Mathematical Statistician and Engineering Applications
ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

Software Quality Evaluation by Cocomo II With NN and SVM

Naveen Malik[#1], Sandip Kumar Goyal[#2]
[#1]Ph.D Scholar, CSE, MM Deemed To be University Mullana,Ambala,India
[#2]Professor, CSE, MM Deemed To be University Mullana,Ambala,India
Email-naveenmalik317@gmail.com
Email- skgmmec@gmail.com

Article Info
Page Number: 26 – 35
Publication Issue:
Vol 71 No. 1 (2022)

Article History
Article Received: *20 November 2021*
Revised: *25 November 2021*
Accepted: *09 December 2021*
Publication: *01 January 2022*

Abstract

Cost, time and quality projection are the crucial aspects in software development process. Incorrect estimations can cause losses which in turn may lead to irreversible damage. It is generally perceived that a imperfectly estimated project always results in a substandard quality due to the efforts being wrongly directed. Firstly Effort Estimation is calculated by actual effort and proposed Effort. That Effort evaluation of 500 NASA projects, after that evaluation is done by four parameters Standard Error, Standard Deviation, Mean Absolute Error, Root Mean Square Error. The author amalgamated the robustness of COCOMO-II with that of Neural Network NN and Support Vector Machine SVM .Quality Which we evaluate that is quality Evaluation of Semantic Web Application. In the last checks the majority of all four parameters for software quality assessment.

Keywords: Software quality, Neural Network, SVM, COCOMO II, Semantic Web..

I. INTRODUCTION

Software development process involves many attributes such as testability, effort estimation, accuracy and usability etc. The effort estimation grabs our attention very easily. The current era is bestowed with a lot of technical developments that leads to competition amongst everything born out of these developments. This paves the way for effort estimation to proffer extremely vigorous results with high facets of reliability and accuracy before kicking
off the project [1, 2]. The evolution in software project is gingerly and a definite forecast is not viable in factual sense [3, 4]. It is foremost to recognize the issues regarding such type of prognosis that might wind up in overestimation or underestimation for the efforts.

Estimation the effort is multifarious task and the estimation prototypes handling such jobs are categorized as algorithm grounded COCOMO model, non-algorithm grounded trained model and also the
models that take on the robustness of machine learning architectonics [5, 6]. Countless effort estimation prototypes developed with the thriving necessity but these could not touch the perfection ceiling yet. PRICE-S model by Park [7], COCOMO by Boehm [8], Putnam and Myers SLIM [9] and Function Point developed by Albrecht [10] hold the lion's share amongst the algorithmic prototypes.

The models pose some restraints in carrying off the faultless estimation due to the requirement of input attributes in view of Line Of Code, complexities that could not be squarely attained at preliminary steps of development process. This induces the incapacity of the models to put forward solutions to resist composite relationships, graded data along with intense deficiency in interpretation potentiality [11]. In this study, COCOMO is utilized in integration with artificial bee colony algorithm to provide more vigorous effort approximation in respect of both sensitivity and accuracy. Furthermore, the paper is organized in the following segments:

Segment I: Sets forth the quality evaluation and the bit part of effort approximation in quality evaluation
Segment II: Talks over the COCOMO substructure
Segment III: Narrates the proposed work procedure and the algorithm employed
Segment IV: Mentions the formula and evaluation attributes
Segment V: Brings up the results
Segment VI: Winds up the paper
Segment VII: Quotes the reference work

II. QUALITY EVALUATION AND EFFORT ESTIMATION

The quality of software is dependent on numerous facets inclusive of the used effort in the route of software development. Productivity is related to the skill of the people deployed to the correct product. Productivity will be lesser and efforts will be higher when the people are not skill oriented. Researchers have always been interested in the quality evaluation and this work intends to classify the quality on the basis of effort approximation.

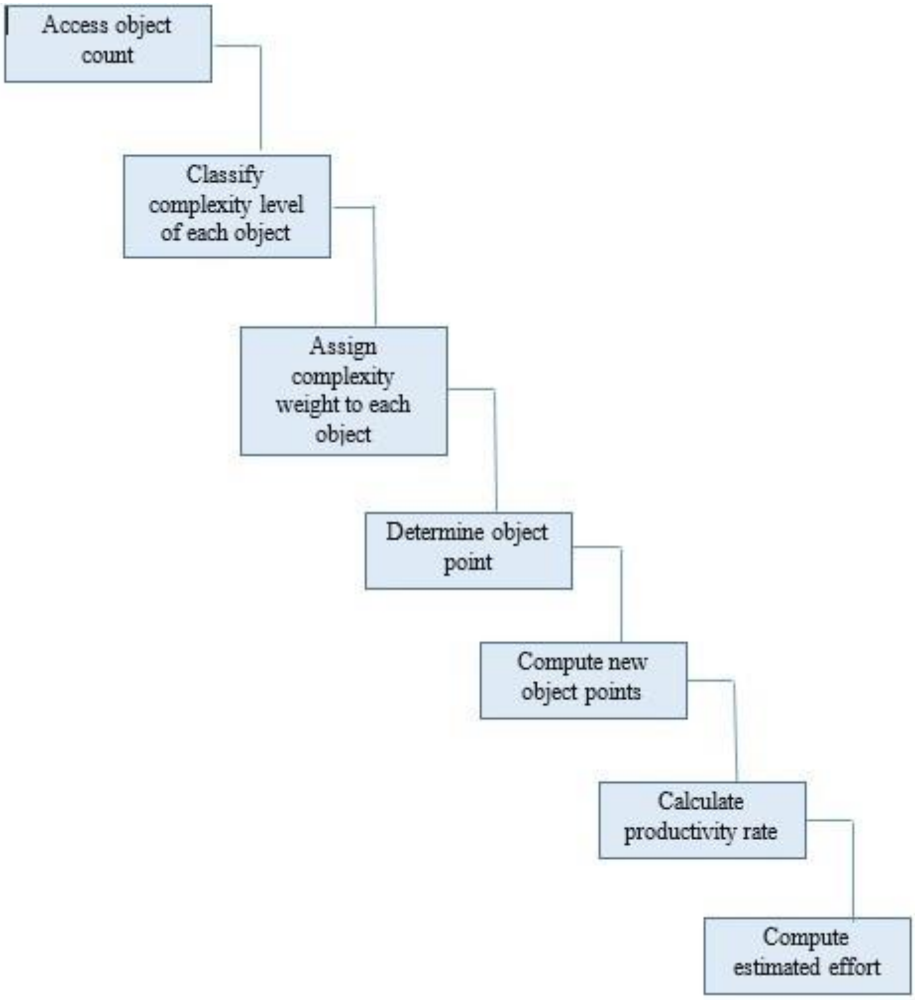

Fig. 1.Software objects count

III. COCOMO SUBSTRUCTURE

Boehm developed the COCOMO. It provides algorithm induced cost approximation. Regression formula is the base for the estimation operability in this model. Feature and historical data of the active project lead to input attributes for the model. The model operates in three modes:

- Organic mode
- Semi-detached mode
- Embedded mode

Organic mode revolves around the working of simple projects with teams working in a well explanatory and firm situations. The teams involved in semi-detached mode manifest diverse knowledge. Embedded mode employs strict checks to match the differing requirements. The basic effort is approximated as follows:

$$E_{months} = p_1 * (KLOC)\, p_2 \qquad (1)$$

A. COCOMO-II PROTOTYPE

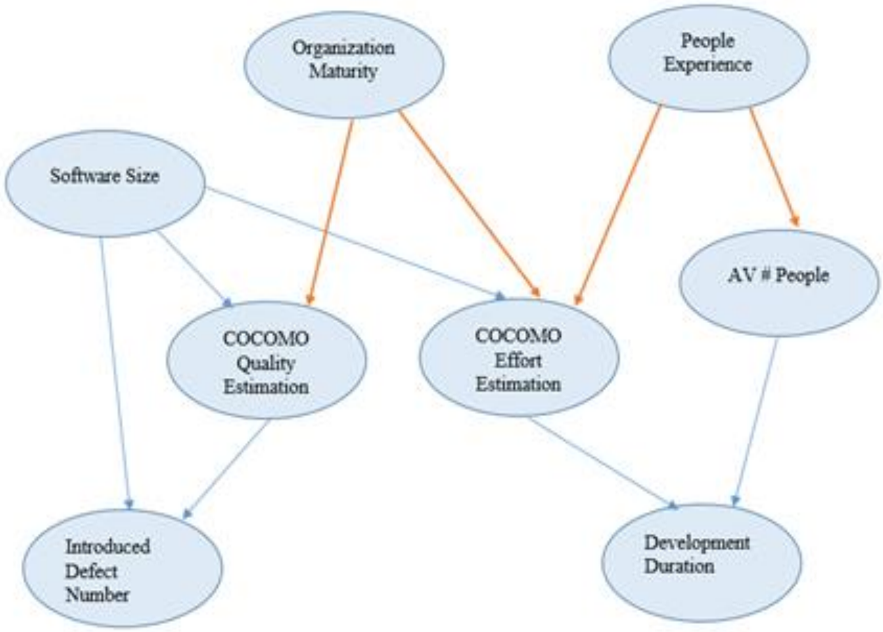

Fig.2. COCOMO II Prototype

COCOMO-II is a mathematical and idolized solution to evaluate the effort on a project. The following mathematical parameters have been utilized as:

i. Required Software Reliability (RELY)

It's an evaluation of the threshold of the extent up to which software must perform

ii. Data Base Size Data

It is a measure of data requirement for the proceeding iii. Iii. iii. Product Complexity (CPLX)

The product complexity is divided into five sections:

a) Control Operations CPLX
b) Computation Operation CPLX
c) Device Dependent CPLX

ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

d) Data Management CPLX

e) User Interface CPLX

iv. Required Reusability (RUSE)

This is the required extra effort to complete the project.

v. Execution Time (TIME)

Total execution time of the project is calculated by the execution time

vi. Storage Constraints (STOR)

Total amount of storage required for the project canbe termed as shown

vii. Programmer Capability (PCAP)

It is a parameter which is affected by the programmer's capability. It is influenced by the way a programmer's capability.

viii. Language and Tool Experience (LTEX

Regression Analysis is the basis of this estimation model. The prototype manifests the architecture as mentioned below [12-13]:

- Application Composition model functions on the speculation that the reclaimable attributes such as scripts and record encodings blueprint the fundamental architecture of any system. It is based on the consideration of the sample efforts to cope with issues concerning user interface, performance, software and system relationship. Effort estimation is accomplished in the initial stages and the size is estimated on the basis of application or the object points such as screens and project reports etc.
- Initial Design Model focuses on the forecast of the duration and the intricate cost of the project earlier than the complete design could Size is calculated by utilizing the unadjusted function points in conjunction with prediction equations.
- Post Architecture Prototype revolves around the bonafide design and prolongation of the software to accomplish the correct prediction of the size of the product. This model is cost efficacious when system risk, task and perceptions are under consideration. LOC or the function points are used for size forecast.

IV. PROPOSED WORK

The proposed work used the data set of 500 projects in the NASA data sets. In the proposed model, COCOMO-II received the feed from ABC in the form of attributive values. Moreover, NN was used for classification amongst the values before being fed to COCOMO-II. Support parameter is accessed by Effort. There is no rule if there is high effort that means quality is high or if there is less effort that means there is no quality low of that software.

In below table we prescribed the range of 4 cost drivers which are used in our proposed model. 18 cost drivers used in post architecture model. These cost drivers are compatible with Reliability, Programmers capability, Required Reusability, Language and Tool experience of the architecture model. All the Effort Estimation And Quality Evaluation process have been done in MATLAB Machine. Firstly Effort Difference Estimation is find by the difference of Actual Effort and Proposed Effort. In the last We apply the Rule Set for Quality Evaluation.

ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

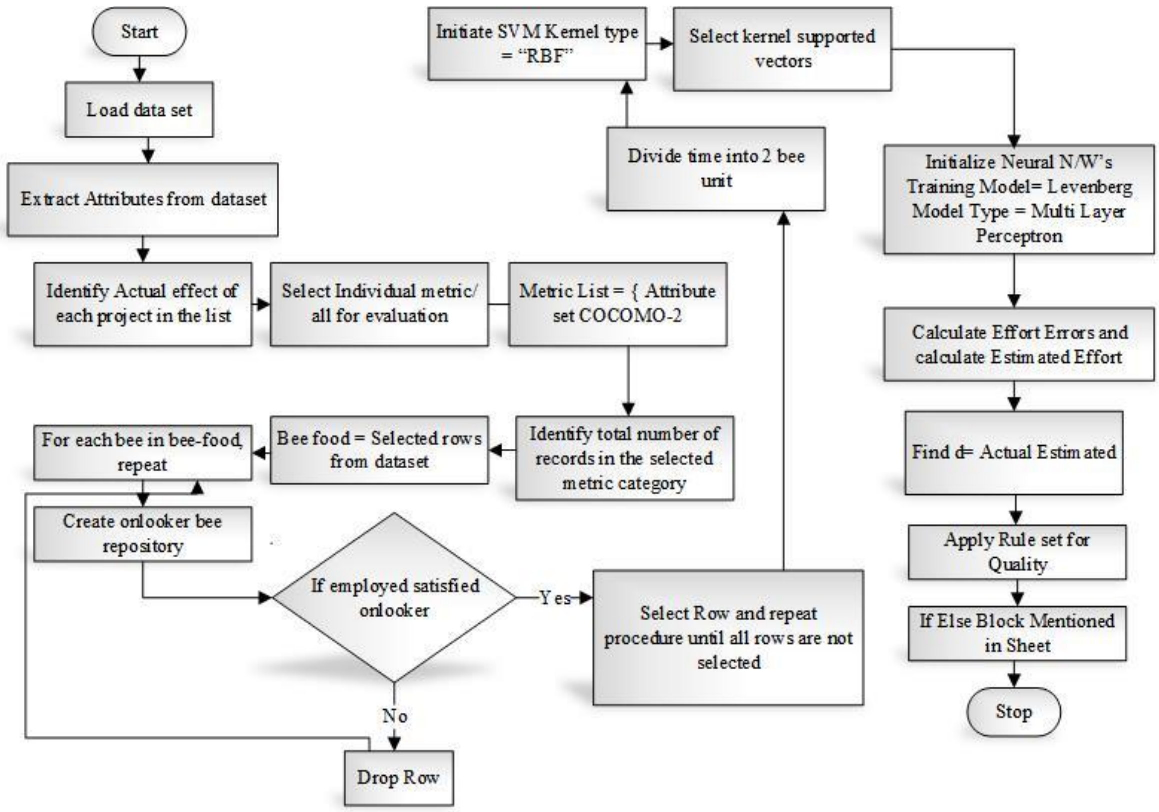

Fig.3 Proposed Methodology

V. RESULTS

In this section comparison of the effort estimation obtained using proposed model is compared with the actual effort and the effort obtained from COCOMO-II and Neural Network and SVM [16]. Effort Is estimated of 500 ISRO projects, firstly we evaluate 100 software quality and take their mean value, Same for next 100 software, difference of actual effort and proposed effort is effort estimation.

Below all these tables qaulity evaluation by four different parameters Standard Error, Standard Deviation, Mean Absolute Error, Root Mean Square Error. All four parameters evaluate different quality evaluation. All mean values are taking of 100-100 data sets for calculating their mean values. In The Below Tables Tp Neural is denoted by True Positive of Neural, Tp SVM is denoted by True positive SVM, Tp COCOMO II is denoted by True Positive of COCOMO II Model. Same as Tp Phenomenon we used Fp For False positive for all models. .After the effort estimation on MATLAB, we find quality evaluation by four parameters SE, SD, MAE, RMSE. In the last we find out accuracy of particular software by the Recall and Precision of COCOMO II, NN, SVM. We apply the Rule Set

TABLE I.

Number of Project files	tp Neural	tp SVM	tp COCOMO 2	fp Neural	fp SVM	fp COCOMO2	tp Proposed	fp Proposed
100	**0.74**	**0.6798**	**0.61452**	**0.26**	**0.3202**	**0.364545**	**0.7896**	**0.10256**
200	**0.7321**	**0.6685**	**0.62348**	**0.2679**	**0.3315**	**0.37652**	**0.745512**	**0.107741**
300	**0.77145**	**0.6785**	**0.66324**	**0.22855**	**0.3215**	**0.33676**	**0.7965**	**0.108963**
400	**0.72145**	**0.69325**	**0.701458**	**0.27855**	**0.30675**	**0.298542**	**0.78552**	**0.1569**
500	**0.71456**	**0.699741**	**0.700145**	**0.28544**	**0.300259**	**0.299855**	**0.75689**	**0.14125**

Mathematical Statistician and Engineering Applications
ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

TABLE II.

Number of Project files	Precision Neural	Precision SVM	Precision COCOMO2	Recall Neural	Recall SVM	Recall COCOMO2	Precision Proposed	Recall Proposed
100	**0.74**	**0.6798**	**0.627660063**	**0.87634116**	**0.771623156**	**0.719477357**	**0.885043**	**0.88539**
200	**0.7321**	**0.6685**	**0.62348**	**0.87414925**	**0.765750286**	**0.721052875**	**0.873729**	**0.872748**
300	**0.77145**	**0.6785**	**0.66324**	**0.86190716**	**0.746014294**	**0.76794998**	**0.87966**	**0.883018**
400	**0.72145**	**0.69325**	**0.701458**	**0.87068339**	**0.748367248**	**0.748574795**	**0.833514**	**0.884515**
500	**0.71456**	**0.699741**	**0.700145**	**0.85253412**	**0.765756065**	**0.773610965**	**0.842731**	**0.881068**

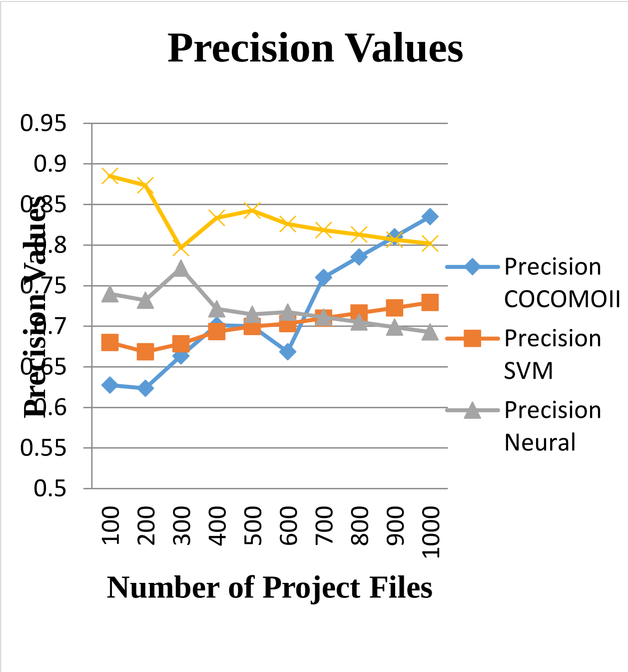

Fig. 4. Precision Values

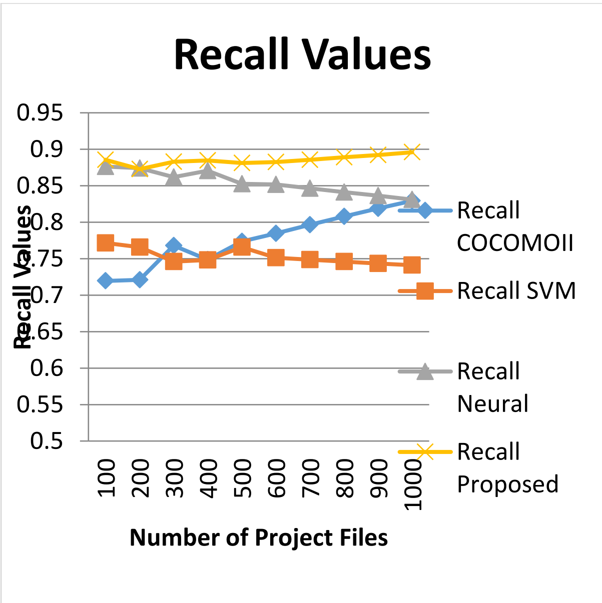

Fig.5. Recall Values

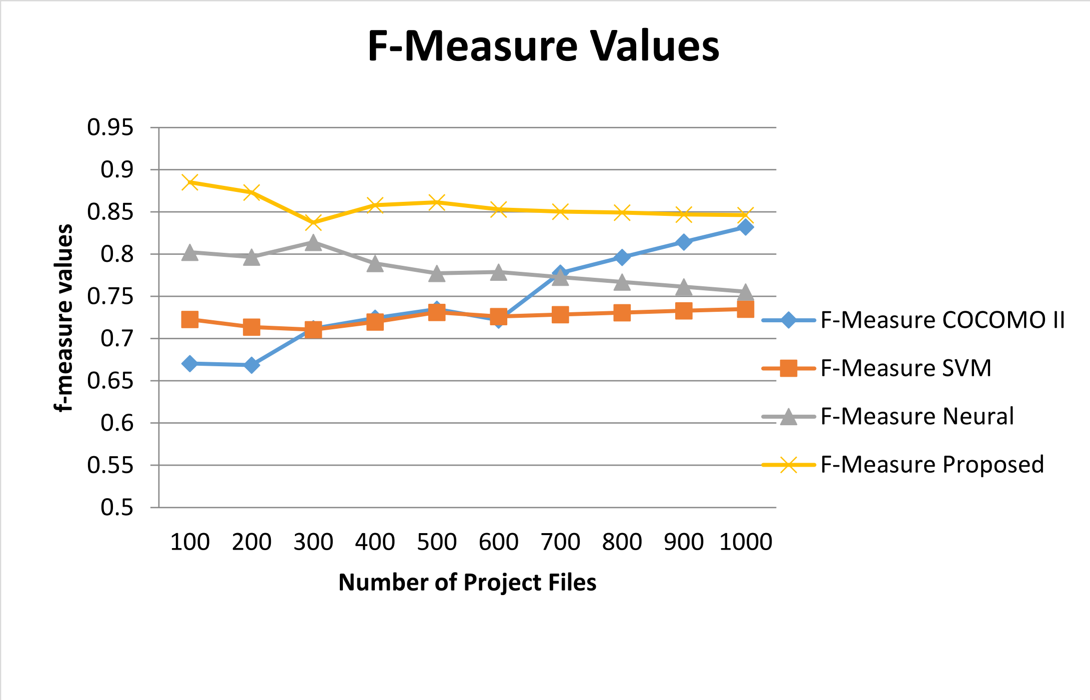

Fig.6. F-Measure Values

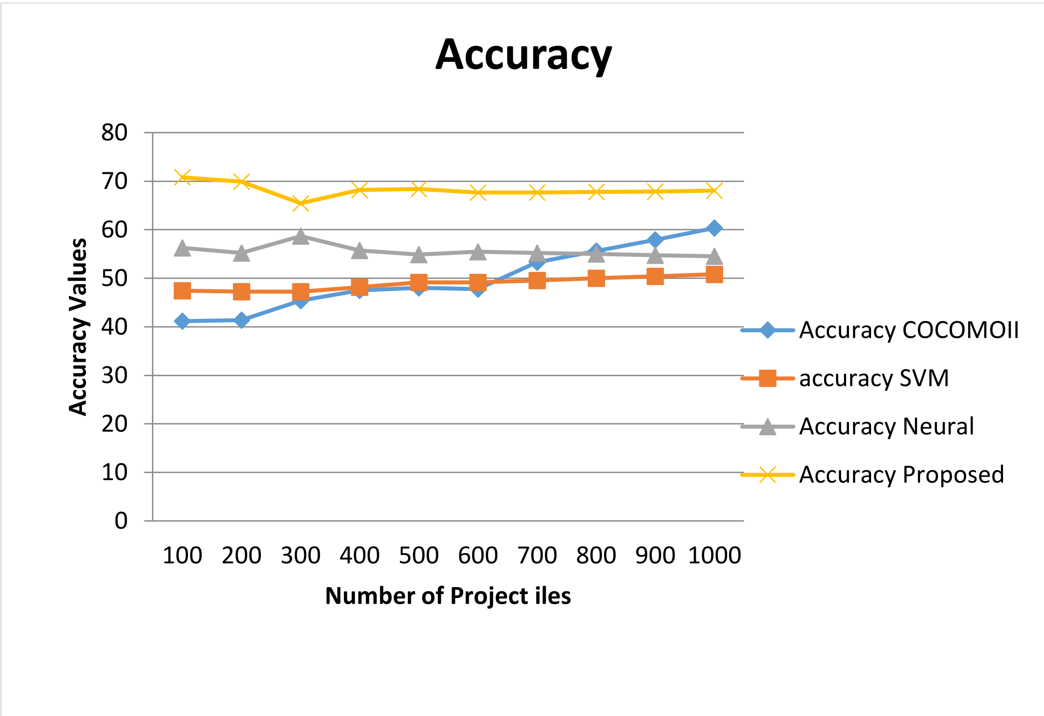

Fig.7. Accuracy

ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

TABLE IV.

Number of Project files	f-measure Neural	f-measure SVM	f-measure COCOMO2	f-measure Proposed	Accuracy Neural	Accuracy Proposed	accuracy SVM	Accuracy COCOMO2
100	**0.8024203**	**0.722807018**	**0.670439699**	**0.88521668**	**56.27718797**	**70.79265**	**47.45218**	**41.15691**
200	**0.7968435**	**0.713828083**	**0.668726001**	**0.8732384**	**55.2299879**	**69.90051**	**47.24882**	**41.38297**
300	**0.8141738**	**0.71065724**	**0.711764548**	**0.88133609**	**58.67700078**	**70.96243**	**47.26675**	**45.419**
400	**0.7890727**	**0.719754977**	**0.7242509**	**0.85825731**	**59.63914782**	**68.16442**	**48.15407**	**47.50012**
500	**0.7774731**	**0.731261669**	**0.735046863**	**0.8614728**	**62.7831374**	**68.38358**	**49.13635**	**48.01631**

VI. CONCLUSION

Over the time, efforts and quality estimation are being done by various authors to overcome the degradation of the developed software. It is really difficult to obtain a satisfactory prediction well before the starting of the project as on the way project gets evolved and influence by numerous factors on the go. In the present work, authors tried to reach a near ideal effort estimation with the combination of ABC algorithm to search for the produced quality at the end. The solution obtained from ABC is then fed to NN for classification and training to enhance the quality of results. Finally, COCOMO-II is employed to predict the effort estimation parameters. After our proposed work we can says that if any Project quality is degraded on the basis of SE,SD, MAE, RMSE that means quality of software is degraded. We checks the majority of all parameters that is shown in below graphs of F-measure and Accuracy Measure.

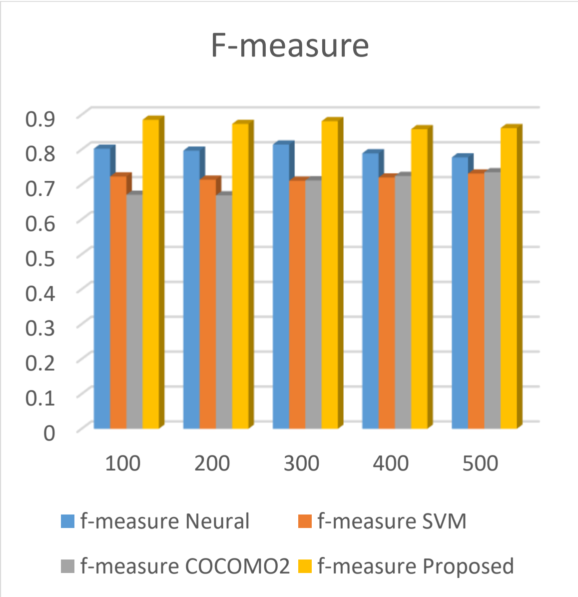

Fig. 8.F-Measure

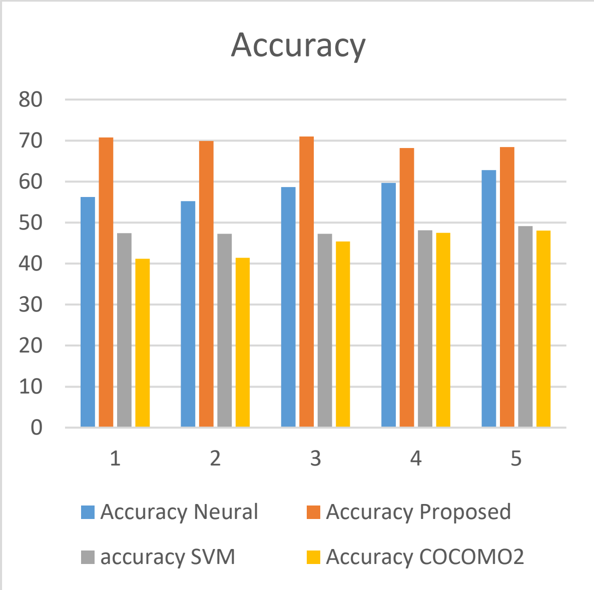

Fig. 9.Accuracy

References

1. Naveen Malik, Sandip Kumar Goyal, Vinisha Malik, "A New Pattern for Software Engineering as a Service " in international journal –" IJSDR ",Volume 4 , Issue 6, pages 408-412 , ISSN:2455-2631 (2017).
2. Gharehchopogh, F. S., Talebi, A., & Maleki, I. Analysis of use case points models for software cost estimation. International journal of academic Research, 6(3) (2014).
3. Maleki, I., Ebrahimi, L., & Gharehchopogh, F. S. A hybrid approach of firefly and genetic algorithms in software cost estimation. MAGNT Research Report, 2(6), 372-388 (2014).
4. Jorgensen, M., & Shepperd, M. A systematic review of software development cost estimation studies. IEEE Transactions on software engineering, 33(1), 33-53 (2006).
5. Srichandan, S. A new approach of software effort estimation using radial basis function neural networks. International Journal on Advanced Computer Theory and Engineering (IJACTE), 1(1), 113-120 (2012).
6. Barry, B. Software engineering economics. New York, 197-207 (1981). Park, R. E. Parametric Software Cost Estimation with an Adaptable Model. AACE International Transactions, G-11.
7. Attarzadeh, I., & Ow, S. H. Software development effort estimation based on a new fuzzy logic model. International Journal of Computer Theory and Engineering, 1(4), 473 (1988).
8. Vinisha Malik, Sandip kumar goyal & Naveen Malik , A Hybrid Model For android Malware Detection, scopus journal International journal of Innovative Technology and Exploring Engineering. IJITEE (Vol. 8, Issue 12, pp 2656-2662) (2019).
9. Saliu, M. O., Ahmed, M., & AlGhamdi, J. Towards adaptive soft computing based software effort prediction. In IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS'04. (Vol. 1, pp. 16-21). IEEE (2004).
10. Boehm B.W, B. Clark, E. Horwitz, R. Madachy, C. Abts, S.Chulani, A.W.Brown and B. Steece,"COCOMO II model definition manual, University of South California Center for Software Engineering, (2000).
11. Gao, W., & Liu, S. Improved artificial bee colony algorithm for global optimization. Information Processing Letters, 111(17), 871-882. (2011)
12. Naveen Malik, Ontology based test case reused- A Semantic Web Technique National Conference RAPS- 2014 ,PEC University Of Technology , Chandigarh (2014).

ISSN: 2326-9865
https://doi.org/10.17762/msea.v71i1.34

13. Naveen Malik, Sandip Kumar Goyal, Vinisha Malik, “Quality Assessment Of Semantic Web Based Applications and Saas” presented in ISPCC- 4th IEEE international conference JUIT solan and published in IEEE Xplore (2017).
14. Yadav, R. K., & Niranjan, S. Optimized model for software effort estimation using cocomo-2 metrics with fuzzy logic. International Journal of Advanced Research in Computer Science, 8 (2017).