Software Cost Estimation using Hybrid Algorithm

Shivani Sharma[1], Aman Kaushik[2], Abhishek Tomar[3]
[1]*Research Scholar,* [2,3]*Assistant Professor*
[1,2,3]*Department of computer science, Baddi University of Emerging Sciences and Technology, Baddi, Solan H.P -173205, India*

Abstract: *Software cost estimation is the vital step to start any project. It gives us the outline of effort, resources and time required for a project. Accomplishment of software enhancement depends on cost estimation. It is really tough to meet approximate cost with actual cost. Such as Software size has remained highest significant factor in software which is increasing day by day, due to which we have realized a huge amount of increase in complexity as well as in size of software. A project will be enabled a success if all the necessities can be fulfilled, the cost is not excessive, did not pass through the strategy that has been planned. There are various budget assessment techniques to compute cost of the development and Function point analysis (FPA) is the technique of calculating the dimension of software. Thebenefit is that it can avoid source code error when selecting dissimilar programming languages.The key objectives of this study we are computing budget of project based on Top down method in which we will compute function points of each module. The whole process will be done by Ant colony optimization algorithm. To compare and evaluate the outcomes of the proposed algorithm with K Modes algorithm and RF model and it has been noticed that when we have compared with K modes and RF model then proposed work gives better results.*

Keywords: *Software cost estimation, Cost estimation methods, Ant colony optimization, K modes, and RF model.*

I. INTRODUCTION

The growing requirement for application software's and the growing size and difficulty of these software's, requires an appropriate paradigm for assessing the budget of software developments in software production organizations [1]. Estimating the budget of software developments plays asignificantpart in organization productivity. In a sense that incorrect estimates can cause enormous financial losses and in various circumstances can indication to total project failure. [2] Therefore, the correctness of estimated cost is of great importance. But concerning the innovative and conceptual nature of software developments, the calculation of cost and time develops very difficult [3]. The process of assessing the budget of software developments is extremely important, so that before early to produce and develop a software project every association first deals with estimation of creature resources and available facilities [4].

Software developers always interest to know the time estimation of software tasks. It could be done by comparing similar tasks that have already been developed. Although, approximating task has an unclear nature, as it depends on numerous and usually not clear factors then it is hard to be displayed mathematically. Software plan and cost approximation supports the development and tracking of software developments. Effectively monitoring the costly investment of software expansion is of high importance. The reliable and correct cost assessment in software engineering is a continuing development due to which it allows for extensive financial and strategic planning [5].

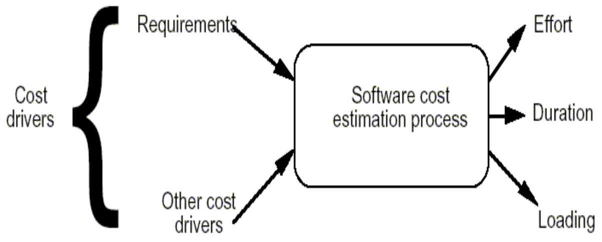


Fig.1 software cost estimation process

Software cost assessment has expandedremarkablesignificance in the last two decades due to its vital necessity for efficient effort assessment in software analysis.The software estimate process includes assessing the measurement of the software product to produced, assessing the effort required, developedinitial project plans, and finally analyzing overall cost of the project [6].

Accurate cost estimation is important because of the following reasons.

- It be able to determine what resources to commit to the project and how well these resources will be used.
- It be able touse to assess the impact of changes and upkeep preplanning.
- Projects can be at ease to manage and control when resources are wellco-ordinated to real needs.
- Customers believe actual development costs to be in line with predictable costs.

We have organized the paper as follows: Section 2, Cost estimation methods. Section 3 describes the related study by various researcher in this field, Section 4 describes Problem formulation and objectives, Section 5 describes Proposed Methodology. In the last conclusion of paper has described in section 6.

II. SOFTWARE COST ESTIMATION METHODS

Cost estimation techniques are mainly of two kinds: algorithmic and non-algorithmic. Algorithmic technique use a formula to calculate the software cost estimation. The formula is developed from models which are produced by combining related cost factors. In addition, the statistical process is used for model construction. Non-algorithmic technique do not use a formula to calculate the software cost estimate.

A) *Non Algorithmic Based Estimation Methods*

i) Expert Judgment Method

Expert judgment techniques includeaccessing with software cost estimation expert or a group of the experts to use their knowledge and understanding of the proposed project to reach at an estimate of its cost. It is the supreme usable technique for the software cost assessment. Mostly companies used this method for generating the cost of the product. Normally a group consensus technique, Delphi technique, is the finest way to be used.

ii) Estimating by Analogy

Estimating by analogy means matching the proposed project to previously accomplished similar project where the project development information id well-known. Real data from the finished projects are generalized to estimate the proposed project. This technique can be used either at system-level or at the component-level.

iii) Top-Down Estimating Method

Top-down estimating technique is too called Macro Model. By top-down estimating technique, acomplete cost assessment for the project is derived from the global properties of the software project, and then the project is divided into many low-level mechanism or modules. The leading technique using this method is Putnam model. This technique is more applicable to early budget estimation when only global properties are identified.

iv) Bottom-up Estimating Method

Using bottom-up estimating technique, the cost of each software modules is assessed and then combines the outcomes to reach at an estimated cost of complete project. It goals at making the estimate of a system from the knowledge collected about the small software modules and their interfaces. The leading technique using this approach is COCOMO's detailed model.

B) Algorithmic Method

The algorithmic technique is considered to offer some mathematical calculations to achieve software estimation. These mathematical calculations are established on research and historical data and use inputs such as Source Lines of Code (SLOC), amount of functions to accomplish, and other cost drivers such as language, design method, skill-levels, risk calculations, etc.

i) Function Point Estimation

Albrecht [7], [8] presented the model of function points as a software size measurement while considering the more general problem of defining application development productivity. His objective was to improve a software size measure that was self-determining of the implementation technology. To measure efficiency, a size of effort product (or output) must be clear. For this Albrecht selected the function value to be provided to the user. To calculate the function value distributed to the user, the amount of inputs, outputs, inquiries, and files (including interfaces to other programs) from the user viewpoint are calculated, weighted, and summed. In October, 1979, IBM and IBM user group SHARE and GUIDE had a meeting in Monterey, California.

During the meeting Allan Albrecht gave a presentation on Function Point metric. Meanwhile, IBM announced the basic function point metrics. Function Point Investigation is a software assessment system that oriented on function, which processes the size of system mainly from its functionality and usability. In its view, system involves of the following 5 characteristics [9]

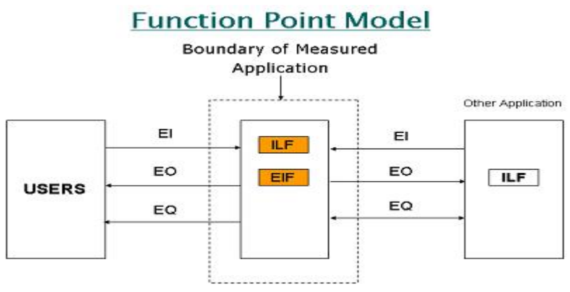


Fig. 2 Function point Model

a) ***Function Types Are As***

- **EI (External Input):**These are end-user actions such as putting in a login or executing a mouse click.
- **EO (External Output):** The system offers the end-user output or interface such as a GUI display.
- **EQ (External Query):** This function is initiated by the end-user.

- **ILF (Internal Logical File):** These files are the master or transaction files that the system interacts with during its session.
- **EIF (External Interface File):** Distinct logical internal files, where the application uses only for its purpose, these files are shared with other applications.

III. LITERATURE REVIEW

Assessing the budget of software developments has long been attention of many researchers, in a way that the model-based methods from the late 1970s continued by demonstration of the models such as: SLIM by Putnam and Myers in 1992[12], Checkpoint model by Jones in 1997[10], PRICE-S model by Park in 1988[11], and COCOMO by Boehm in 1981[3]. Sometime after the development of algorithm models, numerous findings have been accompanied to use non-algorithm techniques such as machine learning approaches as an alternative to model-based techniques. Certain of these papers are mentioned briefly.

Low and Jeffery (1990) have considered that the Function point counts appear to be a more consistent a priori measure of software size than source lines of code. Understanding in the application of function points is a significant factor in their successful application. [13]

Mukhopadhyay and Kekre (1992) have evaluated that the utmost of the previous work in software effort assessment has put heavy importance on LOC or other design attributes as primary variables for cost models. The estimated size be able touse in combination with existing models to generate early estimate of development effort. [14]

In 1994 Matson et al., have introducedtheir research presents an assessment of several published statistical regression models that relate software development effort to software size measured in function points [15].

In 1997 Vijayakumar, S. have evaluated, the research was initiated to recognize and examine the variables which influence the activities that organise software development and which conclude the cost of software. [16].

In 2003 Jorgensen, M., and K. Molokken-Ostvold explored, their research summarizes estimation knowledge through anexploration of studies on software effort assessment [17].

In 2004 Z. Xu, T.M. Khoshgoftaar have designed, Fuzzy logic-based cost assessment models are furthersuitable when vague and imprecise information is to be accounted for [18].

In 2009 Zheng, Y., et al, have discussed Scale increasing in applications and a variation of programming languages using at the similar time, manual measurement based on the LOC (Line of Code) cannot meet the estimating requirements.The emergence of Function Point resolves these difficult issues [19].

Jeng, Bingchiang, et al. (2011) have offeredmethod redefines the function type classifications in the FPA model, on the base of the target application’s characteristics and system architecture. By the support of this technique can makes the function types appropriate for specific application area [20].

Attarzadeh et al. (2012) have explored Accurate and reliable software project estimates such as time, cost, and manpower in the primary part of software development, is some of the crucial objectives in software project organization [21].

In 2012 Malathi and Sridhar have explored Estimation of effort in software budget founded on Fuzzy Analogy is one of the supreme popular existing methods [22].

In 2015 P. S. Bishnu, V. Bhattacherjee have introduced**, “**Software cost assessment created on modified K-Modes clustering Algorithm. The purposes of this research are: first, the modified K-Modes clustering which is an improvement over the simple K-Modes algorithm using a suitable difference measure for mixed data types, is offered and another, the proposed K-Modes algorithm is useful for software budget estimation [23].

In the year of 2015 Moeyersoms, Julie, et al. have explored software fault and effort calculation are significant tasks to reduce budgets of a software project. In software effort calculation the goal is to assessment the effort necessary to complete a software project, while software fault control tries to recognize fault-prone modules. [24]

IV. PROBLEM FORMULATION

In this paper we are computing the cost estimation of the project based on Top down approach in which we will compute function points of each module. The whole process will be done by using Ant colony optimization algorithm. In this paper we are computing the three factors which is Recall, Accuracy and duration. The duration factor is basically run time of algorithm or can say execution time of algorithm. And compare these factors with K modes algorithm and random forests (RF Model).In existing research work Random forest model they have computed the effort prediction and the faults prediction based on that they computed the cost of project. Whereas we have also worked on k modes algorithm to compute the cost estimation using function points methods but here is a drawback and that is it takes more time to calculate the function points of project but using that algorithm we cannot recognize highly accurate errors in project which lead to compute not proper cost estimation. Due to predict wrong fault estimation we compute the wrong estimation man hour.To overcome this in our research work we are calculating the cost of project based on Top down method in which we will compute function points of each module. The whole process will be done by Ant colony optimization algorithm.

A) Objectives

- To develop an algorithm to compute highly correct cost estimation based on highly estimated accuracy.
- To develop a highly optimized algorithm to compute the cost of project based on function point in less time.
- Compare our research (ACO) with RF model and k modes algorithm
- To test the accuracy, recall and duration based on analysis of results.

B) Estimation Challenges

- Estimates are treated as ACTUALS and treated as COMMITMENT.
- Estimates don't change when the requirement changes. Variation in the requirements / technology or lack of scope clarity at the time of estimation.
- Absence of expert on a particular package to convey out the estimation or faulting estimating.
- Approved change requests.

V. PROPOSED METHODOLOGY

In this paper we are calculating the cost estimation of the project based on Top down method in which we will compute function points of each module. The whole process will be done by Ant colony optimization algorithm. In this proposed work we are computing the three factors which is Recall, Accuracy and duration and compare these factors with K modes algorithm and random forests (RF Model).It has been noticed that when we compared with K modes and RF model then proposed work gives better results.

A) Ant Colony Optimization

ACO is one of the utmost known meta-heuristic algorithms. ACO algorithm was first presented by Dorigo in 1996. ACO algorithm is motivated from the natural life of the ants. Ants leave an odorous stuff on the path named pheromone. This stuff evaporates but is left in short time as the ant trace on the earth. The ants are capable to produce pheromone to discovery the nearest path to the food. The ants selecting the nearest path create more pheromone than the ones selecting the longer paths. As the more pheromone attracts further ants, the more and the more ants will select the nearer path and then all ants will discover the nearer path and move on it.

The common algorithm is quite simple and created on a set of ants, each building one of the likely round-trips along the cities.Now-a-days, a numeral of algorithms motivated by the foraging behaviour of ant colonies have been useful to resolvehard discrete optimization problems. In fact, ACO algorithm is the utmost effective and broadly recognized algorithm based on the ant behaviour. [25, 26]

i) Applications of the Ant Colony Optimization

- Routing in telecommunication networks
- Traveling Salesman
- Graph Colouring Scheduling
- Constraint Satisfaction

ii) Benefits of the Ant Colony Optimization

- Inherent parallelism
- Positive Response accounts for quickfind of good solutions
- Be able to be used in dynamic applications.

iii) Drawbacks of the Ant Colony Optimization

- Series of random decisions.
- Research is investigational rather than theoretical
- Time to convergence uncertain (but convergence is guaranteed!)
- Theoretical analysis is tough
- Probability distribution changes by iteration

iv) Algorithm Steps for Count Function Points Using Ant Colony Optimization

Length=vector array of Files; counter=0; i=0;

Step 1: Traverse from the first index of array (counter) using i Update visited node status for current node from 0 to 1.

Step 2: Set Dr is number of function exist in i^{th} index of class then Apply for each (k=0 k< Dr) find the attributes in respected function, sum=sum+attributes.

Step 3: Update Pheromone stack of the selected function point get extracted from stack then after that we will empty that stack.

Step 4: k=k++

Step 5: If i<=length set visit status (v) of all node to 0, sum = 0, set current node = "start" and go to step 2.

Step 6: End of algorithm.

FLOW CHART

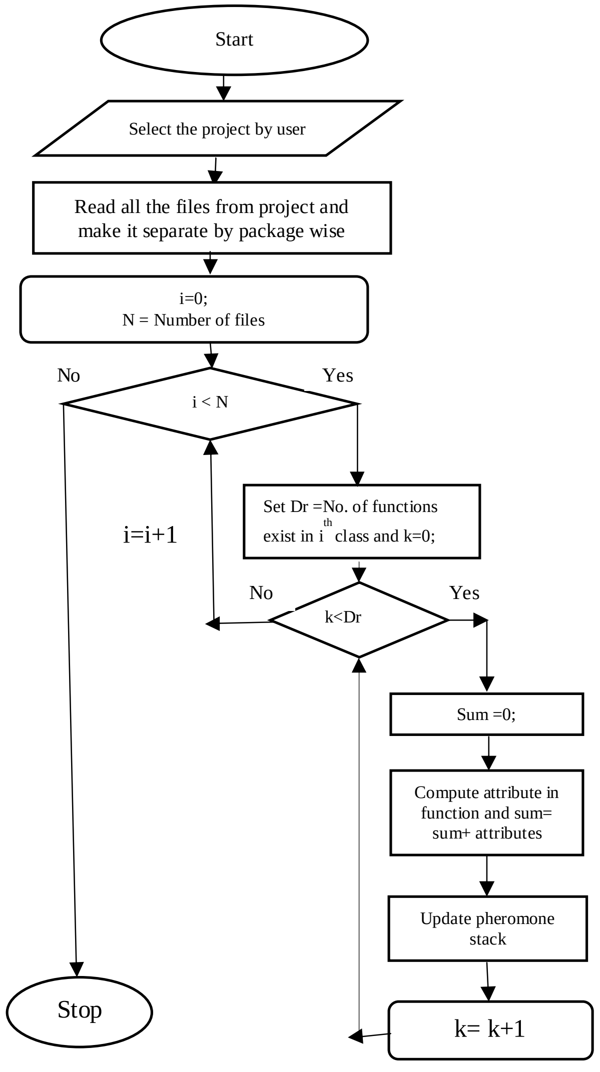

Start
Select the project by user
Read all the files from project and make it separate by package wise
i=0;
N = Number of files
No
Yes
i < N
i=i+1
Set Dr =No. of functions exist in ith class and k=0;
No
Yes
k<Dr
Sum =0;
Compute attribute in function and sum= sum+ attributes
Update pheromone stack
k= k+1
Stop

Fig.3 flow chart of count function points via ant colony optimization.

v) Algorithm Steps for Errors Prediction Using Ant Colony Optimization

Length =vector array of bugs; Counter=0;

Step 1: Traverse from the first index of array (counter) Update visited node status for current node from 0 to 1.

Step 2: If the current node is Decision node then Flag=truethen Apply For each (i=0 i< length of Vector file) Find the error in each file and compute its sum, sum=sum +error exist in file.

Step 3: Update Pheromone stack of the selected path values get extracted from stack after that we will empty that stack.

Step 4: Set count = count ++

Step 5: If count<=length set visit status (v) of all node to 0, sum = 0, set current node = "start" and go to step 2

Step 6: End of Algorithm.

FLOW CHART

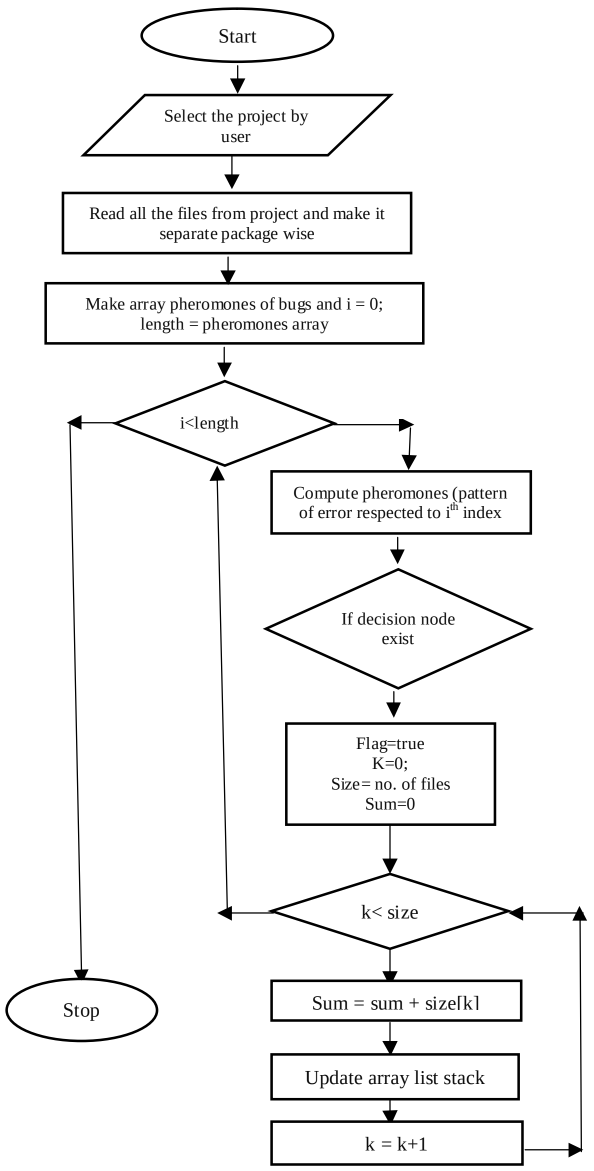

Start
Select the project by user
Read all the files from project and make it separate package wise
Make array pheromones of bugs and i = 0;
length = pheromones array
i<length
Compute pheromones (pattern of error respected to ith index
If decision node exist
Flag=true
K=0;
Size= no. of files
Sum=0
k< size
Stop
Sum = sum + size[k]
Update array list stack
k = k+1

Fig. 4 flow chart of errors prediction via ant colony

We have Compared Our Proposed Algorithm (ACO) With K Modes Algorithm and RF Model Algorithm. We have compared three parameters Recall, Accuracy and execution time of algorithm with another two algorithms.

B) K Modes

K-modes is one of the easiest unsupervised learning algorithms that resolve the well-known clustering problem. The process monitors a simple and informal method to categorize a specified data set by a certain amount of clusters (assume k clusters) fixed a priori. The key notion is to describe k centroids, one for each cluster. These centroids must be located in a cunning way as of dissimilar location causes different result. So, the superior choice is to place them as much as likely extreme left from each other. The next step is to proceeds each point fit in to a given data set and associate it to the nearest centroid. When no point is left, the first phase is finished and an initial groupage is done. At this time we want to re-calculate k new centroids as barycentre's of the clusters resulting from the earlier step.

After we have these k new centroids, a new binding has to be done among the equal data set points and the nearest new centroid. A loop has been created. As an outcome of this loop we may observed that the k centroids change their location step by step until no more modifications are done. In other words centroids do not change any more. Finally, this algorithm goals at reducing an *objective function*, in this case a squared error function. [27]

i) K-Modes Advantages

- There are continuously K clusters.
- There is constantly at least one item in every cluster.
- If variables are enormous, then K-Means most of the times computationally earlier.

ii) K-Modes Drawbacks

- Different early partitions can outcome in different final clusters.
- It does not work well with clusters (in the original data) of Different size and Different density

iii) Algorithm Steps for K Modes Algorithm

Let X = {x1,x2,x3,……..,xn} be the set of files and V = {v1,v2,…….,vc} be the set of bugs.

Step1: First set k=0, t=0;

Step 2: For each while i <x and for each k not equal to size D where D is number of function in each class.

Step 3: Calculate the function point of function then k=k+1.

Step 4: Sum=sum+Fp the data point to the cluster center (File).

Step 5: If i>x length return false.

Step 6: Otherwise go to step 2.

Step 7: Set r= V length.

Step 8: For each i=0 < r

Step 9: Scan the bug in all files if exist

Step 10: Set status=true and calculate the bug count.

Step 11: Then r=r+1 and go to step 9.

Step 12: If r > v length stop.

C) Random Forest Model

It develops lots of decision tree based on random selection of data and random selection of variables it provides the class of dependent based on many trees. As the trees are based on random selection of data as well as variables, these are random tree. Many random tree leads to a random forest.

Many random decision tree leads to a random forest. Random forests are a grouping of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same spreading for all trees in the forest. The simplification error for forests meets as to a limit as the number of trees in the forest develops large.

i) There are two main beliefs

- Maximum of the tree can offer correct prediction of class for most part of the data decision trees are usually correct. It is only some fraction some part of data where it goes wrong.
- The trees are making mistake at different places. All the trees are making mistake not at similar place.[28]

ii) Random Forest Model Benefits

- The Random Forests algorithm is a good algorithm to usage for difficult classification tasks.
- The main benefit of a Random Forests is that the model created can easily be interrupted.

iii) Disadvantages

The main limitation of the Random Forests algorithm is that a large number of trees may make the algorithm slow for real-time prediction.

iv) AlgorithmSteps for RF Model (Random Forest)

Step 1: Create a node for each class t for the tree.

Step 2: t=0 and D=number of function in 1 class;
If all functions values are computed in one function then t will get incremented by 1.

Step 3: If t>D return the single-node tree t.

Step 4: Otherwise Let sum= be the function point from Attributes if there is a user interface attribute then Set A with constant value 1.

Step 5: Add a new tree branch (new file) below t, corresponding to the test A* = "a". And then apply the same procedure from 2 point.

Step 6: Once whole tree parts traversed the return false support vector machine.

Step 1: Set i=0.

Step 2: For each i \in \{1….., n\} where n is the number of files in project.

Step 3: The x= {bug1, bug2, bug3}

Step 4: Take first bug1 and scan in whole project if exist

Step 5: Flag=1.

Step 6: Else i=i+1.

Step 7: If i>n return.

Firstly we have count function points by ACO (Ant Colony Algorithm, K Modes and RF model. The function point will remain same for all three algorithms. We have count six projects function points in which we will count function point for each module.

TABLE 1. Showing count function point results via ant colony (ACO), k modes and RF model (random forest).

FUNCTION POINT

Sr. No.	ACO (Proposed Algorithm)	K MODES	RF MODEL
P1	2599		
P2	7509		
P3	277		
P4	832		
P5	57		
P6	3112		

This table 1 showing count function point results via ant colony(ACO), K Modes, RF model(random Forest). We have taken six projects and count function point for individual project. Function point will remain same for ACO, K MODES and RF MODEL.

D) Parameters for Comparison of Results

The following Factors are used for comparison of our results with the other approaches. We have computed function points, recall value, accuracy and execution time of algorithm via ACO and compared with K modes and RF.

i) Precision

In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query:

$$\text{Precision} = \frac{\text{Relevant documents} \cap \text{Retrieved documents}}{\text{Retrieved documents}}$$

For instance for a text search on a set of documents precision is the number of correct results divided by the number of all returned results. Precision is also used by recall, the percent of all relevant documents that is revered by the search. [29]

ii) Recall

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall} = \frac{\text{Relevant documents} \cap \text{Retrieved documents}}{\text{Relevant documents}}$$

For example for text exploration on a set of documents recall is the number of accurate results divided by the number of results that should have been returned.[29]

Iii) Accuracy

- The accuracy computes some average of the information retrieved from precision and recall.
- The below formula shows the relation necessary to compute the accuracy.[29]

$$\text{Accuracy} = 2\times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

iv)*Duration*

- This duration parameter is run/execution time of algorithm in which computing the start time and end time of algorithm.
    - Start time=System. Current time Millis ();
    - End time= System. Current time Millis () +20000;
    - End time – start time /1000;
- In this formula converting the mille seconds in to seconds and computing the execution time of algorithm. It is also describing how much time occupied by algorithm to calculate function points.

E) Simulation Environment

i) Tools Used

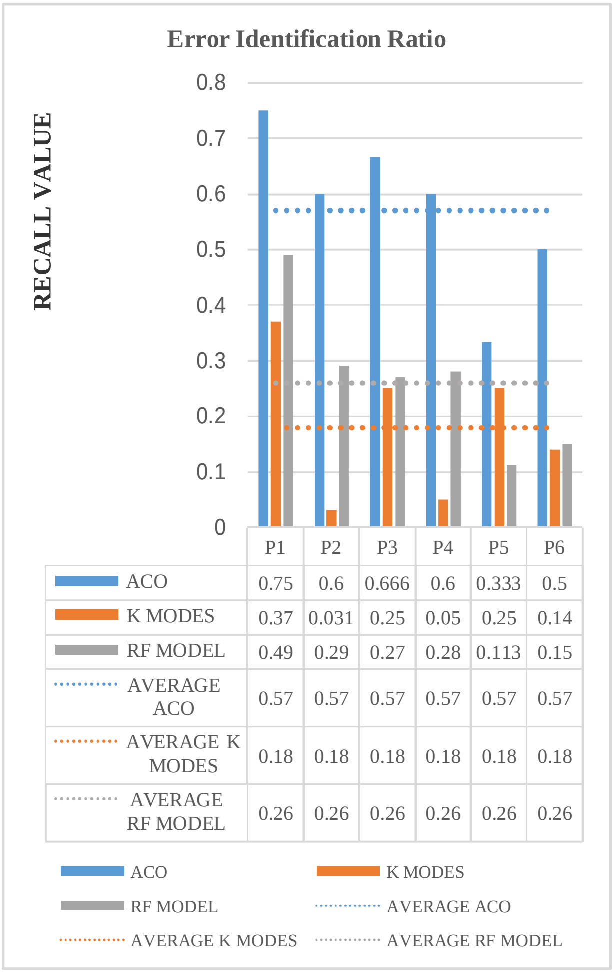
In the current scenario for the implementation of proposed methodology Net Beans version 8.1 is used. Net Beans is a software development platform written in Java. The Net Beans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the Net Beans Platform, including the Net Beans integrated development environment (IDE).The Net Beans Team actively support the product and seek feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback

The short list below enumerates some of its most prevalent features

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)

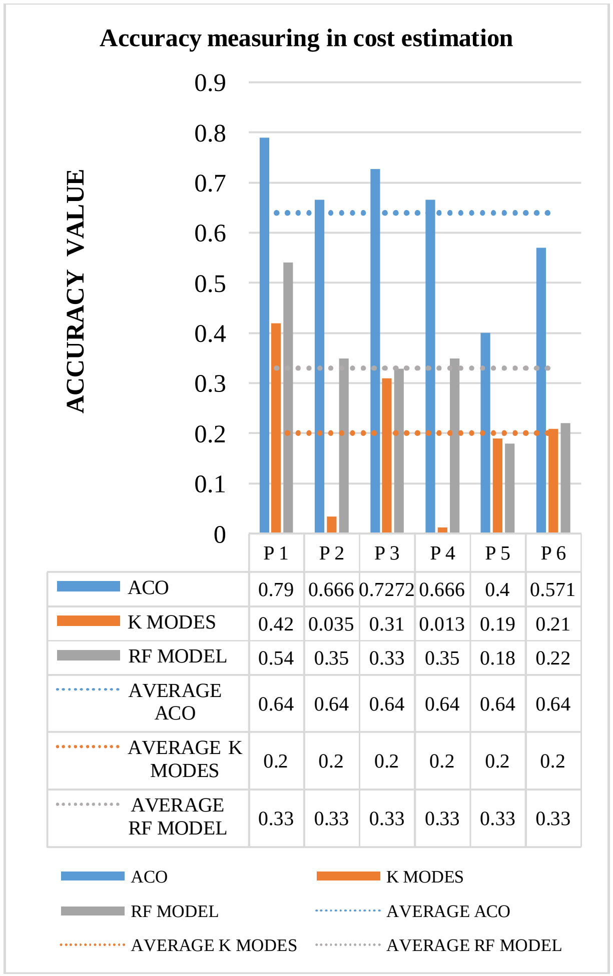
F) Results

We have plotted the result of six projects basis on Recall, Accuracy and duration parameters. Duration is basically run /execution time of algorithm. It has been observed that when we compared with K modes and RF model then proposed work gives better results. Here we have combined all projects in to one chart in which showing comparison of results for ACO, K modes and RF Model.



	P1	P2	P3	P4	P5	P6
ACO	0.75	0.6	0.666	0.6	0.333	0.5
K MODES	0.37	0.031	0.25	0.05	0.25	0.14
RF MODEL	0.49	0.29	0.27	0.28	0.113	0.15
AVERAGE ACO	0.57	0.57	0.57	0.57	0.57	0.57
AVERAGE K MODES	0.18	0.18	0.18	0.18	0.18	0.18
AVERAGE RF MODEL	0.26	0.26	0.26	0.26	0.26	0.26

Fig.5 chart showing the recall value for ACO, K Modes and RF Model for six project

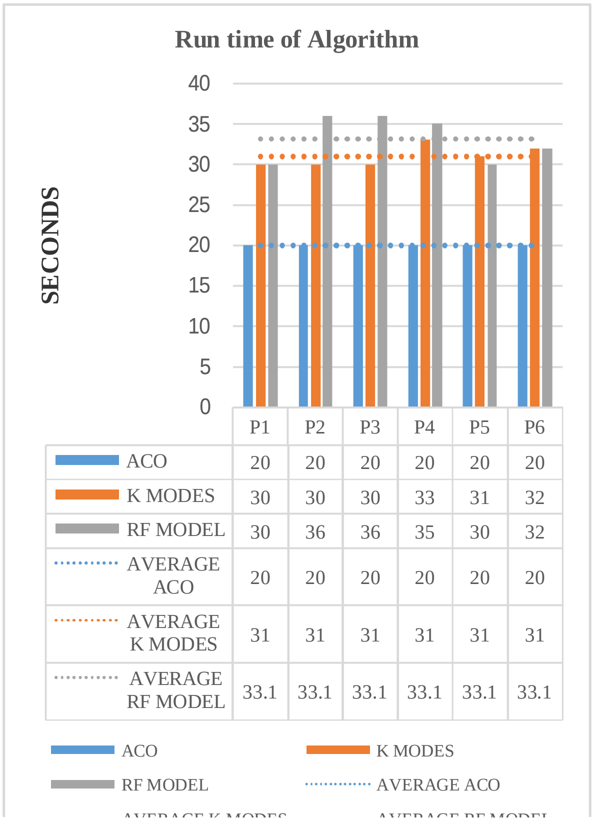


	P 1	P 2	P 3	P 4	P 5	P 6
ACO	0.79	0.666	0.7272	0.666	0.4	0.571
K MODES	0.42	0.035	0.31	0.013	0.19	0.21
RF MODEL	0.54	0.35	0.33	0.35	0.18	0.22
AVERAGE ACO	0.64	0.64	0.64	0.64	0.64	0.64
AVERAGE K MODES	0.2	0.2	0.2	0.2	0.2	0.2
AVERAGE RF MODEL	0.33	0.33	0.33	0.33	0.33	0.33

Fig. 6 chart showing the accuracy value for ACO, K modes and RF Model for six projects

This fig. 5 chart is showing the recall value for six projects. And recall shows us the error identification ratio for all projects. This chart is showing result for ACO, K MODES, and RF Model in which average line showing how on an average the proposed ACO Algorithm is better than average of the other algorithm which is K MODES algorithm and RF (Random Forest) algorithm by which the error in the project are identified. The error identified by the proposed algorithm gives much improved results as in comparison with the other such techniques and algorithms.

This fig. 6 chart is showing the Accuracy value for six projects. Accuracy is representing Correctness by which we are measuring cost estimation. This chart is showing result for ACO, K MODES, and RF Model in which average ACO line showing how on an average the proposed ACO Algorithm is better than other algorithm on average which is K MODES algorithm and RF (Random Forest) algorithm which is used for the purpose of accuracy by which the process of measurement of the accuracy is applied.



	P1	P2	P3	P4	P5	P6
ACO	20	20	20	20	20	20
K MODES	30	30	30	33	31	32
RF MODEL	30	36	36	35	30	32
AVERAGE ACO	20	20	20	20	20	20
AVERAGE K MODES	31	31	31	31	31	31
AVERAGE RF MODEL	33.1	33.1	33.1	33.1	33.1	33.1

Fig. 7 chart showing the execution time for ACO, RF Model and K modes for six projects

This fig.7 chart is showing the duration for six projects. And duration shows us the execution/runtime of algorithms for all projects. This chart is showing result for ACO, K MODES, and RF Model in which average line showing how on an average the proposed ACO Algorithm is better than average of the other algorithm which is K MODES algorithm and RF (Random Forest) algorithm by which the average runtime in the project are identified. The average runtime by the proposed algorithm gives much improved results as in comparison with the other such techniques and algorithms.

VI. CONCLUSION AND FUTURE SCOPE

A) Conclusion

The accurate estimation of software development costs and efforts is a serious subject to take good management decisions. The cost assessment problem Varies largelybetweenorganisations that do their estimation under very different constraints. Software error and effort calculation are both important tasks in instruction to reduce costs in a software company. The predictive model used in these circumstances needs to be both accurate and understandable in this work. In this paper we have introduced a technique ant colony optimization to calculate the cost of project which help for the small companies to predict the cost for similar level requirement project. But quiet there is a limitation that is we are calculating the cost based on finding the malicious error in application but we are occupied at few bugs there are many type of bugs which effect overall performance of application that bugs will consider in future.

B) Future Scope

We are predicting the cost based on finding the malicious error in application but we are occupied at few bugs there are numerous kind of bugs which effect overall performance of application that bugs will consider in future.

REFERENCES

[1] Kim G.H, An, S.H, Kang, K.I. *"Comparison Of Construction Cost Estimating Models Based On Regression Analysis, Neural Networks, And Case-Based Reasoning."* Build Environ. Vol.39, issue 10: pp.1235–42. Oct 2004.

[2] Sharma A, Kushwaha DS. *"Estimation of Software Development Effort from Requirements Based Complexity"*. Procedia Technology. Vol. 4, pp.716–22, 2012.

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

[4] Park H, Baek S. *"An Empirical Validation of a Neural Network Model for Software Effort Estimation"*. Expert Syst Appl. Vol.35, issue 3: pp.929–37, 2008.

[5] A.C. Hodgkinson, and P.W. Garratt, *"A neuro fuzzy cost estimator, Proceedings of Third International Conference on Software Engineering and Applications,"* pp. 401-406, 1999

[6] Mockus A., Weiss D.M. and Zhang P. *"Understanding and Predicting Efforts in Software Projects"*, IEEE Proceedings of 25th International Conference on Software Engineering (ICSE'03), pp. 274-84.

[7] A. J. Albrecht, *"Measuring application development productivity,"* in Proc. Joint SHARE/GUIDE/IBM Application Development Symp., pp.83-92, Oct.1979

[8] -, AD/M Estimating and Productivily Measurement Guidelines, IBM Corp. Information Systems, 1984.

[9] Function Point Counting Practices Manual, Release 4.2, IFPUG

[10] Khatibi V, Jawawi. *"DNA Software Cost Estimation Methods: A Review."* J Emerg Trends Comput Inform Sci, Vol. 2, issue 1, pp.21–29, 2010-11

[11] Kumari S, Pushkar S. *"Performance Analysis of the Software Cost Estimation Methods: A Review"*. Int J Adv Res Comput Sci Software Eng. Vol. 3, issue 7, pp.:229–38, 2013.

[12] Jones C. Estimating Software Costs. Tata Mc-Graw, Hill Edition; 2007.

[13] Low, Graham C., and D. Ross Jeffery, "*Function points in the estimation and evaluation of the software process,*" IEEE Transactions on Software Engineering, Vol. 16, pp. 64-71, Jan 1990.

[14] Mukhopadhyay, Tridas, and Sunder Kekre, "*Software effort models for early estimation of process control applications,*" IEEE Transactions on Software Engineering, vol.18, pp. 915-924, October 1992.

[15] Matson, Jack E., Bruce E. Barrett, and Joseph M. Mellichamp. "*Software development cost estimation using function points.*" IEEE Transactions on Software Engineering, Vol. 20. Issue 4, pp.: 275-287, 1994.

[16] Vijayakumar, S. "*Use of historical data in software cost estimation.*" Computing & Control Engineering Journal, Vol. 8, Issue 3, pp. 113-119, 1997.

[17] JØRGENSEN, M., and K. MOLØKKEN-ØSTVOLD. "*A review of surveys on software effort estimation.*" International Symposium on Empirical Software

Engineering (ISESE'03), Rome. Proceedings. IEEE Computer Society. 2003.

[18] Xu, Zhiwei, and Taghi M. Khoshgoftaar. "*Identification of fuzzy models of software cost estimation.*" Fuzzy Sets and Systems Vol. 145, issue 1, pp. 141-163, 2004.

[19] Zheng, Y., Wang, B., Zheng, Y., & Shi, L. "*Estimation of software projects effort based on function point,*" Proceedings of 2009 4th International Conference on Computer Science & Education, IEEE. pp. 941-943, July 2009

[20] Jeng, B., Yeh, D., Wang, D., Chu, S. L., & Chen, C. M. "*A Specific Effort Estimation Method Using Function Point*." Journal of Information Science and Engineering, Vol. 27, pp. 1363-1376, July 2011.

[21] Attarzadeh, Iman, Amin Mehranzadeh, and Ali Barati, "*Proposing an enhanced artificial neural network prediction model to improve the accuracy in software effort estimation,*" Fourth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, pp. 167-172, 2012.

[22] Malathi, S., and S. Sridhar, "*Effort Estimation in Software Cost Using Team Characteristics Based on Fuzzy Analogy Method–A Diverse Approach,*" Signal Processing and Information Technology, Springer International Publishing, pp.1-8, 2014.

[23] Laqrichi, Safae, François Marmier, and Didier Gourc, "*Software Cost and Duration Estimation Based on Distributed Project Data: A General Framework,*" Springer International Publishing, Vol. 7 pp. 213-224, February 2014.

[24] Julie Moeyersoms, Enric Junqu´e de Fortuny, Karel Dejaeger, Bart Baesens, David Martens, "*Comprehensible Software Fault and Effort Prediction: a Data Mining Approach.*"the Journal of Systems & Software, Vol. 100, pp. 80-90, 2014.

[25] M. Dorigo, V. Maniezzo, A. Colorni, *"Ant system: optimization by a colony of cooperating agents"*, IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol.26, No.1, pp.29-41, 1996.

[26] Selvi, V., and Dr R. Umarani. "*Comparative analysis of ant colony and particle swarm optimization techniques.*" International Journal of Computer Applications,Vol. 5, issue. 4, pp. 0975–8887, 2010.

[27] Huang, Joshua Zhexue. "*Clustering Categorical Data with k-Modes.",* pp. 246-250, 2009.

[28] Breiman, Leo. "Random forests." *Machine learning*, Vol. 45, issue 1, pp. 5-32, 2001.

[29] Powers, David M W. *"Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation"*. Journal of Machine Learning Technologies, Vol. 2, issue 1, pp. 37–63, 2011