iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

An Early Stage Software Effort Estimation in Agile Methodology Based On User Stories Using Machine Learning Techniques

Dr. S. Rama Sree[1], Ch. Prasada Rao[2], M. Mounika[3]
[1]*Professor and Vice-Principal, Department of Computer Science and Engineering, Aditya Engineering College, Surampalem, Andhra Pradesh, India.*
[3]*M.Tech Scholar, Department of Computer Science and Engineering, Aditya Engineering College, Surampalem, Andhra Pradesh, India.*

Abstract: ***Many of the software projects fail due to the cause of over-budgeting or under budgeting. To mitigate the over-budget or under budget, the accurate and effective prediction of software effort in all the software development process is required. Effort estimation at the initial stage of any development of software is a difficult and challenging task due to continuous change in customer requirements. To improve the accuracy and efficient of the software development, most of the software companies are adopting Agile methodologies. By the Agile methodologies the part of software products can be delivered in very short time. So that the estimation of the software development effort like schedule, budget and others can be done in the early stage of software development life cycle. The early and accurate estimations can be possible from Agile Methodologies. Requirements representation in agile methods is often done depending on the sprint size of user stories rendering a WHO, WHAT, WHY and WHEN dimension. The objective of this paper is to estimate size and effort in the agile development from story points, which are calculated from user stories, using machine learning techniques to improve the efficiency and accuracy of the business case from various methods. To evaluate the performance of the estimation, the different metrics or indicators like MMRE, MSE are used and these can be simulated in MAT Lab.***
Keywords: ***effort estimation, user stories, story points, artificial neural networks, support vector regression, random forests.***

I. INTRODUCTION

During the development of software, even the software effort estimation also gained importance in the later years i.e. from 1960's, when it is used for many purposes such as cost estimation, budgeting, monitoring, etc. In doing so, scholars or researchers tend to analyze the problems related to effort estimation goals, suggest new models and use new techniques to achieve greater accuracy as expected [1]. For this to be done, some of the researches followed a criteria which requires goals of estimation, execution steps, applying measurement methods needed and updating the result. In view of effort to be estimated, this process involves managerial problem as well as computational problem.

The effort estimation approach is proceeded by collecting the observed experimental results and practices, solutions to problems and conflicts, from prior studies. Accurate estimation is difficult for software development project since it involves a number of interrelated factors that affect the development of effort and productivity.

In this paper, as the title is put forward, the effort estimation is done by applying various effort estimation methodologies along with various performance measures with the help of MAT Lab tool.

A. The effort estimation methodology

The effort estimation methodology is a combination of both formal and expert estimation methods [1]. The method in which the effort can be estimated may be classified as: the heuristic method and the algorithmic method. Heuristic methods are those methods which primarily focus on expert opinions without implementing any algorithm specifically, while the algorithmic methods require some mathematical formulae from the past experiences along with one or more algorithms.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

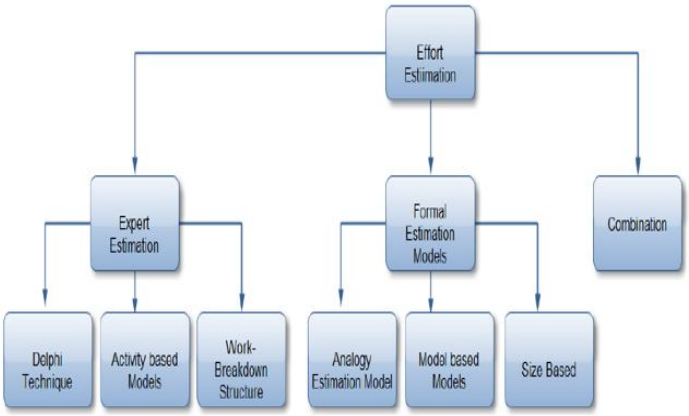


Figure 1: Classification of effort estimation models

Many of the effort estimation models depend on the factor called "size" of the project to be developed, scaling the project on small, medium and large basis. The main motto of these models is that the work load of the project can be divided into smaller parts and combining them again after reforming, in order to produce the best estimate [4].

B. Agile methodology and user stories

From the past to the recent years of advancement in the effort estimation methodology, is said to have been increased due to the intimation of the term "agile", which is an incremental and iterative approach [3], following a set of values and principles progressing over the collective effort of self-organizing cross-functional teams [2]. Agile methods breaks down the product into smaller incremental constructs. Agile methods are bound to primarily focus on process adaptability and customer satisfaction by rapid delivery of working software product. Methodologies of agile include Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), etc [3]**.**

A user story, which is short, provide a medium to communicate more effectively between a user and a client delivering the specified requirements in an iteration [4]. In general, user stories are written on story cards giving a brief description of the need by the customer on a regular conversation taking place iteratively, thereby satisfying the requirements. For the effort to be estimated, user stories are first prioritized and then the velocity is measured for each user story depending on story points. Generally, a user story is represented on a WHO, WHAT and WHY dimension as follows:

As a <role>, I want <goal/desire> so that <benefit>.

E.g., As a registered user, I want to login so that I can access system.

Well-formed user stories can be described by the usage of the acronym INVEST coined by Bill Wake:

Independent	We want to be able to develop in any sequence.
Valuable	Users or customers get some value from the story.
Estimatable	The team must be able to use them for planning.
Small	Large stories are harder to estimate and plan. By the time of iteration planning, the story should be able to be designed, coded, and tested within the iteration.
Testable	Document acceptance criteria, or the definition of done for the story, which lead to test cases.

A story point is a unit of degree in which the size of a User Story or a feature is measured. A story point is measured before the sprint starts to indicate and to compare the feature how big or small [3]. In the process of estimating effort in agile methodology, a point is assigned to each user story followed by the ranking of effort i.e. complex, simple, large, medium, small [4] in a numerical fashion of 5, 4, 3, 2, 1, respectively.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

II. RELATED WORK

In this section of related work, a brief discussion about the project is discussed further in which the advancement of the project can be enhanced and improved by the usage of some of the existing journals relevant to this project. Usually, effort is calculated using Lines Of Code (LOC), Kilo Lines Of Code (KLC) Or Source Lines Of Code (SLOC) [5]. There are number of ways to determine effort in software development projects and by using different metrics, which can be used to measure the estimate produced more accurately and precisely than the previous estimate produced, thereby minimizing errors in the developing software project.

Firstly, as initiated in introduction, in the developing projects related to software engineering, effort estimation plays an imperative role. In the process of estimating effort, various metrics and models are also used. The most commonly used effort estimation model can be initiated as the COCOMO (COst COnstructive MOdel), which is developed by Barry Boehm [6]. The COCOMO model uses Kilo Lines of Code (KLOC) as an input to estimate effort. In general, as developed by Barry Boehm, the COCOMO model has a three layer higher description level (Basic COCOMO, Intermediate COCOMO, and Detailed COCOMO), along with other three models (Organic, Semi-detached, Embedded models). COCOMO is represented by the equation:

$$E=\lambda \times Size^{\mu} \times EAF$$

Tther models, which are predominantly in use in the process of effort estimation is that of the COCOMO II and the Function Point Analysis models.

COCOMO II is an overextended version of COCOMO, which can be embedded within the latest software type for the better estimate possible, with the three possible variants used: namely the Application Composition (AC), Early Design (ED), and Post-Architecture (PA) variants [6].

Function Point Analysis (FPA) is an adaptable model. In the process of estimating effort, size of the project plays a key role. So, FPA is used to determine both the size as well as the effort needed in developing the software project in terms of PM (Person-Months) or MM (Man-Months) [6]. The most commonly used are the Albrecht & Gaffney model and Kermer model. In the process of function point analysis model, several steps are followed for the estimate to be the best. The steps include such as calculating Un-adjusted Function Points (UFP), which range in the complexities of low, medium, or high. This sequence is followed by the calculation of Technical Complexity Factor (TCF). Then, by the multiplication of UFP and TCF yields the result of FPA.

The other estimation is based on the study is the Use Case-based Estimation. This model is similar to that of FPA. The only difference is that use cases and actors are defined first in this model. The sequence involves the calculation of UUCP (Unadjusted Use Case Points) with the complexity range simple, medium, or complex. Then, the TCF (13 elements) and ECF (Environment Complexity Factor - 8 elements) are calculated [7]. At the final point, the estimated result is compared to the current result with correction factor (PF). The formal equation of this model is presented below:

Total Estimation=UUCP x TCF x ECF x PF

III.PROPOSED METHODOLOGY

This section deals with the methodologies or the techniques used in the implementation of the rest of the project thesis. These methodologies or the techniques used include such as Artificial Neural Networks, Support Vector Kernel Regression Method and the Random Forest Technique

A. Artificial neural networks

Artificial neural networks are a mathematical model representation of biological neural networks (BNN) [8]. Neural Networks maintains input/ output relationship [9]. Neural networks implementation using mat lab is an ease, as it provides a GUI interface. Implementation can be done using nntool and nntrain tool.

Implementing the Neural Network using mat lab tool:

1) Define the dataset.
2) Choose a train function to train the network (trainlm).
3) Create a network with hidden layer size (default 10) and train function.
4) Divide the data for training (75%), testing (15%) and validation (15%).
5) Train the network specifying a halting condition (Mean Square Error) to stop training, to achieve higher accuracy with defined number of iterations (500 to 1000).

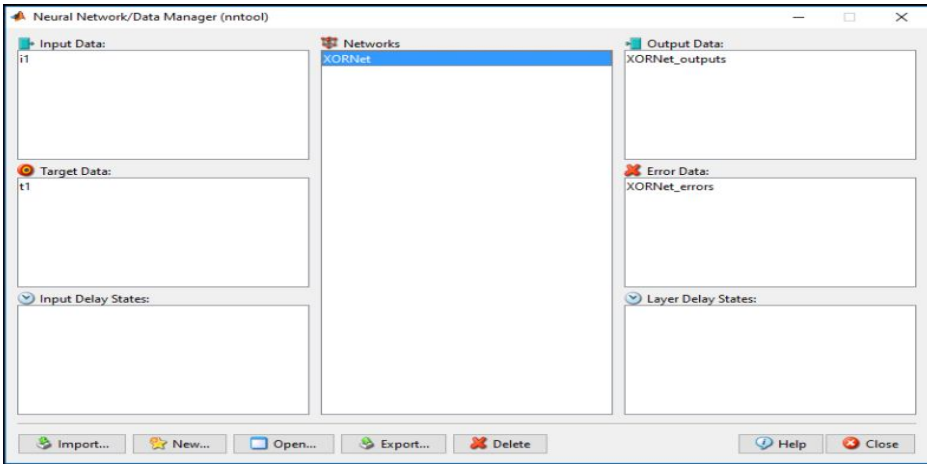


Figure 2: Creating neural network using nntool

B. Support vector kernel regression model

Support Vector Machine was first developed by Vapnik based on Chervonenkis principle. SVR uses the same principles as SVM for classification, but with a few minor differences.

Regression is a machine learning technique used to fit an equation to a dataset.

Algorithm for SVR model:

1. Initialize the dataset.
2. Choose the class label attribute.
3. Transformation of attributes to numeric data types.
4. The data set normalization is done.
5. Perform regression in mat lab
6. Compare the performance.

Out of the different regression functions used in SVR, the Gaussian Kernel Function out performs all the other five functions (Linear Kernel Regression, Sigmoid Kernel Regression, Polynomial Kernel Regression, Radial Basis Function and Gaussian Kernel Regression) by resulting in the output more effectively and efficiently.

C. Random forest technique

The Random Forest technique was first developed Leo Breiman in the year 1999, built on the CART (Classification and Regression Tree Algorithm) decision tree, which is the core building block of a Random Forest implemented using machine learning techniques. [10].

Random Forests are used to predict continuous variables, and also to estimate using probability of “yes/no” happenings or one of several possibilities based on sampling. For this to be implemented, Breiman discovered a machine learning method, “bagger”, yielding in the lack of accurate estimates produced.

So, he proposed a model in which a new random subset of predictors in each node of a tree are selected which are different from other nodes of the tree [11]. As the tree grows larger, the selection of the tree node become more complex. So, he suggested four possible rules to minimize this problem. It is made clear that by the process of “averaging (in regression) or voting (in classification)”, prediction can be determined. These predictions are formed by the combination of trees for the new data.

Another essential idea in Random Forests technique is that of “proximity”. The proximity can be computed by counting the number of trees match divided by number of trees tested. The proximity matrix stores the results of these record pairs of a tree. The proximity matrix is used in the identification of “outliers” and “missing values”. In random forest technique, only training is done and in place of testing, the data which is left is used to evaluate the performance of current tree. Class weights are used in this technique for better accurate performance of the specified class. When a tree is grown every time by input training data, the data often can have missing values. Random forest technique offers two ways for missing value imputation:

1) Dropping the data points with missing values and

2) Fill the missing values with median or mode for numerical or categorical values respectively.

Variable importance in random forest technique is a method to measure relative importance of any predictor. This variable importance can be measured in two ways: traditional approach and the novel approach. The traditional approach averages the variable importance measures of all the individual trees. Whereas the novel approach estimates the importance of the m^{th} variable, which is used in multinomial target models.

D. Working of proposed methodology

In order to measure estimation accuracy of the model, the dataset is collected from 21 previously developed projects from 6 software houses. These projects have been developed using Agile Software development methodology. This experimental analysis was performed by a group of research students, who were unaware of actual results.

1) *Calculation of Total Number of Story Points and Project Velocity*: The data is collected from projects which are developed previously from the past data. Then story points and their corresponding project velocity is calculated from user stories.
2) *Performing Model Selection*: In this step, the model that provides lower Mean Magnitude of Relative Error (MMRE) value is selected as the best model.
3) *Performance Evaluation:* The performance of the model is evaluated using MMRE and MSE values.

IV. PERFORMANCE MEASURES

The Mean Magnitude Relative Error (**MMRE**) is an average error which is the difference between the actual effort and the predicted effort divided by actual effort, given by the formula:

$$MMRE = \sum_{1}^{N} \frac{Actual\ Effort - Predicted\ Effort}{Actual\ Effort}$$

The Mean Square Error (MSE) is the average squared difference between predicted effort and actual effort, given by the formula:

$$MSE = \frac{1}{n}\sum_{1}^{n}(predicted\ effort - actual\ effort)^2$$

V. RESULTS

In the process of estimating effort in agile methodology, by using different machine learning techniques, it is proved that the Random Forest Technique performance using MMRE, is the best technique compared to Artificial Neural Networks and Support Vector Regression model. For this to be done, the data which is collected from 21 previously developed projects from 6 software houses used in [3] is selected. The results are tabulated as shown below.

TABLE 1: Comparison of MMRE and MSE

PERFORMANCE MEASURES / METHODOLOGY	MMRE	MSE
ARTIFICIAL NEURAL NETWORKS	1.0932	12.41
SUPPORT VECTOR REGRESSION	0.2694	1.84
RANDOM FOREST	0	0.26

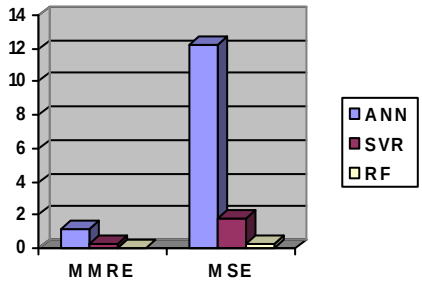


Figure 3: Comparison of results

VI. CONCLUSION

Random Forest is fast to build and to predict. Random Forest technique can be of appropriate use for the analysis of complex datasets. The selection of kernels and their parameters in SVR is a difficult task. Gaussian kernel is made as the default parameter. Training and Testing is quite slow compared to ANN. In SVR, training is easy. Artificial neural networks which are inspired by the learning processes can be seeminglessly used in the implementation of optimisation process. These learning processes takes rough approximation tasks. The optimisation of the learning process is done based on the function approximation error measure. Lower the error rate and higher the prediction value, more accurate is the estimate produced. So, by comparing the results obtained using machine learning techniques, it is proved that Random Forest technique is the best preferred technique compared to ANN and SVR.

VII. THREATS TO VALIDITY

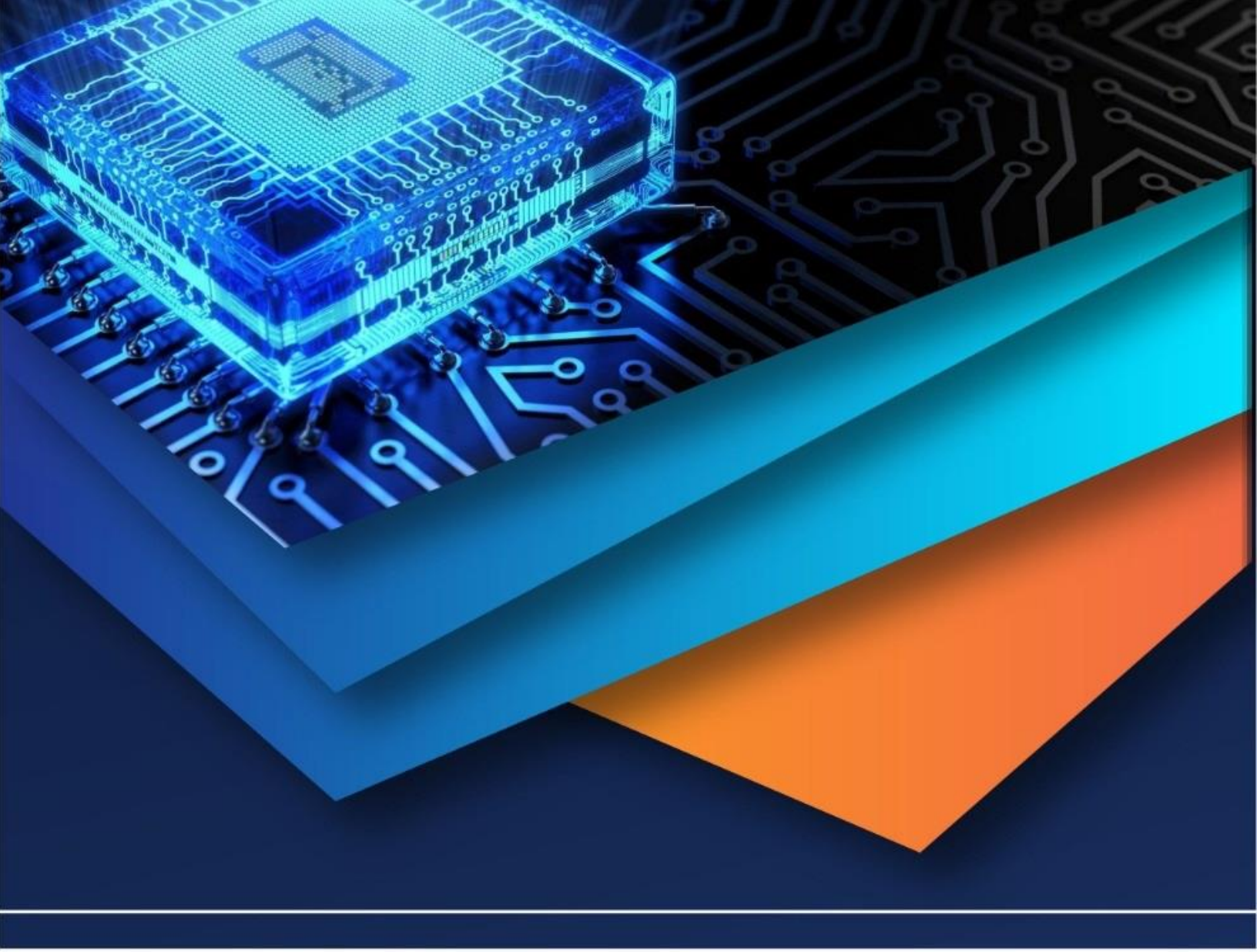
1) The main problem with Agile Story Points are not available in publicly.
2) To train, validate and test the model with higher data set will be given more accurate results but here in this implementation there are only 21 project data set is considered.
3) Machine learning techniques are more popular for accurate predictions but selection and implementation of suitable model is also tedious and time consuming.

VII. ACKNOWLEDGMENT

I would like to express my gratitude to my Guide Dr.S.Rama Sree, Professor and Mr.Ch.Prasad Rao, Associate Professor, without their guidance, this paper is not possible. Their understanding, encouragement and personal guidance have provided the basis for this paper.

REFERENCES

[1] Ali Bou Nassif and Luiz Fernando Capretz Danny Ho**,** "Estimating Software Effort Based on Use Case Point Model Using Sugeno Fuzzy Inference System", 2011, 23rd IEEE International Conference on Tools with Artificial Intelligence.
[2] Kayhan Moharreri, Alhad Vinayak Sapre, Jayashree Ramanathan, Rajiv Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments", 2016 IEEE, 40th Annual Computer Software and Applications Conference.
[3] Ziauddin, Shahid Kamal Tipu, Shahrukh Zia, "An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA), Vol. 2, No. 1, 2012, ISSN 2166-2924.
[4] Evita Coelho ,Anirban Basu, "Effort Estimation in Agile Software Development using Story Points", International Journal of Applied Information Systems (IJAIS), ISSN : 2249-0868, Foundation of Computer Science FCS, New York, USA Volume 3– No.7, August 2012.
[5] Aarti Hans, Sonal Gahlot, "A Review: Software Metrics and Effort Estimation in Aspect Software Development Program", ISSN No. 0976-5697, Volume 7, No. 1, January-February 2016, International Journal of Advanced Research in Computer Science.
[6] Sultan Aljahdali, Alaa F. Sheta, Narayan C. Debnath, "Estimating Software Effort and Function Point Using Regression, Support Vector Machine and Artificial Neural Networks Models", 2015 IEEE.
[7] Pragya Jha, Preetam Pratap Jena, Rajani Kanta Malu, "Estimating Software Development Effort using UML Use Case Point (UCP) Method with a Modified set of Environmental Factors", Pragya Jha et al, / (IJCSIT), International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2742-2744, ISSN-0975-9646.
[8] Sumeet Kaur Sehra, Yadwinder Singh Brar, and Navdeep Kaur, "Soft Computing Techniques For Software Project Effort Estimation", International Journal of Advanced Computer and Mathematical Sciences, ISSN 2230-9624,Vol 2, Issue 3, 2011, pp 160-167.
[9] Lekshmi R, Binu Rajan, "Survey on Different Machine Learning Techniques for Software Effort Estimation", International Journal of Computer Applications (0975 – 8887) Volume 95– No.25, June 2014.
[10] LEO BREIMAN, "Random Forests", Machine Learning, 45, 5–32, 2001, Statistics Department, University of California, Berkeley, CA 94720.
[11] Anjali Sharma, Karambir, "Empirical Validation of Random Forest for Agile Software Effort Estimation Based On Story Points", International Journal of Engineering Sciences & Research Technology, Sharma et al., 5(7): July, 2016, ISSN: 2277-9655, Impact Factor: 4.116.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY