ISSN (e): 2250 – 3005 || Volume, 06 || Issue, 09|| September – 2016 ||
International Journal of Computational Engineering Research (IJCER)

Function Point Software Cost Estimates using Neuro-Fuzzy technique

Ihtiram Raza Khan[1], Prof. M.Afshar Alam[2]
[1]Department Of Computer Science, Jamia Hamdard, New Delhi – 62. India
[2] Professor,Department Of Computer Science, Jamia Hamdard, New Delhi – 62. India

ABSTRACT
Software estimation accuracy is among the greatest challenges for software developers. As Neuro-fuzzy based system is able to approximate the non-linear function with more precision so it is used as a soft computing approach to generate model by formulating the relationship based on its training. The approach presented in this paper is independent of the nature and type of estimation. In this paper, Function point is used as algorithmic model and an attempt is being made to validate the soundness of Neuro fuzzy technique using ISBSG and NASA project data.

Keywords: *Soft Computing, Neuro-fuzzy, Effort estimation, Function point, Mean Magnitude Relative Error, LOC, COCOMO.*

I. INTRODUCTION

Accurate software estimation such as size estimation, effort estimation, cost estimation, quality estimation and risk analysis is a major issue in software project management. If the estimation is not properly done, it may result in the failure of software project. Accurate software estimation can provide powerful assistance for software management decisions. The principal challenges are 1) the relationships between software output metrics and contributing factors exhibit strong complex nonlinear characteristics; 2) measurements of software metrics are often imprecise and uncertain; 3) difficulty in utilizing both expert knowledge and numerical project data in one model. In this research proposal, a soft computing framework is presented to tackle this challenging problem.

Soft computing is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, low-cost solutions, and close resemblance to human-like decision making. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost by seeking for an approximate solution to an imprecisely/precisely formulated problem.

II. LITERATURE REVIEW

As software development has become an essential investment for many organizations, software estimation is gaining an ever-increasing importance in effective software project management. In practice, software estimation includes cost estimation, quality estimation, risk analysis, etc. Accurate software estimation can provide powerful assistance for software management decisions[1]. The principal challenges are 1) the relationships between software output metrics and contributing factors exhibit strong complex nonlinear characteristics; 2) measurements of software metrics are often imprecise and uncertain; 3) difficulty in utilizing both expert knowledge and numerical project data in one model. To solve software estimation problems, soft computing framework is based on the ”divide and conquer” approach[2, 3]. Literature is available in Haykin S, Neural Networks: A Comprehensive Foundation[4] and Zadeh L A, Fuzzy Logic[5].

Huang X, Ho D, Ren J, Capretz L have given an insight into A Soft Computing Framework for Software Effort Estimation”[6]. Ali Idri and M.Khoshgoftaar have published a paper on “Can Neural nets be easily interpreted in Software Cost Estimation”[7].

SOFT COMPUTING FRAMEWORK

The soft computing framework, or NF Model as presented in Figure 1, consists of the following components[6]:

- 1)Pre-Processing Neuro-Fuzzy Inference System (PNFIS) used to resolve the effect of dependencies among contributing factors of the estimation problem, and to produce adjusted rating values for these factors,
- 2)Neuro-Fuzzy Bank (NFB) used to calibrate the contributing factors by mapping the adjusted rating values for these factors to generate their corresponding numerical parameter values,
- 3)Module that applies an algorithmic model relevant to the nature of the estimation problem to produce one or more output metrics.

where N is the number of contributing factors,
M is the number of other variables in the Algorithmic Model,
RF is Factor Rating,
ARF is Adjusted Factor Rating,
NFB is the Neuro-Fuzzy Bank,
FM is Numerical Factor/Multiplier for input to the Algorithmic Model,
V is input to the Algorithmic Model,
and Mo is Output Metric.

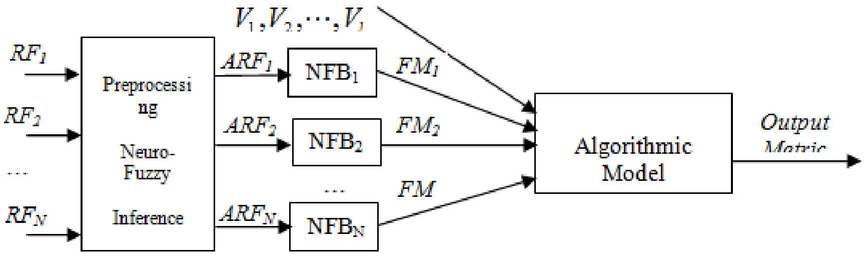


Figure 1: Neuro-Fuzzy Algorithmic (NFA) Model for Estimation

APPLYING IT TO FUNCTION POINT:

The concepts of Function Point is being discussed here, whose aims is to fit specific software application, to reflect software industry trend and to improve cost estimation. Neuro-Fuzzy is a technique which incorporates the learning ability from neural network and the ability to capture human knowledge from fuzzy logic.

Function Points (FP) is an ideal software size metric to estimate cost since it can be obtained in the early development phase, such as requirement, measures the software functional size from user's view, and is programming language independent.

The significant relationship between the software size and cost has been recognized for a long time. In the classical view of cost estimation process (Figure), the outputs of *effort* and *duration* are estimated from software *size* as the primary input and a number of *cost factors* as the secondary inputs. There are mainly two types of software size metrics: Source Lines of Code (SLOC) and Function Points. SLOC is a natural artifact that measures software physical size but it is usually not available until the coding phase and difficult to have the same definition across different programming languages. Function Points is an ideal software size metric to estimate cost since it can be obtained in the early development phase, such as requirement, measures the software functional size, and is programming language independent. Calibrating Function Points incorporates the historical information and gives a more accurate view of software size. Hence more accurate cost estimation comes with a better software size metric.

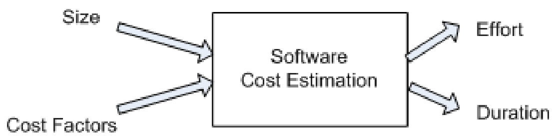


Figure 2: Classical View of Cost Estimation Process.

The paper proposes an approach to estimate cost using Function Points with Neuro-Fuzzy technique. The model overview and two parts of the model: fuzzy logic part and neural network part are described here.

The block diagram shown in Figure 4 gives an overview of this approach. The project data provided by ISBSG is imported to extract an estimation equation and to train the neural network. An estimation equation is extracted from the data set by statistical regression analysis.
Fuzzy logic is used to calibrate Function Points complexity degree to fit specific application. Neural network calibrates UFP weight values to reflect the current software industry trend by learning from ISBSG data.

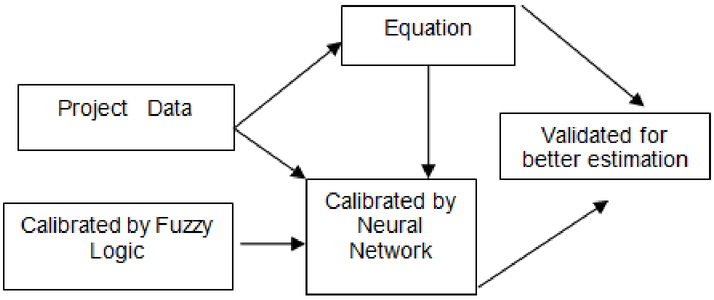


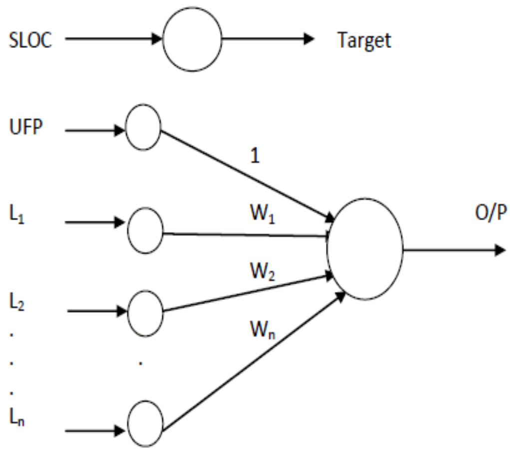
Figure 3: Block Diagram of Neuro-Fuzzy Approach.

Fuzzy Logic Part

The fuzzy logic part calibrates the Function Points complexity degree to fit the specific application. A fuzzy logic system is constructed based on the fuzzy set, fuzzy rules and fuzzy inference. The input fuzzy sets are to fuzzify the component associated file numbers and the output fuzzy set are to fuzzify the complexity classification. The fuzzy rules are defined in accordance with the original complexity weight matrices. The fuzzy inference process using the Mamdani approach is applied based on the fuzzy sets and fuzzy rules.

Neural Network Part

The neural network part is aiming at calibrating Function Points to reflect the current software industry trend. By learning from ISBSG data repository, this part is believed to achieve the calibration goal. The neural network is constructed to receive 15 UFP breakdowns as inputs to give the work effort as the desired output. A back-propagation learning algorithm is conducted in order to minimize the prediction difference between the estimated and actual efforts (We will be checking the performance using RBF network also). An effort estimation equation is extracted based on the data subset using statistical regression analysis. The equation in the form of $\boldsymbol{Effort = A \cdot UFP^{B}}$ is achieved with the help of regression techniques.



$$Z = (W_1L_1 + W_2L_2 + \cdots + W_nL_n) * UFP$$

Figure 4 Neuro-Fuzzy Approach

III. ESTIMATION CRITERIA AND VALIDATION RESULTS

The validation results of the experiments are to be assessed by Mean Magnitude Relative Error (MMRE) for estimation accuracy. MMRE is defined as: for *n* projects,

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}(|\,Estimated_i - Actual_i\,| / Actual_i)$$

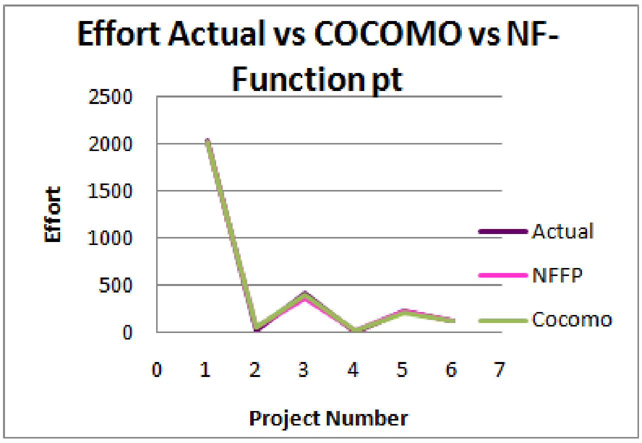
Table 1 shows the MRE and MMRE for Neuro-Fuzzy Function point against the Actual effort

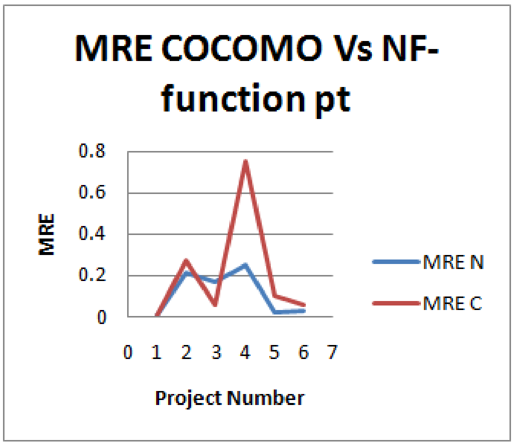
Project No.	Actual Effort	NF-function pt effort	MRE	MMRE
P1	2040	2020	0.009804	
P2	33	40	0.212121	
P3	423	350	0.172577	11.65%
P4	8	10	0.25	
P5	230	225	0.021739	
P6	122	118	0.032787	

Table 2 shows the MRE and MMRE for COCOMO against the Actual effort

Project No	Actual Effort	COCOMO effort	MRE	MMRE
P1	2040	2018	0.010784	
P2	33	42	0.272727	
P3	423	397	0.061466	
P4	8	14	0.75	21.15%
P5	230	205	0.108696	
P6	122	114	0.065574	

The plot shows NF-Function point Effort to be the least (best) in comparison to COCOMO and Actual Effort.





The MRE plot shows MRE of NF-Function Point to be lower than that of COCOMO.

ISBSG project data and NASA project data has been used to validate these approaches.

IV. CONCLUSION

The paper introduces an efficient neuro-fuzzy model along with Function Point model for software cost estimation. The proposed neuro-fuzzy model is based on Mamdani based Neuro Fuzzy system. The advantage of using mamdani based neuro-Fuzzy model is it's intuitive, it has widespread acceptance and it's well suited to human cognition.

This Research work concludes that soft computing promises an efficient and accurate software development cost estimation model than the standard algorithmic Fucntion Point and COCOMO-II model. The Fuzzy inference system based on Mamdani has MMRE of 11.65%.

The result shows that Mamdani Neuro-Fuzzy Function Point model is showing high degree of accuracy than the COCOMO-II model. Thus it brings a new area of research of improvising the software development cost estimation.

ACKNOWLEDGEMENT

I acknowledge and thank all the sources from where reference has been used in the paper.

REFERENCES

[1]. B. Boehm, *Software Cost Estimation with COCOMO II*, Prentice Hall PTR, Upper Saddle River, New Jersey, 2000.

[2]. X. Huang, D. Ho, L. Capretz and J. Ren "Novel Neuro-Fuzzy Models for Software Cost Estimation", *Proc.of the Third International Conference on Quality Software*, IEEE Computer Society Press, Dallas, TX, USA, 2003.

[3]. X. Huang, D. Ho, J. Ren and L. Capretz "An Intelligent Approach to Software Cost Prediction", *Proc. Of the 18th International Forum on COCOMO and Software Cost Modeling*, Los Angeles, CA, USA, 2003

[4]. Haykin S, Neural Networks: A Comprehensive Foundation. Prentice Hall, 1998.

[5]. Zadeh L A, Fuzzy Logic. Computer, Vol 21, pp. 83-93, 1988.

[6]. Huang X, Ho D, Ren J, Capretz L, A Soft Computing Framework for Software Effort Estimation. Soft Computing Journal, Springer, available at www.springeronline.com, 2005.

[7]. Ali Idri, Taghi M. Khoshgoftaar, Alain Abran "Can Neural nets be easily interpreted in Software Cost Estimation", Proc. Of World Congress on Computational Intelligence, Hawaii, 2002.

[8]. Efficient Software Cost Estimation using Neuro-Fuzzy Technique- Ihtiram Raza Khan, Prof M. Afshar Alam In ISCET 2010 International Symposium on Computer Eng& Technology