A Novel Approach to the Early Stage Software Development Effort Estimations using Neural Network Models: A Case Study

Roheet Bhatnagar
Department of CSE
Sikkim Manipal Institute of Technology, Majitar,
Rangpo, East Sikkim, INDIA

Mrinal Kanti Ghose
Department of CSE
Sikkim Manipal Institute of Technology, Majitar,
Rangpo, East Sikkim, INDIA

Vandana Bhattacharjee
Department of CSE
BITEC, Lalpur,
BIT Mesra, Ranchi, INDIA

ABSTRACT

Software development is a team activity which has got well defined stages. At every stage we have a milestone and a milestone needs to be achieved in order to move to the next level. On time and within budget delivery is the ultimate goal of software projects. Software project management is the set of processes and guidelines which help us in achieving the on time & within budget product delivery. Lot of studies has been carried out on software development effort estimations which forms a key part of the software project management. Many models have been proposed over the years for cost estimations but it is still a subject of constant research due to the ever changing nature of software development. This paper describes a novel approach towards the effort estimations at the early stages of software development life cycle (SDLC). A dataset has been created based on the Entity Relationship Diagrams (ERD's) developed by different engineering students as part of their Major Project's spreading different batches & year. In this paper three different NN models are used and their results are compared based on the standard evaluation criteria's such as MMRE, MRE, BRE and Pred(20).

General Terms

Early stage software development effort estimation

Keywords

Software development, software project management, software development life cycle (SDLC), Entity Relationship Diagrams, evaluation criteria's.

1. INTRODUCTION

Software engineering is the science which provides a set of guidelines for on time and within budget software development. Software engineering has got a relevant activity as part of software project management and which is to estimate the how long it will take to develop a software application or product by the development team. Software effort estimation has been seen as one of the three great challenges for more than over 50 years old computer science [1]. Not one method, proposed over the years, could estimate the development effort to 100% accuracy and hence a combination of different methods & tools are used in arriving at the most appropriate development effort estimates. New soft computing based paradigms provide alternatives to carry out effort estimations vis-à-vis the more traditional algorithmic & statistical methods. Neural network has been found to be one of the best techniques for making software development effort estimations [2] and hence an attempt has been made in this paper to estimate the early stage development effort using student's project dataset. Numerous researchers and scientists have proposed different techniques on making effort estimations using Neural networks. Nasser Tadayon [3] developed an adaptive learning machine based on neural network to estimate the software cost using COCOMO model. But literature review reveals that not much have been done for the early stage effort estimations. This paper is an attempt towards this less explored area of effort estimations using soft computing techniques.

Early stage effort estimations can be defined as making software development effort estimations at the initial stages more precisely the Design stage of SDLC. Carrying out effort estimations at the early stages is beneficial because the design stage prediction implies fewer overheads at the later stages of software development. This paper provides an approach for carrying out early stage effort estimations using neural network models. The dataset used in this paper was developed based on the ER diagrams prepared by different B.Tech. in computer science & engineering degree students of Sikkim Manipal Institute of Technology, India as part of their Major Project work which spans 16 weeks. Total Count of Entities, Total Count of Attributes, Total Count of Relationships, Cumulative Grade Point Aggregate (CGPA) and Major Project final marks of students spanning different batches were gathered. The final marks obtained by students in the Major Project were used to obtain the recalculated effort in number of weeks of software development e.g. if a student scores 100 out of 100 marks then the development time assigned is highest i.e. 16 weeks. Other marks are suitably converted to number of week equivalent efforts. Now it is assumed that a student works for 5 days a week, then we can obtain the final software development effort by multiplying the number of weeks with 5 so as to obtain the number of day's values in the dataset. The number of days thus obtained forms the software development effort (in days) for individual instances and it is recorded in the dataset after due calculations.

1.1 Evaluation Criteria

Please There are many evaluation criteria to evaluate the accuracy of the software development effort in literature. The Mean Magnitude Relative error (MMRE) is a widely-accepted criterion in the literature and is based on the calculation of the magnitude relative error (MRE). Eq. (1) as below shows an equation for computing the MRE value that is used to assess the accuracies of the effort estimates. Here, the Y_j represents the actual effort and $\hat{Y}_j$ is the estimated effort for the project j.

$$MRE_j = \frac{|Y_j - \hat{Y}_j|}{Y_j} \qquad \text{Eq. (1)}$$

The MRE calculates each project in a dataset while the MMRE aggregates the multiple projects. The model with the lowest MMRE is considered the best [4]. As shown in Eq. (2), the estimation accuracy of the MMRE is the mean of all the MREs among n software projects. For example, an MMRE with a value of 0.5 means that the estimate matches 50% of the actual effort on average.

$$MMRE = \frac{1}{n}\sum_{j=1}^{n} MRE_j \qquad \text{Eq. (2)}$$

Another measure of Pred(l) was also adopted to evaluate the performance of the established software effort estimation models. This measure is also often used in the literature and is a proportion of a given level l in the accuracy. The equation for computing Pred(l) is shown in Eq. (3), where n is the total number of observations and k is the number of observations with an MRE less than or equal to l. An acceptable value for l is 0.20 in the literature, which is also adopted in this study.

$$\text{Pred (l)} = \frac{k}{n} \qquad \text{Eq. (3)}$$

Balance Relative Error (BRE) is yet another evaluation criteria used for estimation accuracy and is defined as given in Eq (4) below

$$BRE(\%) = 100 * \frac{|E-\hat{E}|}{\min(E,\hat{E})} \qquad \text{Eq (4)}$$

Where E = estimated effort and $\hat{E}$ = actual effort

Another often-used evaluation criterion is the median MRE (MdMRE). The MMRE is fairly conservative, with a bias against overestimates, while Pred(0.20) identifies the prediction systems that are generally accurate. The MdMRE is less sensitive to the extreme values compared to the MMRE [5]. It exhibits a similar pattern to MMRE but it is more likely to select the true model if the under-estimate is served. Either the MMRE or the MdMRE aggregates the multiple observations. In this study, the MdMRE, MMRE, BRE and Pred(0.20) were adopted as the indicators of the accuracy of the established software effort estimation models since they are the ones most widely used in the literature, thereby rendering our results more comparable to those of other work.

2. RESEARCH METHODOLOGY

2.1 Data Gathering & Dataset Generation

As described in the above sections the dataset was developed based on the ER diagrams developed by the engineering students as part of their Major Project of 16 weeks duration. The dataset was prepared based on a methodology proposed by the authors earlier [6]. From the dataset The Total Count of Entities, Total Count of Attributes, Total Count of Relationships, the Cumulative Grade Point Aggregate (CGPA) obtained by students were taken as the Input variables while the recalculated development effort in days is taken as the output variable for the sake of calculations based on different neural network models present in the Neural Network toolbox of Matlab 2007 b software.

The dataset prepared in the above manner is shown in Table 1 below. The dataset was prepared from the student's thesis records.

Serial Number	TCOE	TCOA	TCOR	CGPA	RDE
1	24	70	29	6.219	75
2	24	70	29	8.012	75
3	24	70	29	7.733	75
4	10	56	9	7.564	70
5	5	44	5	5.519	55
6	19	47	11	7.507	70
7	8	33	9	6.171	75
8	8	33	9	6.705	75
9	17	53	7	7.629	75
10	9	37	7	8.130	70
11	10	36	8	8.083	65
12	10	36	8	8.126	65
13	10	36	8	7.202	65
14	5	17	5	8.417	65
15	5	16	7	7.757	70
16	4	26	4	7.431	70
17	4	26	4	7.121	70
18	4	26	4	7.660	70
19	7	34	6	8.017	75

20	7	34	6	9.076	75
21	7	27	5	7.550	70
22	6	37	5	6.583	65
23	6	27	12	7.276	65
24	6	27	12	8.124	65
25	5	26	4	6.530	75
26	5	26	4	6.685	70
27	6	28	6	7.843	65
28	7	38	9	9.160	70
29	7	38	9	8.617	75
30	6	18	3	8.719	80
31	4	22	3	8.860	65
32	5	18	5	7.664	75
33	16	85	15	6.795	65
34	16	85	15	6.757	65
35	9	36	9	6.207	70
36	9	36	9	6.636	70
37	9	36	9	6.790	70
38	8	24	7	8.095	65
39	20	115	22	7.990	75
40	20	115	22	8.095	75
41	15	60	9	6.340	75

Table 1. Dataset & Attributes for Early Stage Development Effort Estimations, TCOE: Total count of entities, TCOA: total count of attributes, TCOR: total count of relationships, CGPA: cumulative Grade Point Aggregate (parameter for judging academic excellence of students), RDE: Recalculated Development Effort in number of days.

2.2 Discussion & Threat to validity

The dataset was generated using the student projects data and also the dataset is limited in the sense that mainly it captures the data pertaining to the application based software development. Also the real industry data will be very much different in terms of both the size of the ER designs & timeline of the project. Hence the same needs major tuning but none the less it is a very unique & novel approach which shows a way for carrying out the effort estimations at the early stages of software development.

2.3 Experimental Setup

Three different neural network models namely Feed Forward Back-propagation (FFBPNN), Cascade Forward Back-propagation (CFBPNN), Layer Recurrent neural network (LRNN) present in Neural Network Toolbox of Matlab 7.5 were used for the experimentation. In all the three cases, the first 30 records from the dataset were used to train the network while the last 11 records were used for testing the trained neural networks. In the dataset TCOE, TCOA, TCOR & CGPA were taken as input variables to the neural networks while RDE was considered as the output variable. Finally the results obtained upon simulation of neural nets were compared with the Actual Development (RDE) effort as present in the dataset.

3. EXPERIMENTAL RESULTS

Same training inputs, target output data and testing input data were fed to all the three neural networks and the new development effort values as obtained by each of the 3 neural network models are as shown in Table 2.

Serial No.	Actual RDE	RDE’ using FFBPNN	RDE’ using CascadeFBPNN	RDE’ using LRNN
31	65	69.39	79.71	79.73
32	75	67.73	66.26	69.17
33	65	79.03	55.06	80.00
34	65	79.03	55.05	80.00
35	70	55.00	77.46	69.11
36	70	55.21	74.66	69.39
37	70	60.07	72.86	69.44
38	65	58.85	62.28	67.77
39	75	79.16	61.54	68.31
40	75	79.16	64.05	70.04
41	75	79.20	55.14	55.06

Table 2. Development Effort as obtained by 3 different neural networks.

The MMRE, BRE and Pred (0.20) results obtained are as shown in Table 3.

Neural Network Model	Mean MMRE(%)	Mean BRE(%)	Pred (0.20) (%)
FFBPNN	12.96	14.39	63.63
CascadeFBPNN	13.59	15.69	63.63
LRNN	11.45	12.50	81.81

Table 3. Comparison of different neural networks based on evaluation criteria's.

From the table above we can infer that as per the Mean MRE (%) & Mean BRE (%) the Layer Recurrent Neural Network (LRNN) is best among the 3 models compared as it has the lowest MMRE and BRE values. Also Pred (0.20) criteria proves that the LRNN to be a better neural network model as it has got the highest value of 81.81.

4. CONCLUSION & FUTURE RESEARCH

This paper illustrated a novel approach towards software effort estimations at the early stages of its development using the dataset prepared from engineering students project data based on the ER diagrams. The results obtained using 3 different neural network models were compared and hence a way is shown based on which one can carry out the early stage software development efforts.

An ongoing research is related to applying fuzzy logic system, neuro fuzzy techniques to the dataset for early stage software effort estimations. Class diagrams and Use-case diagrams based dataset for early stage effort estimations can be prepared in a similar manner and soft computing tools and techniques can be applied to know their efficacy in effort estimations.

5. REFERENCES

[1] Brooks F. P., January 2003, "Three Great Challenges for Half century old computer science". Journal of the ACM, Vol. 50 No. 1 pp. 25-26.

[2] Karunanithi, N., et al. 1992 "Using neural networks in reliability prediction", IEEE Software, 53-59.

[3] Tadayon N. 2005 "Neural Network Approach for Software Cost Estimation". Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)

[4] Shepperd M., Schofield C., 1997 "Estimating software project effort using analogies", IEEE Transactions on Software Engineering 23 (12) 736–743.

[5] Foss T., Stensrud E., Kitchenham B., Myrtveit I., 2003 "A simulation study of the model evaluation criterion MMRE", IEEE Transactions on Software Engineering 29 (11).

[6] Bhatnagar R., Bhattacharjee V., Ghose M. K. 2010 "A Proposed Novel Framework for Early Effort Estimation using Fuzzy Logic Techniques", International in Global Journal of Computer Science and Technology, Vol. 10 Issue 14 (Ver. 1.0) 75 – 81