Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 2, February 2019, pg.116 – 122

A Neural Network Model for Estimation Having Reusable Components in Software Development

Jyoti Mahajan

Associate Professor, Department of Computer Engineering, Govt. College of Engg. & Technology, Jammu, India
jmahajan1972@gmail.com

Abstract— Software effort estimation is one of the most critical and complex task in software engineering, but it is an inevitable activity in the software development processes. Due to the inherent uncertainty in software development projects such as complex and dynamic interaction factors, change of requirements, intrinsic software complexity, pressure on standardization and lack of software data, it is unrealistic to expect very accurate effort estimation of software development processes. Many estimation models are used to prove accuracy but none of them have able to prove that it has accuracy in all cases of software applications. The paper focuses on the reusability in software development effort estimation. The proposed model is based on dynamic neural network technique using back propagation algorithm which provides better accuracy for effort estimation.

Keywords— DNN; Reusability; Perceptron; Fuzzy; MRE.

I. INTRODUCTION

In computer science and software engineering, reusability is the likelihood a segment of source code that can be used again to add new functionalities with slight or no modification. Reusable modules and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modifications when a change in implementation is required. The ability to reuse relies in an essential way on the ability to build larger things from smaller parts, and being able to identify commonalities among those parts. Reusability is often a required characteristic of platform software. It brings several aspects to software developments that do not need to be considered when reusability is not required. Reuse is an act of synthesizing a solution to a problem based on predefined solutions to sub problems.

The reuse activity [1] is divided into six major steps performed at each phase in preparation for the next phase. Reusability is the degree to which a component can be reused and reduces the software development cost by enabling less coding and more integration [1]. The reusability of assets is different in different contexts. However, there are some characteristics that generally contribute to the reusability of assets. Although many of these characteristics apply to assets in general, we focus in this section, we focus on components as assets. At a high level, we distinguish two aspects of reusability i.e. usability and usefulness [2].

Reusability = Usability + Usefulness

Usability is the degree to which an asset is 'easy' to use in the sense of the amount of effort that is needed to use an asset. Usability as such is independent of functionality of the component. Usefulness is the 'frequency' of suitability for use i.e. usefulness depends on the functionality, the generality and quality of a component [2]. Selby et al [3] identified a number of characteristics of those components , from existing systems, that are being reused at NASA laboratory and reported that the developers were successful in achieving a 32 percent reusability index. Selby's recent experimental study has identified two categories of factors that characterize successful reuse-based software development of large-scale systems: module design factors and module implementation factors [4].

The module design factors that characterize module reuse without revision were: few calls to other system modules (i.e. low coupling), many calls to utility functions (i.e. high cohesion), few input-output parameters, few reads and writes;, and many comments. The main objective of software reuse is to minimize repetition of work, development time, cost and efforts and increase reliability of the systems. It also improves the reusability and portability of the system[5]. Sandhu et al [6][7] have suggested that reusability is being considered for appraisals of employees of an organization to reduce costs and maximizing profits. Through these studies it is evident that that adoption and importance of reusable components in the industry today and effort estimation being used on reusability could answer the anomalies that exist in the current estimation techniques adopted.

II. PROPOSED ESTIMATION TECHNIQUE

The proposed model for Effort Estimation consists of 4 phases namely PreProcessing, Training, Estimation and Analysis. The effort estimation technique proposed consists of a pre processing phase where in the project data considered is analyzed to basically derive the reusability matrix. A project is assumed to be split into a number of modules and the reusability of each module is analyzed to derive the reusability matrix using fuzzy rules.
Estimation of the effort involved to achieve the project goals have been achieved using dynamic neural networks. Prior to estimation the dynamic neural networks are trained using the back propagation algorithm. The trained neural network could be used for estimation the effort involved. The results obtained could be analyzed for resource utilization, financial analysis, delivery time line assertion and many more critical analyses. A project is said to be composed of m modules. Modules could be either reusable or could be considered as new modules (m_n). Each reusable module is analyzed using a judgment model to arrive at the reusable component present. The modules are analyzed at an implementation level and for characterization a threshold ϑ is defined which is arrived based on the judgment model. On characterization the modules are further classified into 3 categories as

- Completely reusable.

A module is considered to be completely reusable if it could be utilized without any changes or without altering any code or design and is represented as m_{cr}

- Reusable with fractional adaptation

A module is considered as a reusable model with partial adaption if at the implementation level the changes to be incorporated are less than the threshold ϑ and is represented as m_{fr}

- Reusable with prominent adaptation

If the changes to be incorporated are greater than the threshold ϑ then the module is considered as a reusable module with prominent adaption represented by m_{pr}. Let Δ represent the changes to be incorporated into a module m for it to be compatible with the project for which estimation is to be achieved. Applying the fuzzy rules the modules could be characterized as follows

$$\begin{Bmatrix} m = m_{cr} \; if \; and \; only \; if \; \Delta = 0 \\ m = m_{fr} \; if \; and \; only \; if \; \Delta < \vartheta \\ m = m_{fr} \; if \; and \; only \; if \; \Delta \geq \vartheta \end{Bmatrix}$$

Consider R to represent the reusable matrix. The effort involved to develop the modules earlier is represented as $\mathcal{E}$. Let us consider that there exist $x, y \; and \; z$ number of $m_{cr}, m_{fr} \; and \; m_{pr}$ modules and their development efforts considered be defined as $\mathcal{E}_{cr}, \mathcal{E}_{fr} \; and \; \mathcal{E}_{pr}$.

Then the reusability matrix obtained based on fuzzy logic could be represented as

$$R = \begin{bmatrix} m_{cr1} & \mathcal{E}_{cr1} & m_{fr1}\mathcal{E}_{fr1} & m_{pr1} & \mathcal{E}_{pr1} \\ \vdots & \vdots & \vdots \ \vdots & \vdots & \vdots \\ m_{crx} & \mathcal{E}_{crx} & m_{fry}\mathcal{E}_{fry} & m_{prz} & \mathcal{E}_{prz} \end{bmatrix}$$

To estimate the effort involved the proposed model uses neural networks. Neural networks of static type are not considered to develop this effort estimation technique as they possess adaptive and learning capabilities for only static in-out relationships. To effectively adapt and learn the dynamic input output of the non linear matrices, the proposed model uses dynamic neural networks. The dynamic Neural Network Adopted in this model is as shown as :

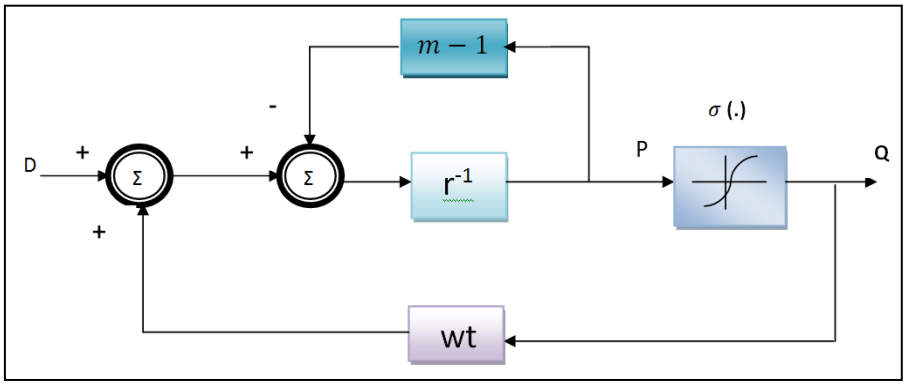


Fig. 1 : DNN Model

The reusable matrix obtained from the pre processing phase is considered for training of the dynamic neural networks. The multilayer perceptron model is used for creating a neural network, as this would be the appropriate network structure which would help in realizing the problem. In a multilayer network there will be one input layer, atleast one hidden layer and one output layer. Backpropagation is used as the training methodology *i.e.*, the learning rule. It is a supervised learning algorithm. It is a learning methodology through which the network trains itself through multiple iterations over the test data. It does so by reducing an error function. The network eventually converges towards accurate values as it is trained with more and more training data. The activation function used here is the sigmoid/bipolar sigmoid function. The activation function is an abstraction of the action potential. It represents whether the cell should fire or not. The output of the dynamic neural network $r(k)$ with respect to the input $p(k)$ is given by

$$p(k+1) = -(m-1)p(k) + wtq(k) + d$$
$$r(k) = \sigma\big(p(k)\big)$$

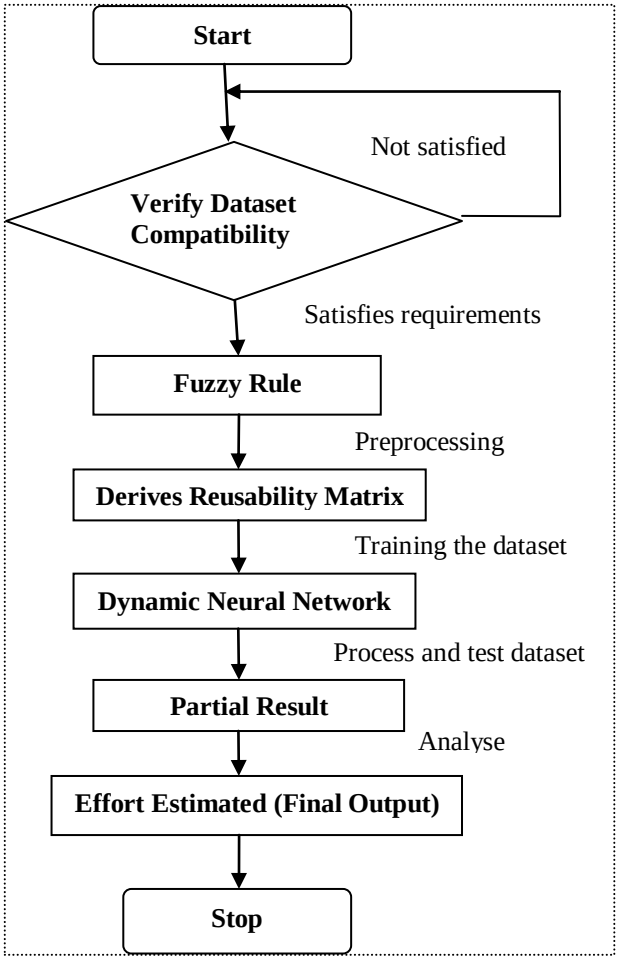
Where σ represents the sigmoid/bipolar sigmoid activation function and $(m-1)$ is the feedback where m is the learning rate constant. The reusability matrix derived using the fuzzy rule was provided to the dynamic neural network in the training process. The trained dynamic neural network on querying provides the effort estimated phase wise and for the entire project.

III. EXPERIMENTATION AND RESULTS

For evaluation two types of datasets are used :-

i. Historical Projects Dataset
ii. Commercial Projects Dataset

The experimental evaluation process considered is shown as a flowchart



The dataset of NASA [8] is used for the comparison of different models and experiments have been conducted to explore strength of the developed DNN based model.

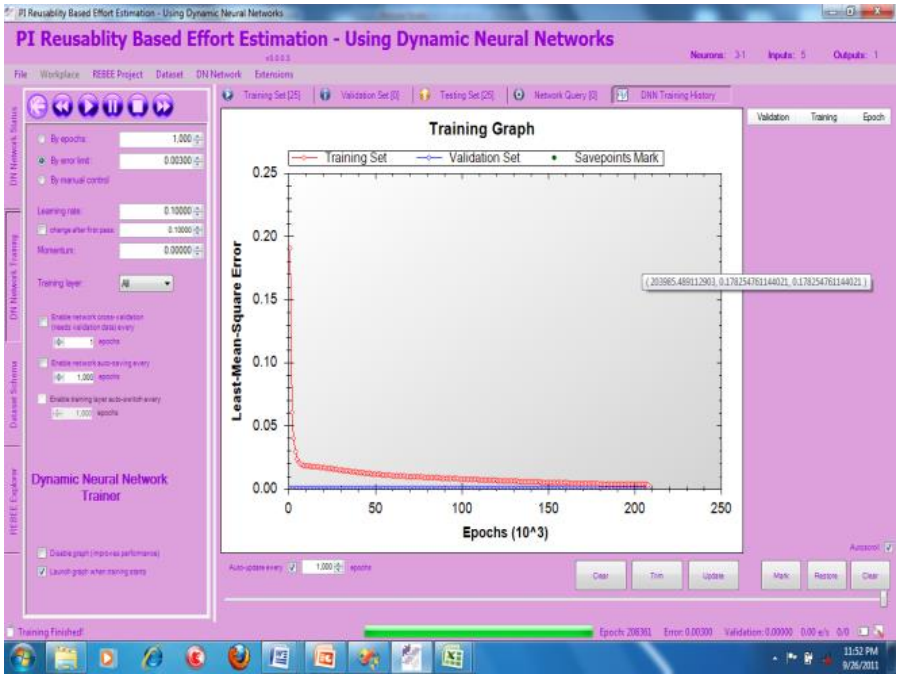


Fig. 2 : Training Graph of Training DataSet

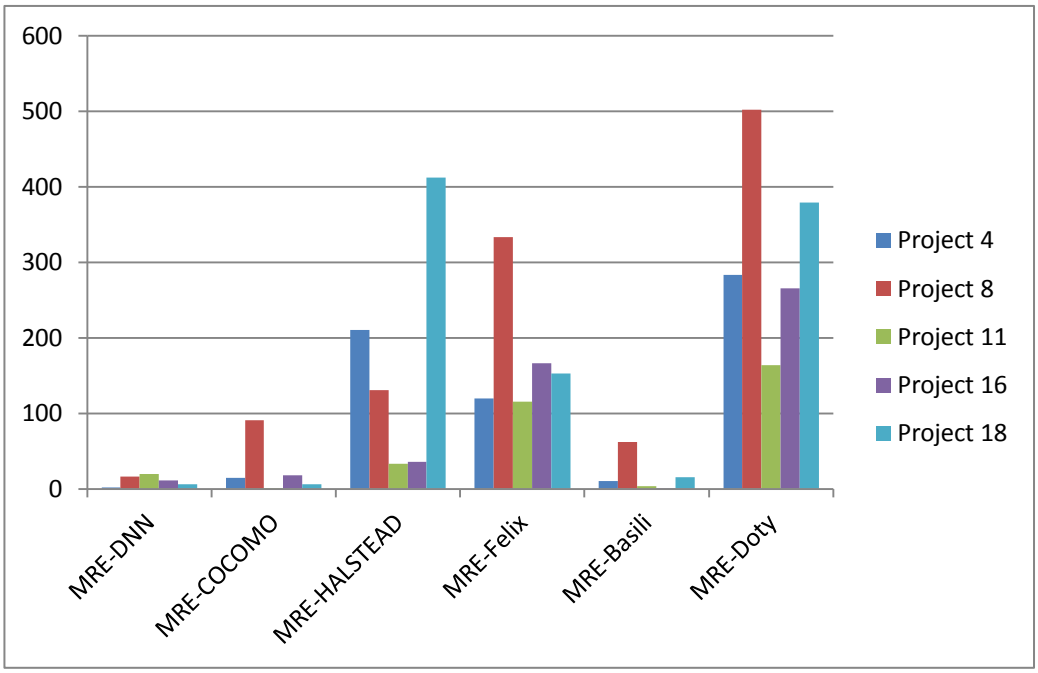


Fig. 3 : MRE Performance of Projects (Test Data) for various models

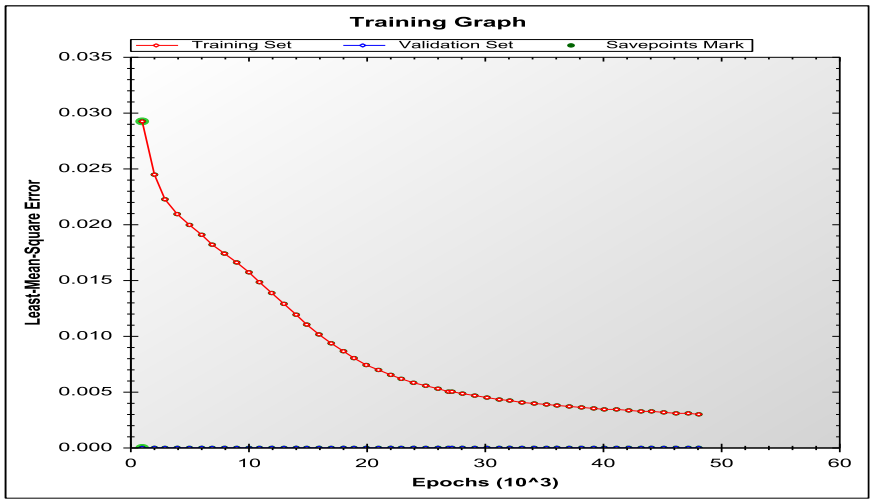
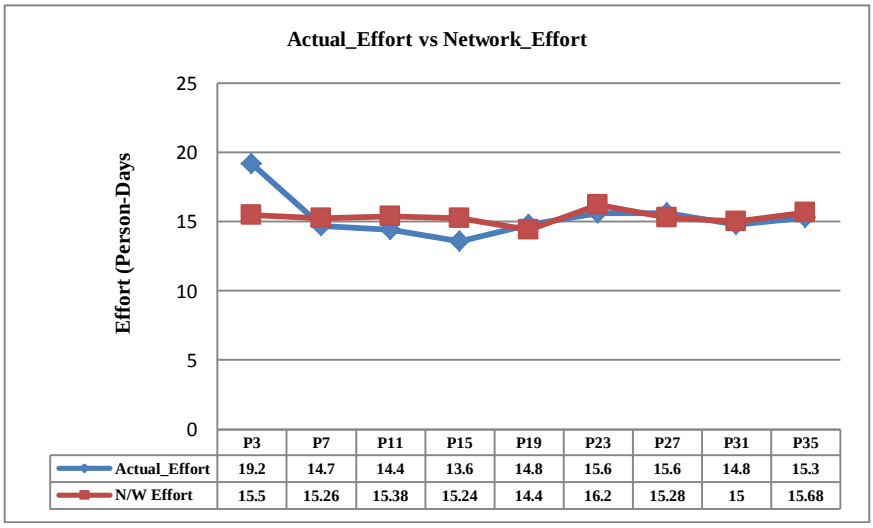


Fig. 4 : Training of Real Project DataSet



	P3	P7	P11	P15	P19	P23	P27	P31	P35
Actual_Effort	19.2	14.7	14.4	13.6	14.8	15.6	15.6	14.8	15.3
N/W Effort	15.5	15.26	15.38	15.24	14.4	16.2	15.28	15	15.68

Fig. 5 : Comparison of Actual Effort and Network Effort

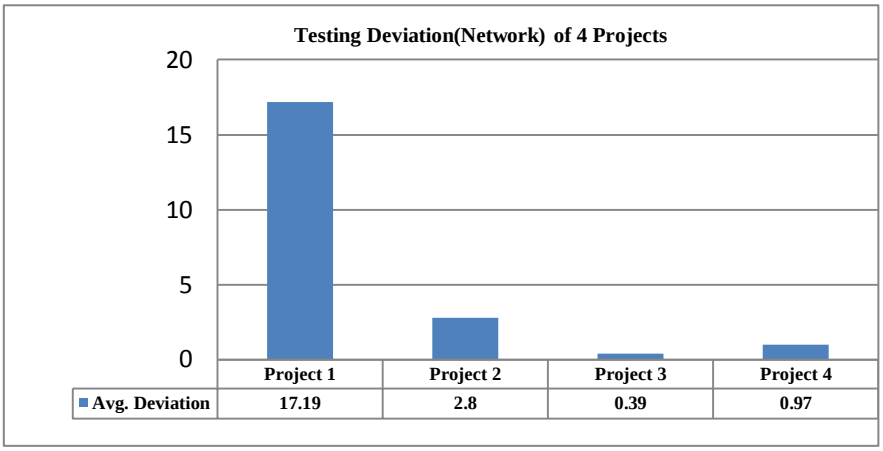


Fig. 6 : Testing Deviation Result of the Real Projects

IV. CONCLUSION

The careful analysis of the results obtained using proposed dynamic neural network based model provides the information that the proposed estimation has a deviation of about 8% over the traditional method that has been chosen. This deviation is not much considering the fact that the effort estimated by the traditional method has also not being found to be accurate when applied to real time projects. The proposed model discussed is evaluated on 39 SEL projects which are of different kinds. The development languages for these projects also varied from project to project. The reusability level of the projects varied from about 0% to a high of 96%. The average estimation error for all the 39 projects was also a low of about 1.25% which proves the efficiency of model.

Further the proposed model has been applied on real time data from four projects that have been specified above and interpretation of the results obtained and mentioned in the graph indicate that the estimated effort has been found to be always on the higher side of actual effort. The deviation found here is found to be on the positive side. When the results were analyzed with the real time data the proposed model has been found to be more accurate as the deviation varies from 0.3 % to 17%.

The DNN model trained using experimental data was found to have good generalization capabilities and is able to successfully predict the effort closely matching the actual effort. From the evaluation results obtained it could be concluded that effort estimation technique discussed in this thesis could be a possible solution for accurate effort estimation for projects of varied types which is not possible with the currently existing effort estimation techniques proposed by various researchers during their study and research.

V. FUTURE SCOPE

The future scope for the proposed model is based in the direction that the model developed needs to be applied to large number of test cases *i.e.*, real time projects as the proposed model has a unique feature of learning through usage. The model converges towards more accurate values as it used over time. The model developed can be evolved even further in the view that more number of parameters which have a minor effect on the effort estimation be also considered for effort estimation and the model can be evolved.

REFERENCES

[1] Jorgensen, M. and D.I.K. Sjoberg, *Impact of effort estimates on software project work.* Information and Software Technology, 2001. 43(15): p. 939-948.

[2] Jones, C., *Estimating software costs: Bringingrealism to estimating (2nd ed.).* New York, NY: McGraw-Hill, 2007.

[3] Selby, Richard W, Empirically analyzing Software reuse in a production environment in Software Reuse : Emerging Technology, IEEE Computer Society Press.

[4] Selby, Richard W., Enabling Reuse-Based Software Development of Large Scale Systems, IEEE Trans. on Software Eng., vol. 31 no. 6, June 2005 pp. 495-510.

[5] Addison, T. and S. Vallabh. *Controlling Software Project Risks - an Empirical Study of Methods used by Experienced Project Managers*. in SAICSIT 2002, Port Elizabeth, South Africa.
[6] P.S. Sandhu, H. Singh, *Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach*, International Journal of Information Technology, vol. 3, no. 3, pp. 209-214, 2006.
[7] P.S. Sandhu, H. Kaur, A. Singh , *Modeling of Reusability of Object Oriented Software System*, Journal of World Academy of Science, Engineering and Technology, no. 56, pp 162 August 2009.
[8] Abran, A. and P.N. Robillard, *Function points analysis: an empirical study of its measurement processes.* IEEE Transactions on Software Engineering, 1996. 22(12): p. 895-910.
[9] B. Peischl, M. Nica, M. Zanker, *Recommending effort estimation methods for software project management*, in Proc. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. 03, pp 77-80, 2009.
[10] M. Jørgensen, *Practical Guidelines for Expert-Judgment- Based Software Effort Estimation*, IEEE Software, vol. 22, pp. 57-63, May-June 2005.
[11] M.J. Basavaraj, K.C Shet, *Empirical validation of Software development effort multipliers of Intermediate COCOMO Model*, Journal of Software, vol. 3, vo. 5, pp 65 May 2008.

