International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

PRESENTING A NEW MODEL FOR SOFTWARE COST ESTIMATION USING ARTIFICIAL NEURAL NETWORKS

S Aleem Basha

SCHOOL OF ENGINEERING, Enrollment No: SSSCSE1517,

P.hD Scholar, Computer Science & Engineering,

Sri Satya Sai University of technology & Medical Sciences, Sehore, Bhopal, (India)

ABSTRACT

The exactness of programming wander estimation, for instance, wander cost estimation, wander quality estimation and undertaking risk examination are basic issues in programming wander organization. The ability to definitely gage programming change costs is required by the assignment boss in masterminding and coordinating programming headway works out. Since programming effort drivers are questionable and vague, programming effort gages, especially before all else times of the progression life cycle. The examinations are every now and again the base correct, in light of the fact that by no detail is pondered the endeavor and the thing is to begin the new implementation. The prerequisite for tried and true and exact cost desires in programming outlining is an advancing test for programming engineers. In this paper a novel neural system Constructive Cost Model (COCOMO) is proposed for programming cost estimation. This model conveys a portion of the attractive highlights of neural systems approach, for example, learning capacity and great interpretability, while keeping up the benefits of the COCOMO demonstrate.

I.INTRODUCTION

The capacity to precisely and reliably appraise programming advancement endeavors, particularly in the beginning times of the improvement life cycle, is required by the task directors in arranging and leading programming improvement exercises on the grounds that the product value assurance, asset portion, plan game plan and process checking are needy upon it.

This issue lies in the way that product advancement is an unpredictable procedure because of the quantity of components included, including the human factor, the intricacy of the item that is created, the assortment of improvement stages, and the trouble of overseeing substantial undertakings.

As programming advancement has turned into a basic speculation for some associations, precise programming cost estimation models are expected to successfully foresee, screen, control and survey programming improvement. Since estimation exactness is generally influenced by displaying precision, discovering great models for programming estimation is currently a standout amongst the most vital destinations of the product designing group.

Among those techniques, COCOMO II (Constructive Cost Model) is the most normally utilized in light of its straightforwardness for assessing the exertion in person-month for a venture at various stages. In this paper, creators expect to utilize the idea of delicate figuring to build up another COCOMO II demonstrate in view of neural systems to accomplish precise programming cost estimation.

The process is composed as takes after. Area 2 and 3 portray the qualities of COCOMO II show and simulated neural systems ideas that will be utilized as a part of the proposed display. Segment 4 and 5 examine the related works and issue explanation that it is deduced from the related works. Area 6 and 7 shows another COCOMO II display in light of manufactured neural systems, and the preparation calculations actualized. Segment 8 examines the framework yield, results and breaking down of the outcomes.

II.NEED AND IMPORTANCE OF RESEARCH PROBLEM

Limitations of Existing System

- The estimates are often the least accurate, because very little detail is known about the project and the product at the beginning.
- It requires manual support for updating monthly reports.
- Least bother about the final results.
- Managing the results for the large systems may be not adequate.
- Not suggestible for large software systems due to less accuracy of results.

Proposed System

• The present model carries some of the desirable features of neural networks approach, such as learning ability and good interpretability, while maintaining the merits of the COCOMO model.

• The proposed model can be interpreted and validated by experts, and has good generalization capability.

• The model deals effectively with imprecise and uncertain input and enhances the reliability of software cost estimates.

• The concept of soft computing to develop a new COCOMO II model based on neural networks to achieve accurate software cost estimation.

III.TRAINING ALGORITHM

Utilizing this approach, we emphasize forward and in reverse until the point when the ending condition is fulfilled. This ending condition is the point at which all adjustments in weights are beneath some edge or a particular number of cycles have been finished. The preparation calculation is as indicated by underneath steps.

• Choose a preparation test and engender the info vector over the system to get the yield.

• Determine the blunder in yield, and the mistake inclination in the various layers.

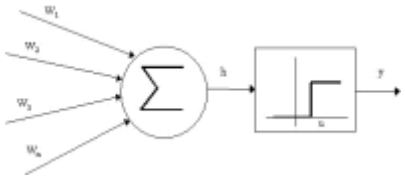
• Determine the parameter changes for the neural systems weights and refresh the neural systems weights

• Do again until the neural networks error is efficiently small after an epoch is complete.

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

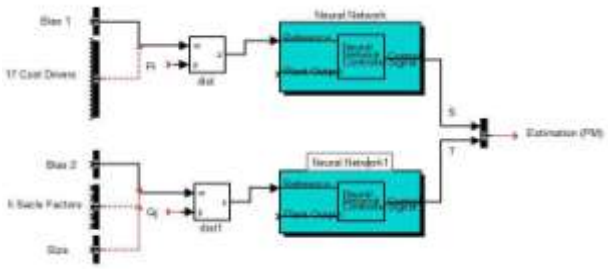
The learning rate that influences the adjustments in weights comparing to EM's and SF's has been subscripted with their related hubs.

IV.SIMPLIFIED MODEL FOR COST ESTIMATION



As has been expressed, a simulated neuron processes the weighted entirety of its N inputs, where j = 1,2,… … n, and creates a yield of 1 if this sum is above a certain limit u. Something else, an output x of 0 results.
It appears in below equation:

The proposed COCOMO II model based on neural networks



V.RESULTS

Investigations were finished by taking two datasets, initial one was unique information from COCOMO dataset and second one was manufactured dataset.

B.W. Boehm [1] is the principal scientist to take a gander at programming building from a monetary perspective, and he concocted cost estimation models from two datasets, COCOMO and COCOMO II. The COCOMO [2] dataset incorporates 63 recorded activities with 17 exertion drivers and one ward variable of the product advancement exertion. In this way, the main utilized dataset for assessing the proposed display depends on COCOMO demonstrate. The second endeavor was to make a counterfeit dataset, Table 1, in view of COCOMO model and preparing calculation that it clarified in past area.

TABLE I. The ARTIFICIAL dataset generated for system validation Consists of 100 DATA SAMPLES

No.	Mode	Size	Effort
1	1.1200	51.2500	246.5900
2	1.2000	12.5500	58.2800
3	1.0500	81.5200	550.4000
…	...	…	…
97	1.2000	56.5300	354.7300
98	1.0500	16.0400	67.1400
100	1.1200	54.1700	262.3800

The proposed neural networks model was validated by two approaches. In the first approach, has used the COCOMO dataset that comprises of 63 ventures (Dataset #1). In the second approach, has utilized the fake dataset that comprises of 100 specimen ventures (Dataset #2). At that point both datasets are connected to the new neural systems model and COCOMO II display. The approval of the new neural systems model to building prepared neural systems show for exertion estimation has been finished utilizing manufactured dataset and COCOMO dataset. The correlation between the aftereffects of COCOMO dataset and manufactured dataset that connected on the new neural systems model and COCOMO II display demonstrates more precision if there should be an occurrence of exertion estimation by the new neural systems show. The examinations between comes about are shown in Tables 2 and 3.

In this examination, each dataset independently connected to the COCOMO II show and proposed demonstrate. At that point for each model, the MMRE and Pred were computed. At long last mean of those estimations are utilized to think about the two models. The outcome for 163 connected activities demonstrates the MMRE for COOCMO II show is 0.531796411 and for proposed display the esteem equivalents to 0.440855158. It demonstrates the proposed show has MMRE not as much as COCOMO II display, so it implies the precision of proposed show is superior to COCOMO II. If there should be an occurrence of Pred , the last outcome demonstrates the proposed show esteem is 45% in Pred(25%) and COCOMO II esteem is 35% in same Pred. As it specified above, Pred demonstrates the quantity of tasks that they have MMRE young lady than 25%. As indicated by this definition, the proposed display demonstrates better precision. Table 3 demonstrates how much the proposed display is exact than COCOMO II show.

TABLE ACCURACY OF THE PROPOSED MODEL

Model	Evaluation	
		MMRE
Proposed Model vs. COCOMOII	COCOMO II	0.531796411
	Proposed Model	0.440856158
	Improvement %	17.1

For comparing proposed model with COCOMO model, the improvement is 17.1% based on the MMRE ~0.53 and ~0.44. The experimental results show that the proposed software effort estimation model shows better estimation accuracy than the other two models, i.e., COCOMO. In summary, an output with more terms or neural networks sets provided a better performance due to the high granularity demanded from the results. Most of the sample data in the dataset with the proposed neural networks model resulted in a more accurate estimation when compared to the COCOMO II model.

VI.TESTING

In software engineering, performance testing is processed to check the workload, usage of system, memory, processing, network and other system functionalities. It can also serve to investigate measure the program structure and its process activities inside the system.

S.No	Test cases	Pass/fail
1	Applying different values at different situation	pass
2	Random access of values may give wrong values	pass
3	Algorithm leads to different results if the given data is wrong	pass

VII.CONCLUSION

Software effort drivers usually have properties of uncertainty and vagueness when they are measured by human judgment. A software effort estimation model utilizing neural networks can overcome these characteristics of uncertainty and vagueness exist in software effort drivers.. Software effort estimation using neural networks is an attempt in the area of software project estimation. The objective of this work is to provide a technique for software cost estimation that performs better than other techniques on a given set of test cases. This work has shown by applying neural networks on the algorithmic and non-algorithmic software effort estimation models

accurate estimation is achievable. The proposed neural networks model showed better software effort estimates in view of the MMRE, Pred(0.25) evaluation criteria as compared to the traditional COCOMO. The above-mentioned results demonstrate that applying neural networks method to the software effort estimation is a feasible approach to addressing the problem of uncertainty and vagueness existed in software effort drivers. Furthermore, the neural networks model presents better estimation accuracy as compared to the COCOMO dataset. The utilization of neural networks for other applications in the software engineering field can also be explored in the future.

REFERENCES

[1]Boehm B. W. "Software Engineering Economics", Englewood Cliffs, NJ, Prentice-Hall, 1981.

[2]Boehm B., Abts, C., and Chulani, S., Software Development Cost Estimation Approaches – A Survey, University of Southern California Center for Software Engineering, Technical Reports, USC-CSE-2000-505, 2000.

[3]Putnam, L. H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering, Vol. 4, No. 4, pp. 345 – 361, 1978.

[4]Srinivasan, K. and Fisher D., "Machine Learning Approaches to Estimating Software Development Effort", IEEE Transactions on Software Engineering, Vol.21, No. 2, 1995.

[5]Molokken, K., Jorgensen, M., "A review of software surveys on software effort estimation", in Proceedings of IEEE International Symposium on Empirical Software Engineering, ISESE 2003, pp: 223 – 230, Oct. 2003.

Bibilography:

First Author:

S Aleem Basha'

SCHOOL OF ENGINEERING,

Enrollment No : SSSCSE1517,

P.hD Scholar, Computer Science & Engineering,

Sri Satya Sai University of technology & Medical Sciences, Sehore, Bhopal, India.

Mobile: 9642741435

Email id: aleemshaik02@gmail.com