each of the database, then it will reduce the number of features. Feed forward neural network is used in the final step to transform the effort estimation problem into classification and functional approximation.

Iman Attarzadeh[25] proposed a new model to accommodate COCOMO II. 5 Scale factors and 17 Effort multipliers were used as input. A sigmoid activation function is used to create network in order to accomplish post architecture COCOMOII model. Results shown in terms of MMRE, and Pred(0.25) to compared with traditional COCOMO.

Ivica Kalichanin-Balich [26] used Feed forward neural network to estimate software development effort of short-scale projects. 132 projects are used to verify the proposed mechanism. Accuracy is measured in terms of MER, i.e., MMER is 0.26, LRM is 0.26 and NN is 0.25.

Reddy et al.[27] compared intermediate COCOMO, radial basis neural network and regression neural networks to predict software development effort. They used the COCOMO’81 database, consisting 63 projects data. Out of these 63 projects, randomly selected 53 projects were used as training data. After evaluating experimental result with different criteria, authors concluded that the radial basis neural networks model performs more accurately for effort prediction.

Attarzadeh[28] proposed a new software development effort estimation model using neural network Initial weights of the networks were set in such a way that it resulted in COCOMOII model. The training method is also explained by the authors. The proposed neural network model gives better result as compared to COCOMO model after proper training.

Vachik S. Dave and Kamlesh Dutta[29] proposed a Modified MMRE. They used NASA dataset consists of 60 projects. They conduct experiments with three different evaluation methods, i.e., MMRE, Modified MMRE, and Relative Standard Deviation (RSD). Three estimation modes are used for this purpose, i.e., Regression analaysis, FFNN, and RBFNN. According to authors, RBFNN is found to be a better method for effort estimation, on the basis of RSD and Modified MMRE.

Sriman[30] proposed RBFN to produce good results for problems having noise inputs, complex relationships between inputs and outputs and where inputs have high noise levels. COCOMO 81 and Tukutuku is used as datasets. Clustering algorithm is used to configure the hidden layer of RBFN. After using widths for models, it is found that accuracy of RBF using minimum width is better than using maximum width.

Park and Baek[31] studied various parameters affecting the software development effort and identified six variables other than the size of the software for more accurate effort estimation using neural network. Authors compared neural network model with the two existing regression models and with human expert judgments, it is found that neural network model is more accurate than all the other estimation methods.

Tronto et al.[32] implemented neural network based model and stepwise regression model for software development effort. Results reported by the authors, restate that the neural network based model estimates software development effort more accurately. Authors also compared their results with the multiple regression, COCOMO and SLIM (Software Lifecycle Management) models, which shows that the neural network model is suitable for effort estimation.

Iwata et al.[33] established neural network based estimation model for complementing missing values. Authors compared the estimation accuracy of neural network model and multiple regression analysis using Welch'st test (1947), which proves that neural network based model is more accurate than multiple regression models.

Reddy and Raju[34] proposed a multilayer feed-forward neural network to accommodate the COCOMO model. COCOMO database is used as a dataset consisting 63 projects. Data set is divided into training set and validation set in the ratio of 80 %: 20 %. Training set consists of 50 projects selected randomly and validation set consists of remaining 13 projects.

Bhatnagar et al[35] compared FFNN and regression modeling approach for the soft-ware effort estimation. Authors estimated effort using McCabe Complexity (1976), Dhama Coupling (Martin et al. 2008) and Source Lines Of Code (SLOC) as independent variable for predicting development time for Martin et al.(2008) database. Authors concluded that the neural network model trained using previous data has good generalization capabilities and is able to successfully predict the effort closely matching the experimental observations with less error value and performs better than the regression modeling.

Afzal [36] compared FFNN with multiple regression model for the database in which 132 projects were used to train the models and 77 projects were used for the validation or testing the models. The experiment was carried out under controlled environment having some specific characteristics mentioned in the paper. The result of the experiment shows that FFNN estimation accuracy is same as that of regression model when it is used to estimate small-scale projects developed by personal practice.

Setiono [37]compared intermediate COCOMO, radial basis neural network and generalized regression neural networks for estimating software development effort. The authors used the COCOMO'81 database, consisting 63 projects data. Out of these 63 projects, randomly selected 53 projects were used as training data. After evaluating experimental result with different criteria, authors concluded that the radial basis neural networks model performs more accurately for effort prediction.

Attarzadeh [38]proposed neural networks based software development effort estimation model. Initial weights of the networks were set in such away that it resulted in COCOMO II model. The training method is also explained by the authors. The proposed neural network model gives better result as compared to COCOMO model after proper training.

Kalichanin-Balich [39] compares linear regression, and Logarithmic regression with FFNN. On the basis of test results, it has been observed that software prediction is more accurate and realistic using FFNN rather than regression and logarithmic models. They used MMRE as evaluation criteria.

IV. GENERAL SUMMARY

Table 1. General Summary

Authors	Methods used	Dataset (number of projects)	Evaluation
Kumar(2008)	Wavelet Neural Networks	IBMDPS(24),CF	MMRE, Pred(0.25), MdMRE
B.Tirimula Rao(2009)	C-FLANN,P-FLANN,L-FLANN	NASA(60)	RMSE
Kazunori Iwata(2009)	ANN Model	Real Data Set (73)	MAE,VAE,MRE,VRE
Sriman Srichandan (2010)	Radial Basis Functional Neural Networks	COCOMO81(252), Tukutuku(53)	MMRE, Pred(0.25)
Jaswinder Kaur(2010)	Back propagation artificial neural network	NASA	MMRE, RMSSE
Pichai Jodpimai(2010)	FeedForward Neural Network, Functional Approximation	COCOMO81(63),NASA60(60),NASA93(93), Albrecht(24),CF(21), Desharnais(81)	MMRE, Pred(0.25)
Iman Attarzadeh (2010)	BackPropogation ANN	COCOMO(61)	MMRE,Pred(0.25)
Ivica Kalichanin-Balich(2010)	Feed Forward Neural Network	132 Projects developed by 40 programmers, 77 projects developed by 24 programmers	MMER
Vachik S. Dave(2011)	RBFNN, FFNN, Regression Analysis	cocomonasa_v1(60)	MMRE, Modified MMRE, RSD
CH.V.M.K.Hari (2011)	Cluster based PSO	COCOMO 81	MARE
Iman Attarzadeh (2012)	ANN-COCOMOII	COCOMO-1(63), NASA93(93)	MMRE, Pred(0.25)
Ali Bou Nassif andLuiz Fernando Capretz(2012)	FeedForward ANN(MLP) based on UCP model	Questionnaire(240)	MMER, Pred(0.25), Pred(0.50), Pred(0.75)
Tirimula Rao Benala(2012)	UKW-FLANN, DBSCAN-FLANN	COCOMO81, NASA93,DESH	MMRE,MdMRE, Pred(0.25)
Jagannath Singh(2012)	Cascade Forwad ANN, Elman ANN, Feed Forward ANN, Recurrent ANN	NASA(60)	MMRE, RMSE, Means BRE, Pred(0.25)
Anupama Kaushik(2012)	Feed Forward Back Propagation	COCOMO(63), NASA2(93)	MRE
V.Khatibi Bardsiri(2012)	ANN, ABE,CART	Desharnais, Maxwell	MMRE, Pred
Sriman Srichandan(2012)	RBFNN	COCOMO 81, Tukutuku	MMRE, Pred(0.25)

V. CONCLUSION

In this paper, currently used Artificial Neural Network based Effort Estimation techniques are presented and evaluated. Latest work on all the techniques is presented with their results on the basis of different evaluation criterion, i.e., MMRE, Pred, RSD etc. It is found that Functional Link ANN (FLANN) reduces the error to maximum limit and decreases the complexity better than other ANN techniques. Hence, it is better to use FLANN in order to achieve accurate software effort prediction.

ACKNOWLEDGEMENTS

I acknowledge COMSATS Institute of Information Technology to support me in this work.

REFERENCES

[1] K. Molokken and M. Jorgensen, A review of software surveys on software effort estimation, in *International Symposium on Empirical*

Software Engineering, 2003. ISESE 2003. Proceedings. 2003, (2003), pp. 223-230.

[2] M. Shepperd and C. Schofield, Estimating software project effort using analogies, *IEEE Transactions on Software Engineering,* , vol. (23) (1997), pp. 736-743.

[3] M. W. Nisar, *et al.*, Software development effort estimation using fuzzy logic-A survey, in *Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08.*,(2008), pp. 421-427.

[4] C. G. Mamoona Humayun, Estimating Effort in Global Software Development Projects Using Machine Learning Techniques, *International Journal of Information and Education Technology* (2) (2012), pp. 208-211.

[5] C. J. Burgess and M. Lefley, Can genetic programming improve software effort estimation? A comparative evaluation, *Information and Software Technology,* (43) (2001), pp. 863-873.

[6] M. Lefley and M. J. Shepperd, Using genetic programming to improve software effort estimation based on general data sets, in *Genetic and Evolutionary Computation—GECCO 2003*,(2003) pp. 2477-2487.

[7] N.-H. Chiu and S.-J. Huang, The adjusted analogy-based software effort estimation based on similarity distances, *Journal of Systems and Software*, (80) (2007), pp. 628-640.

[8] X. Huang, *et al.*, A soft computing framework for software effort estimation, *Soft Computing*, (10) (2006), pp. 170-177.

[9] V. S. Dave and K. Dutta, Comparison of regression model, feed-forward neural network and radial basis neural network for software development effort estimation, *ACM SIGSOFT Software Engineering Notes*, (36) (2011), pp. 1-5.

[10] M. O. Elish, Improved estimation of software project effort using multiple additive regression trees, *Expert Systems with Applications,* (36) (2009), pp. 10774-10778.

[11] R. L. Narendra Sharma, Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement, *International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, (2) 1(2012)*, pp. 301-309.

[12] K. T. Lum, *et al*., The effects of data mining techniques on software cost estimation, in *IEEE International Engineering Management Conference, 2008. IEMC Europe 2008.*, (2008), pp. 1-5.

[13] E. Rashid, *et al.*, The Application of Case-Based Reasoning to Estimation of Software Development Effort, *International Journal of Computer Science and Informatics (IJCSI) ISSN,(2012)* pp. 2231-5292.

[14] C. A. A. W. B. Boehm. (2007). *Software Cost Estimation with COCOMO II*.

[15] C. Mair, *et al.*, An investigation of machine learning based prediction systems, *Journal of Systems and Software*, (53) (2000), pp. 23-29.

[16] N. Tadayon, Neural network approach for software cost estimation, in *International Conference on Information Technology: Coding and Computing, (2005).* pp. 815-818.

[17] G. Wittig and G. Finnie, Estimating software development effort with connectionist models, *Information and Software Technology,* (39) (1997), pp. 469-476.

[18] I. B. Tronto, *et al.*, The artificial neural networks model for software effort estimation,(2006) *Laboratory for computing and applied mathematics, Brazilian national institute for space research–INPE*.

[19] M. Sharif, Mussarat Yasmin, Sajjad Mohsin, Neural Networks in Medical Imaging Applications: A Survey, *World Applied Sciences Journal,* (22) (2013), pp. 85-96.

[20] K. Vinay Kumar, *et al.*, Software development cost estimation using wavelet neural networks, *Journal of Systems and Software,* (81) (2008), pp. 1853-1867.

[21] J. Kaur, *et al.*, Neural Network-A Novel Technique for Software Effort Estimation, *International Journal of Computer Theory and Engineering,* (2) (2010), pp. 1793-8201.

[22] B. T. Rao, *et al.*, A novel neural network approach for software cost estimation using Functional Link Artificial Neural Network (FLANN), *International Journal of Computer Science and Network Security*, (9) (2009), pp. 126-131, 2009.

[23] A. Idri, *et al.*, Validating and understanding software cost estimation models based on neural networks, in *International Conference on Information and Communication Technologies: From Theory to Applications, (2004)*, pp. 433-434.

[24] P. Jodpimai, *et al.*, Estimating software effort with minimum features using neural functional approximation, in *International Conference on Computational Science and Its Applications (ICCSA), (2010)*, pp. 266-273.

[25] I. Attarzadeh and S. H. Ow, Proposing a new software cost estimation model based on artificial neural networks, in *2nd International Conference on Computer Engineering and*

Technology (ICCET), (2010) pp. V3-487-V3-491.

[26] I. Kalichanin-Balich and C. Lopez-Martin, Applying a feedforward neural network for predicting software development effort of short-scale projects, in *Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA), (2010),* pp. 269-275.

[27] P. Reddy, *et al.*, Software effort estimation using radial basis and generalized regression neural networks, *arXiv preprint arXiv:1005.4021*, *(*2010).

[28] I. Attarzadeh, *et al.*, Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation, in *Fourth International Conference on,Computational Intelligence, Communication Systems and Networks (CICSyN), (2012),* pp. 167-172.

[29] V. S. Dave and K. Dutta, Neural network based software effort estimation & evaluation criterion MMRE, in *2nd International Conference on Computer and Communication Technology (ICCCT)* (2011), pp. 347-351.

[30] S. Srichandan, A new approach of Software Effort Estimation Using Radial Basis Function Neural Networks, *International Journal on Advanced Computer Theory and Engineering (IJACTE)* (1) (2012) pp. 2319-2526.

[31] H. Park and S. Baek, An empirical validation of a neural network model for software effort estimation, *Expert Systems with Applications,* (35) (2008) pp. 929-937.

[32] I. F. de Barcelos Tronto, *et al.*, An investigation of artificial neural networks based prediction systems in software project management, *Journal of Systems and Software,* (81) (2008) pp. 356-367.

[33] K. Iwata, *et al.*, Using an artificial neural network for predicting embedded software development effort, in *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD'09.*, (2009), pp. 275-280.

[34] C. S. Reddy and K. Raju, A concise neural network model for estimating software effort, *International Journal of Recent Trends in Engineering,* (1) (2009) pp. 188-193.

[35] R. Bhatnagar, *et al.*, Software development effort estimation neural network vs. regression modeling approach, *International Journal of Engineering Science and Technology,* (2) (2010) pp. 2950-2956.

[36] W. Afzal and R. Torkar, On the application of genetic programming for software engineering predictive modeling: A systematic review, *Expert Systems with Applications,* (38) (2011) pp. 11984-11997.

[37] R. Setiono, *et al.*, Software effort prediction using regression rule extraction from neural networks, in 22nd IEEE International Conference on *Tools with Artificial Intelligence (ICTAI), (2010)* pp. 45-52.

[38] I. Attarzadeh and S. H. Ow, A novel algorithmic cost estimation model based on soft computing technique, *Journal of Computer Science,* (6), (2010),p. 117.

[39] I. Kalichanin-Balich, Applying a Feedforward Neural Network for Predicting Software Development Effort of Short-Scale Projects, presented at Eighth ACIS International Conference on the Software Engineering Research, Management and Applications (SERA), (2010).

Muhammad Waseem Khan is a student of MS (Computer Science) at COMSATS Institute of Information Technology, Wah Campus, Pakistan. He has completed his Bachelor degree from COMSATS Institute of Information Technology, Lahore in 2010. His areas of interest are Image Processing, Information Security and Mobile Computing