

A Study on the use of Artificial Neural Network Approach to Software Cost Estimation

Amit Shrivastava^{1*}, Dr. Rajeev Yadav²

¹ Research Scholar, Shri Krishna University, Chhatarpur M.P.

² Professor, Shri Krishna University, Chhatarpur M.P.

Abstract - Various cost estimating models are there, each of these models has advantages and disadvantages when it comes to predicting the price tag and time involved in creating a product. The estimate of software costs by use of Back-Propagation neural networks. The model is developed in a way that accommodates and enhances the commonly used COCOMO model. As a result, software cost estimates are more accurate, even when dealing with ambiguous or imprecise information. Three publicly available datasets are used to test the model. In this study, the software cost estimating expertise is modeled using an artificial neural network technique, and the findings are compared with the COCOMO model.

Keywords - Artificial Neural Network, Software, Cost Estimation, etc.

-----X-----

INTRODUCTION

Predicting the costs of applications correctly and reliably improves an organization's competitiveness, while still being incredibly beneficial to developers and consumers alike. Because they may assist to alleviate problems in the software development process as well as improve the scheduling, tracking, and monitoring of projects, techniques that provide improved estimations are an important consideration in software development. Product pricing models may be used in an increasingly dynamic and complicated environment to build successful software programs once they have been built. Contract agreements, project billing, task categorization and resource allocation, work development scheduling, worker supervision, and other resources may all be built on top of this paradigm. (1)

Metrics that are predicted to impact software development costs are difficult to detect, subjective, and subjective in their early phases. There is a theory here that if we are able to identify specific project parameters that have a significant impact on software cost development, we can produce accurate predictions. As a result, it is important to identify the software method's core properties, which lead to the creation of numerous computer models that may calculate or anticipate such variables as effort, efficiency, and productivity that affect this process. ISA (input sensitivity analysis) is used in this paper's research to properly estimate software development costs and determine the appropriate range of input parameters that more accurately reflect software

projects' expenses in early development cycles (SDLC). (2)

Cost-oriented estimations, direct effort calculations, or minimum models that highlight the link between time-consuming elements are some of the most used models for software cost evaluation. An example of a cost model having a major cost driver and complicated secondary factor modifications or drivers of performance, such as COCOMO, may be found. Several revisions to COCOMO II have been made since the original publication, including: three additional cost trends, each of which corresponds to a different stage in the product life cycle: An early design and a post-architecture composition of an application.

Software cost models take such error criteria into account, which is the most common method of matching predicted costs to actual effort. Because 75 percent of the projected values fall below 25 percent of the actual values, there are fundamental flaws in existing software cost models based on the results of the parameters. Accuracy, durability, objectivity, and homogeneity ratings must be achieved while also indicating which measures should be utilized as inputs into the cost model. (3) The use of parameters to improve estimates is controversial in most models, which is unfortunate. Measures might contain values predicted by project administrators or analysts, or they can be derived from past project outcomes, because the significance of some

measurements cannot be determined until the end of the project. There is a lack of publicly available, precise, and uniform metrics in the former instance. When compared to the later instance, which suffers from ambiguity. Only when a comparable measuring method is appropriately assembled under the same circumstances (same systems, technology, settings, people, and demands), can data sets that are homogeneously projected be made available. There are a few parts in the world where collaboration is fairly limited, whereas past cost calculation databases are limited and distributed due to a lengthy and costly data collection procedure.

Artificial Neural Networks

The artificial neural network (ANN) is an effective information processing system that resembles the properties of a biological neural network (ANN). An ANN has a wide range of neural processing components that are tightly interconnected. The relationship between each individual neuron and the rest of your brain, Input signal information is encoded in the weights of each communication connection. This information is utilized by the neuron network in order to resolve a specific problem. Each neuron has its own unique internal state. It's termed the neuron activation stage, and it occurs as a result of the neuron's inputs. Net input can take on a number of activation functions, such as Gaussian, Linear, Sigmoid, and Tanh. The Sigmoid function is the most often seen in neural networks. As a result, the ANN models are defined by three essential entities: (4)

- Synaptic links between the model;
- The teaching or learning guidelines for upgrades and weight adjustment;
- They are enabled.

The neural network method begins with the creation of the network configuration and the development of the network training methodology utilizing an established data collection. Architectures of the neural network are split into two groups: (5)

- Feed networks where no network route loops are present.
- Recursive feedback networks.

Software cost estimation models

COCOMO Model

Basic and medium models of the COCOMO model exist. Early and quick estimations of effort are captured in the basic model. Additional information will be provided as cost drivers for the intermediate and detailed models. An estimated KDSI app package size; a constant developer mode B-value (also known as a scaling factor; and 15 controls) are

the COCOMO model's inputs. Bio, half-developed and embedded device generation options are all included in the project design concept. Based on a criteria and how much it affects efficiency with the real number, each ranking is determined (effort multiplier). The following is the typical effort in person-months (PM) for intermediate COCOMO: (6)

$$Effort = A \times [Size]^B \times \prod_{i=1}^{15} EM_i$$

Constant A is related to as the coefficient of efficiency. By multiplying them together, effort multipliers are integrated in the effort estimation formula according to expenditure drivers.

COCOMO II Model

The COCOMO Model, developed in 1981 by Barry Boehm, is the most well-known example of an algorithmic cost model. 63 software initiatives were studied, and this helped the company's growth. Simple, intermediate, and sophisticated software cost estimating methods are all included in this hierarchy. One of the most often cited and practical cost estimation methods was this one. COCOMO II is an improved and modernized version of the first COCOMO edition. The influence of software modifications on the life cycle costs and timelines of an organization may be evaluated using a set of tools and procedures that are part of this process. In all, the structure has three sub-structures: They're as follows: (7)

- For quick-to-create applications, this approach makes use of interoperable components like GUI-based ones and relies on the most recent object stage.
- An early design model can be used for applications, system integration, or infrastructure development. Scale is calculated using feature points that haven't been altered (UFP).
- This model was utilized after the final project architecture had been established, making it the most detailed of the three models. Scale estimations may be made using role points or LOC in this model. We do genuine tech product development and upkeep in this department. (8)

COCOMO II defines 17 costs generators in the post-architecture model and 5 scale variables. The COCOMO II cost drivers are graded in a very low to an extra-large range. The product is used for nominal effort adjustment. (9)

Basic COCOMO

Details of project qualities for parameters of the simple COCOMO model equation really aren't needed.

$$\text{Person Month} = a (\text{KLOC})^b \quad (1)$$

$$\text{Development Time} = 2.5 * \text{PM}^c \quad (2)$$

There have been three modes of project development, depending on all three parameters: a, b and c.

Intermediate COCOMO model

The COCOMO form is the most prevalent variation. It has been shown that the researchers' intermediate COCOMO model is both more dependable than the basic version and comparable to the complete version. Effort and development should be measured using the following: (10)

$$E = a_i (\text{KLOC})^{b_i} * \text{EAF}$$

$$D = c_i (E)^{d_i}$$

Effort Adjusting Factor where EAF is concerned. Typical EAF values vary between 0.9 and 1.4. Table 1 displays the coefficients used. Table 2 shows the values of the EAF approximation multipliers. (11)

Table 1: Intermediate COCOMO coefficients

Project	a_i	b_i	c_i	d_i
Organic	3.2	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	2.8	1.2	2.5	0.32

Table 2: Cost Drives

Cost Drivers			
Product Attributes	Computer Attributes	Personnel Attributes	Project Attributes
RELY DATA CPLX	TIME STOR VIRT TURN	ACAP AEXP PCAP VEXP LEXP	MODP TOOLS SCED

Product attributes

- Necessary or better known as RELY program
- Database size or DATA
- Product difficulty or better known as CPLX 2.

Computer attributes

- Time limits or better known as TIME
- Main or better-defined storage restriction as STOR
- Virtual or best known as VIRT computer instability
- Timing or more known as TURN Computer (12)

Personnel attributes

- Capability of an Analyst or more known as ACAP
- Application or as AEXP Programmer Power or as PCAP
- Experience of virtual machinery or VEXP
- Language programming knowledge or LEXP.

Project attributes

- Modern or better regarded as MODP programming activities
- The use of software tools or TOOLS
- Necessary or best known as SCED development schedule.

OBJECTIVES OF THE STUDY

- The purpose of studying is software cost estimation model based on input selection and artificial neural networks.
- To investigate a methodology for estimating software costs based on a Functional Link Artificial Neural Network and an Improved Particle Optimization algorithm.
- To investigate a model for estimating software costs based on artificial neural networks and the Firefly algorithm.

RESEARCH METHODOLOGY

Hybrid of input selection system & artificial neural network model focused on software cost estimation

Better software cost estimating models may be convinced of the less important data collection. In this paragraph, the program cost estimate is

rendered by first carrying out a suggested input selection method to define and leave behind the irrelevant features the appropriate collection of cost drivers. The next move is to provide Artificial Neural Network with just these relevant characteristics, which are allocated as their feedback to accurately estimate the software development effort and costs. The elimination of irrelevant cost drivers in the very first phase results in reliable calculations of the cost of software. In comparison, the suggested model contributes to a substantial reduction in uncertainties relevant to software cost estimate models focused on standard neural artificial network. The issue of the detailed estimates of software costs is achieved in the proposed model through two steps.

Data Sets Used in The Experiment

Four databases from various organizations will be identified for assessment of the cost estimate of the software utilizing the recommended artificial neural network-based models. The thesis of Mair et al provided one of the data sets. 32 data groups in which only one data collection COCOMO 81 was chosen as a lone dataset comprising more than 50 software development projects became accessible for the public during this review. Furthermore, after inspecting the new literature, there have been three other public domain data pages, including USP05, Maxwell and CocNasa. A fixed collection of attributes is used in all data sets above:

Size attributes:

The scale of every software project is characteristics of this. Source Code Lines (SLOC), Function Points or some other metric can be provided the scale of the project. Size characteristics play an important role in measuring commitment.

Environment information attributes:

These qualifications are details about the production team, including information such as the number of developers and their respective development expertise. The details on the climate are often the company's information, including the industry.

Developmental attributes:

These attributes are the technological and managerial knowledge on the software development projects created. Management and technologies provide knowledge such as programming language or data base structures used to build these software projects.

RESULTS

A software cost estimating model based on a blend of input selection technique and artificial neural network model is presented in this part. A cost estimation model based on a functional link artificial

neural network and an enhanced particle-swarm optimization is presented in this section. Finally, the findings of a software cost estimate model based on an artificial neural network and the firefly method will be shown to round off this topic.

Using two datasets randomly chosen from the four described in the datasets subsection, the MRE value of an input selection procedure and an artificial neural network model hybrid-based software cost estimation model is calculated and then compared with the results obtained using the COCOMOII model, which is widely considered the basic software development project estimation model, and with another already existing model by. As can be seen in the following two tables, Tables 3 and 4, the suggested strategy has a significantly higher MRE than the two currently used methods:

Table 3 : Based on the COCOMO81 Dataset, the percentage of MRE (percent) of the Proposed Model and two existing techniques on 11 randomly selected projects

S No	Project ID	MRE (%) on COCOMO dataset		
		COCOMO II Model	B.T Rao, et al. Model	Proposed Model
01	05	7.44	5.23	3.44
02	12	19.83	9.18	7.12
03	30	6.49	3.21	2.22
04	38	50.98	14.40	11.10
05	40	12.40	4.89	3.89
06	45	5.35	3.69	2.72
07	47	16.40	7.60	4.07
08	59	8.66	3.20	2.97
09	61	13.10	8.61	4.12
10	62	6.22	3.30	2.88

Table 4: Ten randomly selected projects from the Cocnasa Dataset were used to test the proposed model and two other existing models.

S No	ProjectID	MRE (%) on Cocnasa dataset		
		COCOMOII Model	B.T Rao, et al.	ProposedModel
01	1	9.33	5.12	3.68
02	5	8.84	3.78	2.89
03	15	16.75	8.80	6.44
04	25	14.09	5.68	3.88
05	30	8.81	4.40	3.12
06	42	13.9	6.60	4.17
07	54	13.67	7.80	4.90
08	60	11.78	6.83	5.22
09	62	13.2	5.43	2.12
10	75	17.09	8.66	5.10

Median of MRE (MdMRE) is utilized as a second assessment criterion since Mean MRE is susceptible to outliers. Table 3 provides the

MdMRE for each of the three methods for the two datasets.

Table 5: Proposed Model's MdMRE on 11 randomly Selected COCOMO and CocNasa Dataset Projects, as well as two additional existing techniques

	MDMRE COMPARISON		
	COCOMO II MODEL	B.T RAO, ETAL. MODEL	PROPOSED MODEL
COCOMO DATASET (10 PROJECTS OF TABLE 1)	10.53	5.06	3.66
COCNESA DATASET (10 PROJECTS OF TABLE 2)	13.43	6.14	4.03

Using two datasets randomly chosen from the four described in the datasets subsection, the MRE value of an input selection procedure and an artificial neural network model hybrid-based software cost estimation model is calculated and then compared with the results obtained using the COCOMOII model, which is widely considered the basic software development project estimation model, and with another already existing model by. Tables indicate the significant difference in MRE between the suggested methodology and the two current techniques method.

In this section, we will discuss the findings of the suggested software cost estimate model based on the functional link artificial neural network (FLANN) and the improved particle swarm optimization (IPSO) method. Randomly selected projects from all four datasets are used to compute the proposed model's MRE, as shown in the datasets section. Following that, findings are compared to those of a previous model, the COCOMOII model, which is widely accepted as the primary estimating model in software development projects. The suggested model's MRE differs significantly from the two already-existing models:

Table 6: For the COCOMO81 dataset, MRE (percentage) of the Proposed Model and two other existing techniques on 11 randomly chosen projects.

MRE (%) on COCOMO dataset				
S. No	Project Ids	COCOMO II Model	Novel ANN Based SCE	Proposed Model
01	05	7.44	5.23	3.61
02	12	19.83	9.18	6.92
03	30	6.49	3.21	2.62
04	38	50.98	14.40	12.21
05	40	12.40	4.89	3.09
06	45	5.35	3.69	3.21
07	47	16.40	7.60	3.17
08	59	8.66	3.20	2.92
09	61	13.10	8.61	3.13
10	62	6.22	3.30	2.17
11	63	19.95	9.65	7.21

Table 7: Proposed Model vs. Other Two Existing Methods on 10 Randomly Selected Projects from the USP05 Dataset.

S No	ProjectIds	MRE (%) on USP05 dataset		
		COCOMOII Model	NovelANN based SCE	Proposed Model
01	3	12.56	7.24	3.80
02	6	10.14	6.44	3.29
03	10	13.70	6.22	4.45
04	12	12.90	7.62	3.77
05	14	7.23	3.16	2.12
06	23	12.90	6.67	5.13
07	29	9.66	5.44	4.49
08	37	8.78	6.20	5.21
09	41	13.76	5.44	4.70
10	45	15.89	3.96	3.88

Table 8: On 11 randomly selected projects from COCOMO Dataset and 10 randomly selected projects from the other three datasets, we ran the proposed model's MdMRE together with two other existing techniques.

	MdMRE Comparison		
	COCOMO II Model	Based SCE	Proposed Model
COCOMO Dataset (11 Projects of Table 2)	8.66	5.23	3.17
USP50 Dataset (10 Projects of Table 3)	12.73	6.21	4.16
Maxwell Dataset (10 Projects of Table 4)	8.74	5.31	3.09
Cocnasa Dataset (10 Projects of Table 5)	13.43	6.61	4.05

After implementing and testing the suggested software cost estimate model based on artificial neural network and firefly algorithm, two datasets, COCOMO81 and Cocnasa, were randomly selected to test the model. As a consequence, the suggested model's Magnitude of Relative Error values are compared to findings produced using

the COCOMOII estimate model, which is regarded to be the industry standard.

Both the COCOMO model and the proposed model are applied to each dataset to determine the MdMRE. Table 9 includes the MdMRE and a bar chart depicting its results.

Table 9: 10 COCOMO and Cocnasa Datasets Selected at Randomly for MdMRE of Proposed Model and COCOMO Model

	MdMRE Comparison	
	COCOMO II Model	Proposed Model
COCOMO Dataset	10.53	3.30
Cocnasa Dataset	13.43	3.84

CONCLUSION

Overview of software development effort estimation and a detailed literature review was conducted in order to determine the current state of the art in effort estimation. Clients of any software product are concerned with finding a facility that can satisfy their functional needs, is of the requisite quality, and can be provided within an acceptable time and cost limit. To undertake a benefit-cost analysis, acquire finance, and keep costs under control throughout the lifecycle of a software project, clients rely on cost estimates provided early in the development process. Early decisions in a software development project can consequently have far-reaching economic effects and can determine the future of any software development company. Different effort estimation methods were introduced through a categorization that split them into categories like: Model based, Expert based, Learning based, Dynamic, Regression and Composite techniques.

REFERENCES

1. Shaina Arora, Nidhi Mishra, "Software Cost Estimation using Single Layer Artificial Neural Network", International Journal of Advanced Engineering Research and Science (IJAERS) [Vol-4, Issue-9, Sep- 2017]
2. Papatheocharous, Efi&Andreou, Andreas. (2017). Software cost estimation using artificial neural networks with inputs selection. ICEIS 2007 - 9th International Conference on Enterprise Information Systems, Proceedings. 398-407.
3. Hamza H., Kamel, Shams A., K.: Software Effort Estimation Using Artificial Neural Networks A Survey of the Current Practices, Tenth International Conference on Information Technology: New Generations (ITNG), pp.731-733, 15-17, (2013).
4. Bayindir, Colak R., Sagiroglu I., Kahraman S., H.T.: Application of Adaptive Artificial Neural Network Method to Model the Excitation Currents of Synchronous Motors, 11th International Conference on in Machine Learning and Applications (ICMLA) , vol 2, pp.498-502, (2012).
5. Kaushik A., Chauhan A., Mittal D.: COCOMO Estimates Using Neural Networks, International Journal of Intelligent Systems and Applications (IJISA), vol 4, No.9, (2012)
6. Abbas Heiat, (2012) Comparison of Artificial Neural Network and Regression Models for Estimating Software Development Effort, Information and Software Technology, vol.44, pp.911-922, 2012.
7. AbeerHamdy, "Fuzzy Logic for Enhancing the Sensitivity of COCOMO Cost Model", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 9, pp. 1292-1297, Sep. 2012.
8. Sharma T.: A Comparative study of COCOMO II and Putnam models of Software Cost Estimation, International Journal of Scientific & Engineering Research , Issue 11, vol 2, (2011).
9. Y. Miyazaki et al., "Method to Estimate Parameter Values in Software Prediction Models," Information and Software Technology, vol. 33, no. 3, pp. 239-243, 2011.
10. F.S. Gharehchopogh, "Neural networks application in software cost estimation: A case study", IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 69-73, 15-18 June 2011.
11. Nasser Tadayon, "Neural Network Approach for Software Cost Estimation", IEEE proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '2010).
12. Attarzadeh, Siew Hock Ow, "A novel soft computing model to increase the accuracy of software development cost estimation", IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE), Vol. 3, pp. 603 - 607, 26-28 Feb. 2010.

Corresponding Author

Amit Shrivastava*

Research Scholar, Shri Krishna University, Chhatarpur M.P.