

A Neuro Fuzzy Algorithm to Compute Software Effort Estimation

N. Shivakumar¹, N. Balaji² and K. Ananthakumar³

¹ Thiagarajar College of Engineering

Received: 9 December 2015 Accepted: 1 January 2016 Published: 15 January 2016

Abstract

Software Effort Estimation is highly important and considered to be a primary activity in software project management. The accurate estimates are conducted in the development of business case in the earlier stages of project management. This accurate prediction helps the investors and customers to identify the total investment and schedule of the project. The project developers define process to estimate the effort more accurately with the available mythologies using the attributes of the project. The algorithmic estimation models are very simple and reliable but not so accurate. The categorical datasets cannot be estimated using the existing techniques. Also the attributes of effort estimation are measured in linguistic values which may leads to confusion. This paper looks in to the accuracy and reliability of a non-algorithmic approach based on adaptive neuro fuzzy logic in the problem of effort estimation. The performance of the proposed method demonstrates that there is a accurate substantiation of the outcomes with the dataset collected from various projects. The results were compared for its accuracy using MRE and MMRE as the metrics. The research idea in the proposed model for effort estimation is based on project domain and attribute which incorporates the model with more competence in augmenting the crux of neural network to exhibit the advances in software estimation.

Index terms— ANFIS, effort estimation, MRE, MMRE

1 A Neuro Fuzzy Algorithm to Compute Software Effort Estimation

Abstract-Software Effort Estimation is highly important and considered to be a primary activity in software project management. The accurate estimates are conducted in the development of business case in the earlier stages of project management. This accurate prediction helps the investors and customers to identify the total investment and schedule of the project. The project developers define process to estimate the effort more accurately with the available mythologies using the attributes of the project. The algorithmic estimation models are very simple and reliable but not so accurate. The categorical datasets cannot be estimated using the existing techniques. Also the attributes of effort estimation are measured in linguistic values which may leads to confusion. This paper looks in to the accuracy and reliability of a non-algorithmic approach based on adaptive neuro fuzzy logic in the problem of effort estimation. The performance of the proposed method demonstrates that there is a accurate substantiation of the outcomes with the dataset collected from various projects. The results were compared for its accuracy using MRE and MMRE as the metrics. The research idea in the proposed model for effort estimation is based on project domain and attribute which incorporates the model with more competence in augmenting the crux of neural network to exhibit the advances in software estimation.

Keywords: ANFIS, effort estimation, MRE, MMRE.

2 I. Introduction

achieving software economics in large-scale software development projects are very important today. Software effort estimation is the process of determining the accurate effort required to maintain or develop a software. It is always an important practical problem in software engineering which is still unsolved. Effort estimates are done in initial stages of software engineering to calculate the effort in person-months required for the software development. Accurate effort estimation helps in planning design construction and transition phases of development and prioritize the components in business case. Unreliable estimates is the main important reason for project failure, which is expressed in 2007 Comp TIA survey of thousand IT professionals, finding that three of the four most-cited causes of IT project failure are due to poor estimation. Boehm proposed a method called COCOMO that utilizes some experimental equation to estimate the effort using inputs like Kilo lines of code (KLOC), number of functions and other effort drivers. Neural network sare introduced in effort estimation process mainly for the training and learning from previous data. The model identifies a positive correlation between the dependent (effort) and independent variables (effort drivers). The half of the available data sets can be given for training and the remaining can be used to derive effort. The other techniques of software effort estimation are bottom-up, top-down, analogy estimation and expert judgments.

3 II.

4 Related Work

Cuauhtemoc [1] provides justification that Fuzzy logic can be used to predict the effort of the small programs based on lines of code obtained from new and changed (N&C) and reused code from small programs developed by 74 programmers. This was used as the input for the fuzzy model for estimating effort and the accuracy of output was compared with the accuracy of Statistical regression model using the comparison criterion Mean Magnitude Error Relative to the estimate.

Shiyna [2] compares the frameworks designed by using fuzzy logic and Neural Networks based on the accuracy of effort estimation. COCOMO NASA dataset had been used as the input for both the frameworks. These frameworks are validated using the parameters MMRE (Mean Magnitude of Relative Error) and Pred (Prediction Accuracy). The results show that Fuzzy Logic based framework works better when compared to the Neural Network framework.

Ochodek [3] proposed the usage of Use Case Points (UCP) method to estimate the effort based on the use case model and two adjustment factors (With or Without Unadjusted Actor Weights). The cross-validation procedure has been used to compare the variants of adjustment factors. A group of 14 projects is considered as input which are used to arrive at a conclusion that the UCP method can be simplified without the use of adjustment factors.

Iman [4] compares the software effort estimation computed by the conventional methods like function points, regression models and COCOMO with the model designed using fuzzy logic. The parameter Mean Magnitude of Relative Error (MMRE) is used to compute the accuracy of the considered methods.

Anjana Bawa [5] explains the usage of Artificial Neural Networks to estimate the project effort as it is capable of learning from the previous data. The machine learning algorithms, Back-Propagation and Cascade Correlation are used to learn and classify the dataset and hence estimate the effort using the Neural Networks.

By analyzing the previous work, it is evident that fuzzy logic is better than the conventional methods of effort estimation. By using the package points the complexity of estimating the lines of code for the considered software is reduced. By using the factor refinement, the time taken to compute the effort is less compared to the previous method where 15 attributes were obtained from the programmer to compute the effort.

5 III.

6 Proposed Approach

We have considered 93 instances of NASA historical project data and also investigated and gathered thirty projects from many case studies and experiments [11][12][14] [15] with consists of 15 attributes and actual effort along with domain, area of work, Size. The fifteen attributes are converted to three index valued labelled as Human Perception and Performance Index (HPPI), Machine Requirement and Performance Index (MRPI), Process Requirement and Performance Index (PRPI). Adaptive Neuro-Fuzzy model (The Figure 1) for software development effort estimation is perfect in the learning and good interpretability. Artificial neural networks are made up of processing units in a parallel manner called as neurons these neurons are inter linked by connections. The input for this model is six grouped attributes. Each attribute represents one factor which leads to the development effort. Table 1 describes the refinement of attributes in such a way that 15 effort multipliers are grouped in 3 clusters of refined attributes whose values are obtained from the software project developer. This proposed model consolidates neural networks and fuzzy-logic principles in a combined ANFIS framework. This inference system correlates to the learning capability of fuzzy IF THEN rules to approximate the nonlinear functions. The package point (figure 2) provides an alternate way to estimate the size that needs to be applied in a software project. Unlike function points and class points approach, the package points proves to be highly efficient in aiding to estimate effort and it is proven to work well with the ERP projects. In order to compute

the package points, inputs are obtained namely, scope, tasks and complexities. Package points are defined by standardizing the number of modules in the project primarily. Then the tasks to complete the modules are prioritized and defined by the client as per their requirement. Finally the complexity factors are loaded to arrive at the package point for the considered module.

7 b) Validation in ANFIS Model

ANFIS is a hybrid supervised method which adopts a hybrid learning algorithm to determine the parameters for fuzzy inference systems. It utilizes both least-squares method and propagation gradient descent method. This is used for training FIS membership function parameters to examine the given training data set. ANFIS can be executed using an optional argument for model validation. This is called as checking model for over fitting. The argument used for this is called as checking data set.

8 c) Fuzzy Rule Inference

Fuzzy rules are generated using package points, domain, type, Human Perception and Performance Index (HPPI), Machine Requirement and Performance Index (MRPI), Process Requirement and Performance Index (PRPI) and Actual effort. A fuzzy set is illustrated using a membership function that relates with every point in the fuzzy set that comprises of numbers in the interval $[0, 1]$, known as degree or grade of membership. Membership function used in this research work is Triangular Membership Function. Triangular Fuzzy Number (TFN) is defined using a triplet $(?, m, ?)$, where m denotes modal value, $?$ and $?$ signify the right and left boundary correspondingly and is expressed as: The equivalent characteristics of the rules the most significant aspect of fuzzy logic systems. Instead of sharp swapping among modes based on breakpoints, logic flows efficiently from sections where the system's performance is governed by one rule or another. $\mu(x) = \begin{cases} 0, & x \leq x_1 \text{ or } x \geq x_3 \\ \frac{x - x_1}{m - x_1}, & x_1 < x < m \\ \frac{x_3 - x}{x_3 - m}, & m < x < x_3 \end{cases}$

Defuzzification converts from fuzzy to crisp conversions. The process converts the fuzzy value to the estimated value for the single data set. This is similar to a "rounding off" method. Defuzzification converts the collection of membership function data to a single scalar quantity in corresponding membership degrees. It is characterized instructure of rules that convert variables to a fuzzy result, that is, the outcome is defined in terms of membership in fuzzy sets.

(2) IV.

9 Experimental Design a) Evaluation Criteria 1. Mean Magnitude Relative Error (MRE), is an error ratio

between the absolute deviations of prediction to the actual effort in each of the referred project $\frac{\sum |e_i - \hat{e}_i|}{\sum e_i}$ (3)

10 Mean Magnitude Relative Error (MMRE) is the average value of MER of all the referred projects

$\frac{\sum \text{MRE}_i}{n}$ (4)

The Table 2 indicates the Package point and its subsequent domain with its actual effort and the nonalgorithmic estimated effort using the proposed method. The effort in the dataset is compared with this estimated effort $\frac{\sum |e_i - \hat{e}_i|}{\sum e_i}$ (3)

The preceding list of algorithmic models are tested including KLOC (Kilo Lines Of Code) value from the data sets and efforts were estimated and these estimates are compared with the results obtained from Adaptive Neuro Fuzzy model .

11 ? Halstead Model-This model developed by

Halstead, concerning the supplied lines of source code from the programmer and formulates a relation, The following table summarizes the MMRE value of all the algorithmic and non-algorithmic models discussed above for all the referred projects. Effort = $0.7 * (KLOC)^{0.5}$

12 Conclusion

Early effort estimation in software development lifecycle is an important activity for project planning and resource allocation. This research work proposes an efficient model in estimating the software effort. The outcomes of the estimation obtained using the direct algorithmic methods indicates the divergence between the actual and the estimated effort. The outcome of nonalgorithmic method comprising of the adaptive neuro technique based estimation decreases the Mean Magnitude of Relative Error (MMRE). Hence the examination of effort from algorithmic method and nonalgorithmic method prove that adaptive neuro fuzzy based estimation is more efficient than the algorithmic methods for the estimation process. The success of estimation depends upon the

155 accuracy and stability of the method in various measures. Future work is planned to investigate the clustering
algorithms in estimation process and apply Neuro Fuzzy model on large datasets. ¹



Figure 1: Figure 1 :

¹© 2016 Global Journals Inc. (US)

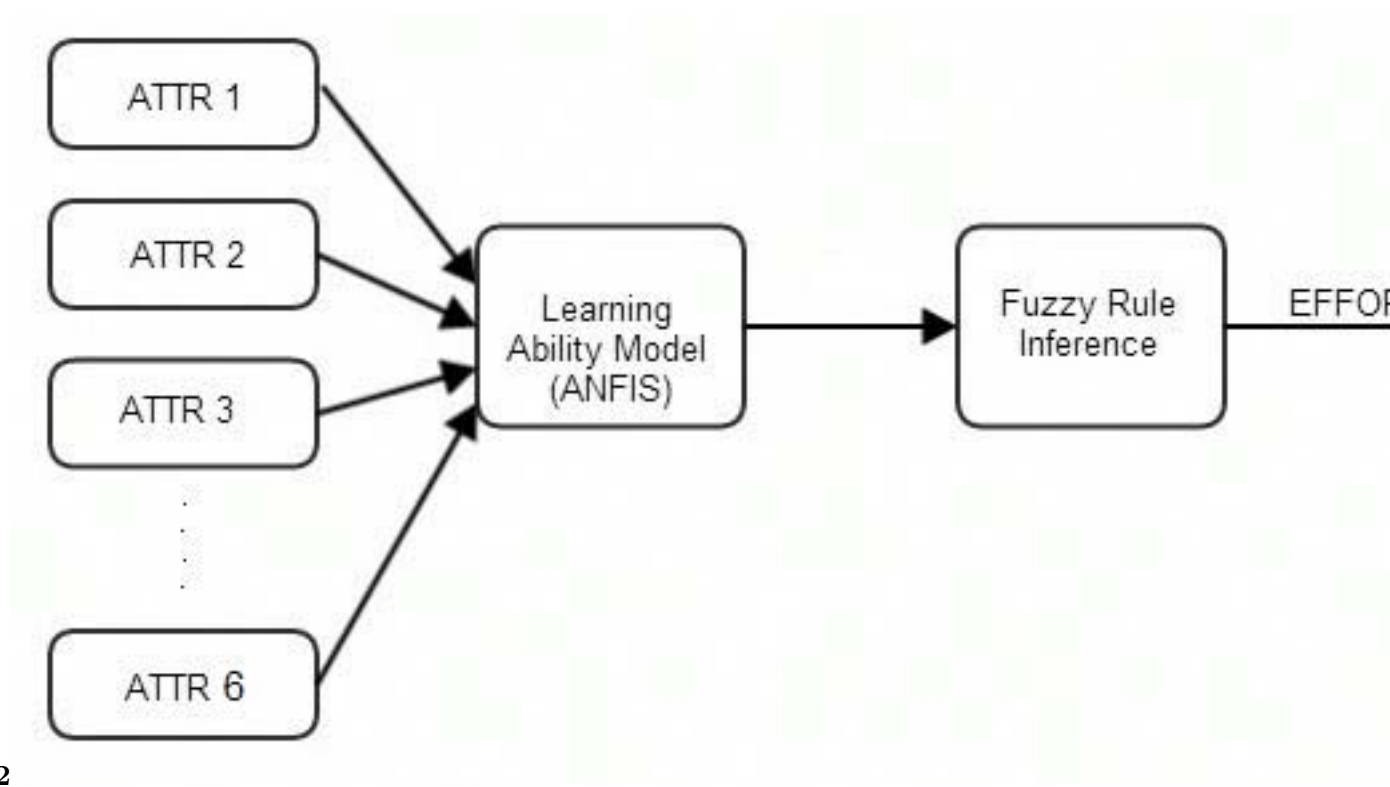


Figure 2: Figure 2 :

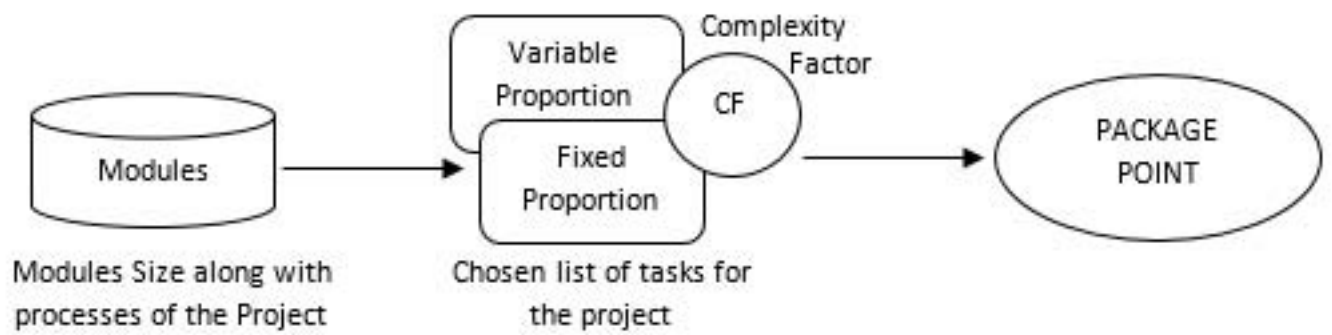


Figure 3: Global

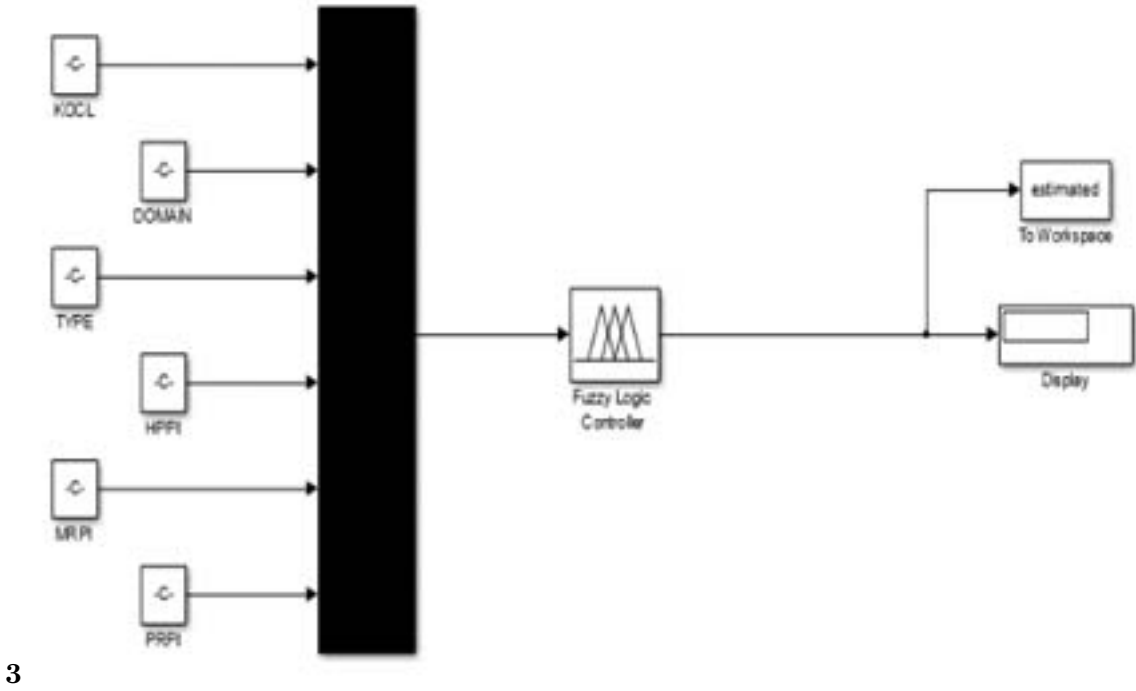


Figure 4: Figure 3 :

1

REFINED AT- EFFORT MULTIPLIERS	
TRIBUTES	
Human Perception and Performance Index (HPPI)	1. Analyst Efficiency
	2. Programmer Efficiency
Machine Requirement and Performance Index (MRPI)	3. Application Maturity
	4. Modern Programming Practices
Process Requirement and Performance Index (PRPI)	5. Use of Software tool
	6. Experience
	7. Language Experience
	8. Time Constraint for CPU
	9. Turnaround Time
	10. Machine Volatility
	11. Process Complexity
	12. Storage Requirement
	13. Schedule Constraint
	14. Database Size
	15. Required Reliability

Space

Software

Figure 5: Table 1 :

2

										Year 2016
										25
										Volume XVI
										Issue I Version
										I
										() C
										Global Journal
										of Computer
										Science and
										Technology
Area Of Domain	Avionics mon-	Package	Actual	Estimated						
itoring Mission	planning Sim-	Point 25.9	Effort	Effort						
ulation Monitor	Control Real	31.5 66.6	117.6 60	138.4 112						
Data Processing	Communications	70 177.9	352.8 458	402 561						
Batch Data Processing		240 25.9	124 192	397 322						
			117.6	119						
Data Capture		31.5	60	67.7						
Launch Processing		66.6	352.8	360.9						
Application Ground		70	458	459						
Utility		177.9	124	128						
Operating Systems		240	192	193.2						

Figure 6: Table 2 :

3

Doty Bailey Halstead			COCOMO	COCOMO	Fuzzy Estimation
			I	II	
MMRE 5.15	6.47	6.06	9.14	7.54	3.11
V.					

Figure 7: Table 3 :

Mean Relative Error	0 5 10 15 20	1 11 21 31 41 51 61 71 81 91	Error	15	1 11 21
			Mean	20	61 71 81
			Rela-	0 5	Actual I
			tive	10	Estimat
		Projects			
		Actual Effort	COCOMO		
			II		
20					
15					
Mean Rel-	0 5	1 11 21 31 41 51 61 71 81 91	Projects	Actual Effort	Halstead
ative Error	10				

Figure 8:

-
- [Azzeha et al. (2015)] ‘An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation’. M Azzeha , Ali Bou Nassifb , Leandro L Minkuc . *Journal of Systems and Software* may 2015.
- [El Bajta ()] ‘Analogy-Based Software Development Effort Estimation in Global Software Development’. M El Bajta . *Global Software Engineering Workshops (ICGSEW)*, 2015. p. . (IEEE 10th International Conference on. on page(s))
- [Azzeh et al. ()] ‘Analogy-based software effort estimation using Fuzzy numbers’. Mohammad Azzeh , Daniel Neagu , Peter I Cowling . *The Journal of Systems and Software* 2011. 84.
- [Jørgensen ()] ‘Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation’. Magne Jørgensen . *Information and Software Technology*, 2011. 53.
- [Ap Subriadi et al. (2014)] ‘Critical review of the effort rate value in use case point method for estimating software development effort’. Ap Subriadi , A P Sholiq , Ningrum . *Journal of Theoretical and Applied Information Technology* January 2014.
- [Anda et al. ()] *Estimating Object-Oriented Software Projects with Use Cases*, Bente Anda , Endre Angelvik , Kirsten Ribu . 2002. 2001. Springer. University of Oslo (Improving Estimation Practices by Applying Use Case Models)
- [Nassif et al. (2012)] ‘Estimating Software Effort Using an ANN Model Based on Use Case Points’. A B Nassif , L F Capretz , D Ho . *Machine Learning and Applications (ICMLA)*, 2012 11th International Conference on, Dec. 2012. p. . (On page(s))
- [Emad and El-Sebakhy ()] ‘Functional networks as a novel data mining paradigm in forecasting software development efforts’. A Emad , El-Sebakhy . *Expert Systems with Applications* 2011. 38.
- [Kazemifard et al. ()] ‘Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCCE) using Multi-Agent Systems’. M Kazemifard , A Zaeri , N Ghasem-Aghae , M A Nematbakhsh , F Mardukhi . *Applied Soft Computing* 2011. 11.
- [Huang et al. ()] ‘Integration of the grey relational analysis with genetic algorithm for software effort estimation’. Sun-Jen Huang , Nan-Hsing Chiu , Li-Wei Chen . *European Journal of Operational Research* 2007. 188.
- [References Références Referencias Network International Journal of Electronics and Computer Science Engineering ()] ‘References Références Referencias Network’. *International Journal of Electronics and Computer Science Engineering* 2011. 1 (3) .
- [Nassif and Capretz ()] ‘Software Effort Estimation in the Early Stages of the Software Life Cycle Using a Cascade Correlation Neural Network Model’. A Nassif , L F Capretz . *13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD)*, 2012.