Innovations Syst Softw Eng (2017) 13:191–200
DOI 10.1007/s11334-017-0288-z

ORIGINAL PAPER

Empirical assessment of machine learning models for agile software development effort estimation using story points

Shashank Mouli Satapathy[1] · Santanu Kumar Rath[2]

Received: 15 September 2016 / Accepted: 1 June 2017 / Published online: 13 June 2017

Abstract In the present day developing houses, the procedures adopted during the development of software using agile methodologies are acknowledged as a better option than the procedures followed during conventional software development due to its innate characteristics such as iterative development, rapid delivery and reduced risk. Hence, it is desirable that the software development industries should have proper planning for estimating the effort required in agile software development. The existing techniques such as expert opinion, analogy and disaggregation are mostly observed to be ad hoc and in this manner inclined to be mistaken in a number of cases. One of the various approaches for calculating effort of agile projects in an empirical way is the story point approach (SPA). This paper presents a study on analysis of prediction accuracy of estimation process executed in order to improve it using SPA. Different machine learning techniques such as decision tree, stochastic gradient boosting and random forest are considered in order to assess prediction more qualitatively. A comparative analysis of these techniques with existing techniques is also presented and analyzed in order to critically examine their performance.

Keywords Agile software development · Decision tree · Random forest · Stochastic gradient boosting · Software effort estimation · Story point approach

✉ Shashank Mouli Satapathy
shashankamouli@gmail.com

Santanu Kumar Rath
skrath@nitrkl.ac.in

1 School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India

2 Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Orissa, India

1 Introduction

Development of software using agile methodologies helps organizations in producing a software that responds to the requirements volatility. These methodologies provide opportunities to assess the bearing of inclusion of changing requirements during software development throughout the development life cycle [8]. Agile software development allows the teams to clearly re-plan their releases so as to improve their worth all through the development [4,33]. Thus, companies following agile methods bring competition to others in the marketplace that do not use agile methods [6].

Since predictability of requisite resources is the primary goal at the starting phase of project management, appraisal of the size and unpredictability of the products becomes the focus in agile development [31,41]. For this purpose of prediction of resources, requirements should be gathered. Current practices rely on expert-judgment, analogy-based estimation which can be biased, labor intensive and inaccurate in many cases [37,42]. There are many reasons behind performing effort estimation for agile software. As per Trendowicz and Jeffery [38], it is performed to oversee and lessen the risk associated with development of agile projects. It is also used for process management and for discovering essential rules and productivity measures in order to negotiate on project scope and resources. Hence, it is extremely important to know the consequences of improper estimation before finding out the most suitable approach for effort estimation of software using agile methodologies [34].

It is observed that the gathered requirements during agile software development process are scribbled down in cards, termed as user stories [22,36]. Story points are utilized in order to evaluate these stories. The job of relating story points to the corresponding effort value is accomplished by the development team. The velocity of development is computed by finding out the total number of story points completed in a typical sprint by a team. While following SPA, the total number of story points are used along with project velocity to compute the required development effort. The average error in effort estimation is measured between 20 and 30% [1,11,15]. Therefore, an attempt has been made to lower the number of errors in the accuracy rate by applying three different types of machine learning (ML) techniques such as DT, SGB and RF on the story point dataset. The results and performance obtained by applying these machine learning techniques are compared among themselves as well as with the results obtained by other authors available in the literature and their performance is assessed.

The content of the work presented in this article is organized as follows. First, the theoretical background about effort estimation and effort estimation in an agile software development project is presented. Then, a systematic literature review followed by the various methodologies considered for implementation has been described. The next section describes different evaluation criteria considered for evaluation. The work proposed along with the results and a discussion and comparative analysis of various results is presented in the next three sections. The last two sections present the conclusions drawn from the proposed work with accentuation on the work done. The limitations associated are also highlighted. The extension for further research work in this direction has been explained at the end.

2 Related work

Keaveney and Conboy [16] have investigated the applicability of conventional estimation techniques toward software development approaches using agile methodologies considering various case studies based on agile methods utilized within diverse organizations. Coelho and Basu [5] have depicted the strides followed in story point-based technique for agile software effort estimation and highlighted various areas which should be investigated for further research. Schmietendorf et al. [33] have given an examination about estimation of conceivable outcomes, particularly considering the paradigm using the case of extreme programming. Lenarduzzi et al. [18] have presented function point metrics to enhance estimation precision and to quantify the exactness of expert estimation. They further extended this process to deal with Scrum requirements, where the first study was duplicated twice and acquire extremely exact and higher prediction accuracy than the values obtained through function point metrics.

Zia et al. [43] have developed an effort estimation model for software designed using agile methodologies especially Scrum. The designed model was adjusted with the assistance of the empirical dataset procured from twenty-one projects developed by six software companies. Usman et al. [40] have provided a point by point review of the best in class in agile software development effort estimation area. They considered 25 primary studies for review and identified several research gaps relating to the methods utilized for agile software development, metrics utilized in order to evaluate the size of the software and cost drivers. Hearty et al. [13] have proposed a Bayesian network model considering XP approach and indicated how it could gain from project data keeping in mind the end goal to predict the effort and risk appraisals without obliging any extra metrics.

Popli and Chauhan [28] have considered regression analysis in order to develop an agile software effort estimation model. Hussain et al. [14] have proposed a methodology which helps in expelling issues like formalized client prerequisites and hence apply function points for estimating the effort required to develop software using agile methodologies. Hamouda [12] presented a procedure and strategy that sureties relativity in sizing the software utilizing story points. This proposed procedure and system was connected in a Capability Maturity Model Integration (CMMI) level three company on different projects. Ungan et al. [39] have looked at SPA used by Scrum and Planning Poker, with COSMIC Function Points (CFP)-based effort estimation model for selection of projects by using regression models and ANN methodology.

Viljan Mahnic [19] studied the behavior of development teams with the help of a case study in order to bring Scrum into their development process. He found that although the initial estimated results were over-optimistic, the planning and estimation abilities improved in the subsequent sprints. Mahnic and Zabkar [20] extended the previous approach by describing a set of measures that helped in understanding the Scrum-based software development process continuously by IT administration. The proposed measures were considered and applied in order to rebuild a Web site, which ultimately served as a contextual investigation for assessment. Garg and Gupta [10] have applied principal component analysis (PCA) for dimension reduction and applied constraint solving approach for satisfying the agile manifesto criteria. From the analysis of results, it is observed that the proposed model outperforms existing models in terms of lower MMRE.

Lenarduzzi et al. [18] considered functional size metrics (FSM) in order to evaluate the accuracy obtained from expert-based estimation technique. They also proposed an extended approach by replicating the original Scrum process-based study two times. Results obtained by analyzing the replicated study exhibit an improved prediction accuracy than FSM. Raslan et al. [29] have proposed a fuzzy logic technique-based effort estimation framework considering user stories. They implemented fuzzy logic technique using trapezoidal membership functions for characterizing inputs parameters.

Britto et al. [4] investigated the effort estimation practice followed in Agile Global Software Development (AGSD) context empirically and observed that estimation results remained unchanged.

Agile software effort estimation is also one of the promising areas of research. Many researchers have proposed various methodologies for agile software development process. But there is lack of availability of good amount of research work for providing a systematic procedure in order to estimate the effort of softwares developed using agile methodology. Story point approach is one of the popular ways to estimate the effort of softwares developed using Scrum methodology. But there is a very scarcity of dataset based on story point approach due to unavailability of project velocity information. Hence, by applying the different statistical and machine learning techniques on the considered story point dataset, the accuracy of agile software effort estimation process will be improved.

3 Methodologies used

The methodologies mentioned below are employed in this paper in order to determine the effort of an agile software product.

3.1 Story point approach

Story point is a measure to quantify the size of a user story or feature. A story point might be assigned in view of the effort including the complexity and the inalienable risk in building up a story [5]. An appraisal of the effort of building up a user story requires the designer to have some experience of evaluating, to have admittance to historical data and to have the opportunity to utilize a trial-based estimation approach. The block diagram, appeared in Fig. 1, demonstrates the steps to ascertain the project development effort utilizing story point approach.

The story points and the final velocity value of the agile projects are then taken as input arguments for various ML models in order to calculate normalized effort.

3.2 Decision tree

A decision tree (DT) is an intelligent model characterized by a binary tree that illustrates the prediction of a dependent variable using a set of predictor variables. In 1963, Morgan and Sonquist developed the first DT program called as *Automatic Interaction Detection (AID)* [24], enhanced the aspect by the THAID program in 1973 [23]. DT model helps the non-technical persons to visualize a clear and large picture of a certain problem, which in turn improve their understanding about the problem. Still, the DT model has certain disadvantages. In DT, each node is optimized locally rather than globally for the whole tree. In addition to that, DT models also might suffer from the over-fitting problem, which enhances the inaccuracy in the result.

3.3 Stochastic gradient boosting technique

The stochastic gradient boosting (SGB) is also an intelligent model often termed as Treeboost model [9]. “Boosting” implies applying the function iteratively in a series and consolidate the yield of every function with a weighting coefficient keeping the goal in mind the end goal to minimize the aggregate prediction error and improve the accuracy. The stochastic gradient boosting (Treeboost) algorithm has already been used few researcher for Software Development Effort Estimation by considering use case point approach and class point approach, respectively [25,30]. The model of the SGB can be represented as:

$$G(n) = G0 + F1 \times S1(n) + F2 \times S2(n) + \cdots + FM \times SM(n) \quad (1)$$

where $G(n)$ is the predicted value. $G0$ is the initial value for the series. Vector n represents the pseudo-residuals prevailing at a specific point in the series. For fitting pseudo-residuals, a progression of trees $S1(y)$, $S2(y)$, etc., is enforced. These are coefficients of the tree node anticipated values, obtained by employing the SGB technique.

In case of regression, the *Huber-M loss function* is utilized for SGB technique. The purpose of this function is to square the residuals having less value than that of the *Huber's quantile cutoff* value and to consider the absolute values of

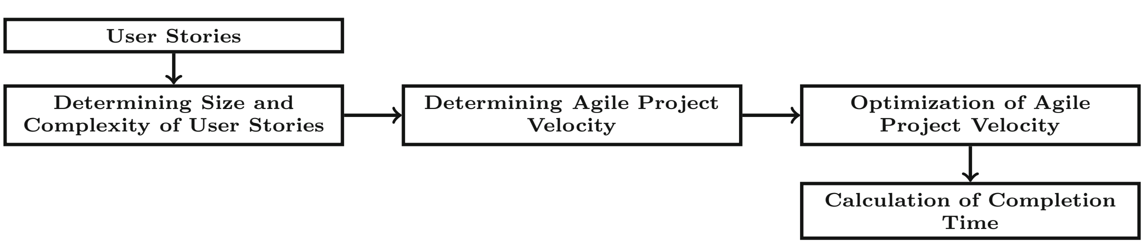

Fig. 1 Steps to calculate effort using story point approach

residuals for other cases. *"Stochastic"* implies that an arbitrary percentage (50% recommended) of training data are utilized for every iteration rather than all. A *shrinkage factor*, having value around 0 and 1, is augmented to every tree in the series in order to defer the learning process and prolong the length of the series. This process in turn improves the prediction accuracy. An *Influence Trimming Factor* is also employed in order to amend the procedure that permits to eliminate the rows having small residuals.

3.4 Random forest technique

Random forest (RF) is an ensemble learning approach that constructs various DTs amid training period and picks the final class through selecting its mode [3]. RF has been applied for software development effort estimation by few researcher by considering use case point approach [26,32]. The concept behind ensemble learning is that RF produces many classification trees. In case of regression, the average of prediction results obtained from individual tree denotes the prediction accuracy of the forest. While using RF technique for regression purpose, the developing trees are considered for creation of forest with the help of a random vector λ. The random vector λ specifies that the tree predictor $f(s, \lambda)$ considers numerical data instead of class labels. The predictor output is denoted by $f(s)$ and the original value of effort is denoted by A. The generalized mean-squared error for any numerical predictor $f(s)$ is calculated as

$$E_{s,A} = (A - f(s))^2 \tag{2}$$

In order to model the RF predictor, the average value obtained over k trees $f(s, \lambda_k)$ is considered.

4 Evaluation criteria

The evaluation of the performance obtained using various models is accessed by considering below-mentioned criteria [2,21,27].

- The Mean Absolute Error (MAE) is the average of the absolute errors between the actual and the predicted effort as shown in Equation 3.

$$MAE = \frac{1}{TP}\sum_{i=1}^{TP} |AE_i - PE_i| \tag{3}$$

where $AE_i =$ Original effort value collected from the dataset for the ith test data. $PE_i =$ Output (predicted effort) obtained using the developed model for the ith test data. $TP =$ Total no. of projects in the test set.

- The Mean of Magnitude of Error Relative to the estimate (MMER) is one of the criteria used for effort estimation models evaluation. It is contended that MMER can provide higher accuracy than the Mean Magnitude of Relative Error (MMRE) [7,17]. MMER is the mean of MER as shown in Equation 4.

$$MMER = \frac{1}{TP}\sum_{i=1}^{TP} \frac{|AE_i - PE_i|}{PE_i} \tag{4}$$

- The Prediction Accuracy (PRED (x)) is PRED which can be described as the average of the MAE's off by no more than x as shown in Equation 5.

$$PRED(x) = \frac{1}{TP}\sum_{i=1}^{TP} \begin{cases} 1 & \text{if } \mathrm{MAE}_i \le x \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

The accuracy of the estimates is directly corresponding to *PRED*(x) and conversely relative to MMER.

5 Dataset description

The proposed methodology is actualized utilizing the dataset of twenty-one software projects developed by six number of software houses [43]. In the dataset, three parameters of each project have been identified and collected for implementation purpose. The main parameter demonstrates the total story points required, the next parameter signifies the project velocity considering the deceleration factor value, and the third parameter speaks about the actual effort utilized to finish the development of the project. This dataset is utilized to calculate the Scrum-based agile software development effort.

6 Proposed approach

Assessment of effort required to develop applications based on agile methodologies has been implemented using different machine learning techniques by considering the twenty-one projects in the dataset as input to the model. The statistical profile of the dataset based on SPA is depicted in Table 1.

Figure 2 depicts the relationship between software size (total number of story point) and actual effort (person-hours) based on SPA using 21 project dataset.

From these figures, it is observed that the 21 project dataset based on SPA contains a few number of outliers. From Table 1, it has been observed that the dataset is not normally distributed based on the values of parameters, i.e., skewness and kurtosis. Hence, in order to make the data normally distributed, logarithmic transformation is applied over the dataset. Figure 2 also displays the histogram of effort

Table 1 Statistical profile of datasets based on SPA

Project type	Minimum	Maximum	Mean	Median	SD	Skewness	Kurtosis
21 Project dataset	21	112	56.43	52	26.18	0.65	−0.77

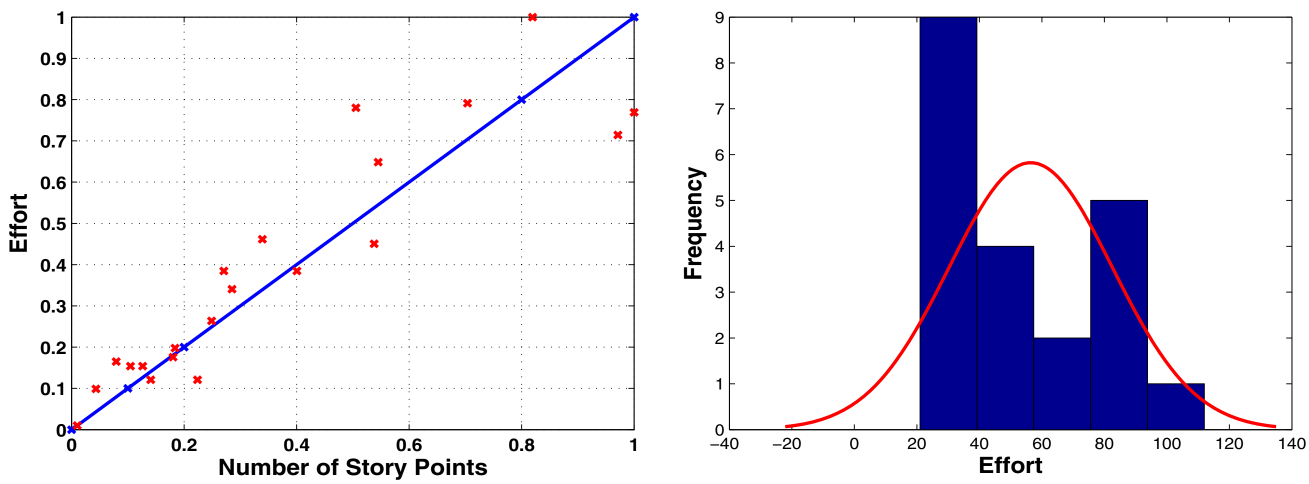

Fig. 2 Software size versus effort graph and histogram of effort values based on story point approach

value based on SPA for agile software effort estimation. The values for number of story point and effort have been collected from Zia et al. [43] and are normalized with in the range 0–1 before presenting in the figure in order to obtain an uniform and clear result to be represented in the figure. From the figure, it can be observed that the data are more normally distributed. Figure 3 demonstrates the steps carried out in the proposed research work in order to compute the effort required to develop softwares considering Scrum methodology and optimizing the results by applying different machine learning techniques on the considered dataset.

Steps followed for agile software effort estimation:

1. *Collection of story point, project velocity and actual effort values*: The parameters such as number of story points required to complete the project, velocity of the project after considering deceleration factor values and the actual effort values are gathered from the article by Zia et al. [43].
2. *Data normally distributed?*: The statistical analysis of the collected dataset has been performed and verified in order to check whether the collected dataset follows normal distribution or not based on the values of skewness and kurtosis. If data are normally distributed, then it will directly proceed to the data normalization step. Otherwise, the data need to be transformed to make it more normally distributed.
3. *Transformation of data*: If the dataset is not normally distributed, then the logarithmic transformation process may be applied over the dataset in order to make it normally distributed. Histograms have been plotted to properly verify the distribution of data before and after transformation.

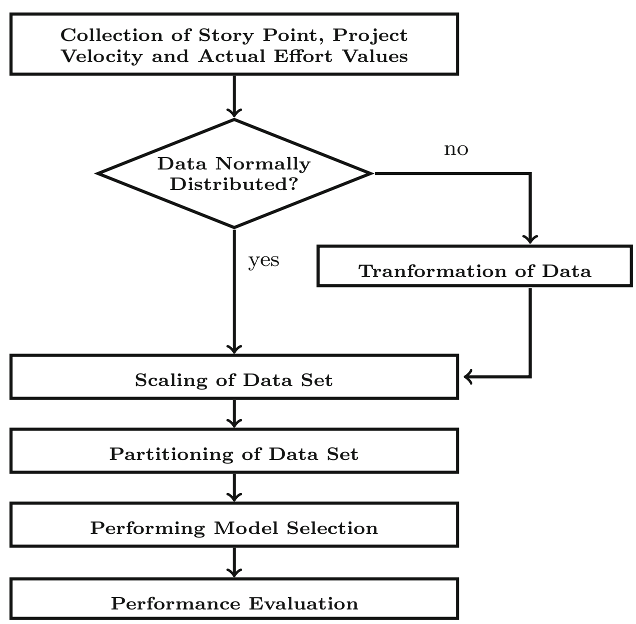

Fig. 3 Proposed steps for agile software effort estimation purpose applying machine learning techniques

4. *Scaling of dataset*: The input parameter values, i.e., the story point count and velocity of the project are individually scaled between the range 0 to 1. Let T denote the total dataset and t denote a single project record provided in T. Then the scaled value of t can be evaluated as:

$$\text{Scaled}\ (t) = \frac{t - \min(T)}{\max(T) - \min(T)} \tag{6}$$

where $\min(T)$ = smallest esteem in T, $\max(T)$ = largest esteem in T, if $\min(T)$ equals $\max(T)$, then Scaled(t) computed as 0.5.

5. *Partitioning of dataset*: The complete dataset is segregated into two sub datasets, i.e., training dataset and test dataset for both SGB and RF techniques. In case of SGB technique, tenfold cross-validation and for RF technique leave-one out validation is considered.
6. *Performing model selection*: The effort value is predicted using various ML techniques used in this study, i.e., SGB and RF. The detailed description of the steps carried out for predicting the effort using these machine learning techniques is provided in the following Experimental Details section.
7. *Performance evaluation*: In this study, the Mean Magnitude of Error Relative to the estimate (MMER), MdMER and the Prediction Accuracy (PRED (x)) are considered as the measures in order to compute the model's performance. The proposed SGB- and RF-based effort estimation model's results are compared and assessed with the results obtained from exiting approaches as available in the literature.

The ML techniques are implemented using the above steps. At last, a correlation of results acquired utilizing the considered two techniques-based effort estimation model with other existing models is displayed in order to evaluate their performances.

7 Experimental evaluation

For actualizing the considered approaches, dataset available in article [43] has been utilized. The inputs to the various ML models are story point count and velocity of project development. Output of the model is predicted effort value, i.e., amount of time necessary to finish the project. The RF model considered leave-one-out validation method during implementation and other models are validated using tenfold cross-validation in order to obtain better accuracy. The implementation process for the considered techniques is executed utilizing MATLAB.

7.1 Model design using decision tree technique

The different parameters considered for development of DT technique-based effort estimation model are carefully identified, and their values are finalized in such a way that the model should exhibit minimal error. But one exception is there, which is tree pruning. The minimum value obtained by employing cross-validation is considered in order to perform tree pruning. The tree was pruned based on the minimum value of the cross-validation error. Pruning helps to simplify

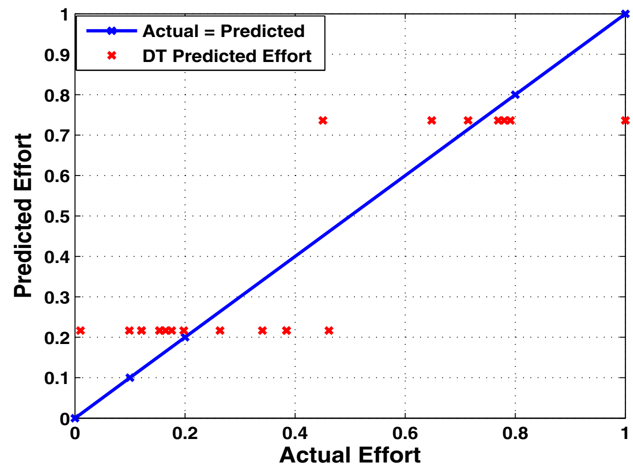

Fig. 4 Deviation of predicted effort from actual obtained using DT technique

the model, but it has a negative impact on the value of accuracy. The various parameters considered in order to obtain the DT model and their corresponding values are presented below:

– Min. Number of Rows in a Node: 5
– Min. Node Size for Splitting: 10
– N-Fold Cross-validation: tenfold
– Max. Number of Levels in a Tree: 10
– Smooth Minimum Spikes: 3

These parameters value are decided by picking proper combinations in order to produce results with maximum accuracy considering DT-based model for estimating web development effort.

Figure 4 depicts the variation of predicted effort values from actual by applying DT-based effort estimation model for softwares developed using agile methodologies. From the figure, it is observed that the deviation between the results of predicted effort and the actual effort is on a bit higher side. Hence, the accuracy is observed to be lower than other models.

7.2 Model design using stochastic gradient boosting technique

To outline a SGB technique-based effort estimation model, the accompanying parameters are considered, which help to find the predicted effort. It is assumed that:

– Individual Tree Depth: 5
– Smooth Minimum Spikes: 5
– Huber's Quantile Cutoff: 0.95
– Influence Trimming Factor: 0.01
– Shrinkage Factor: 0.05

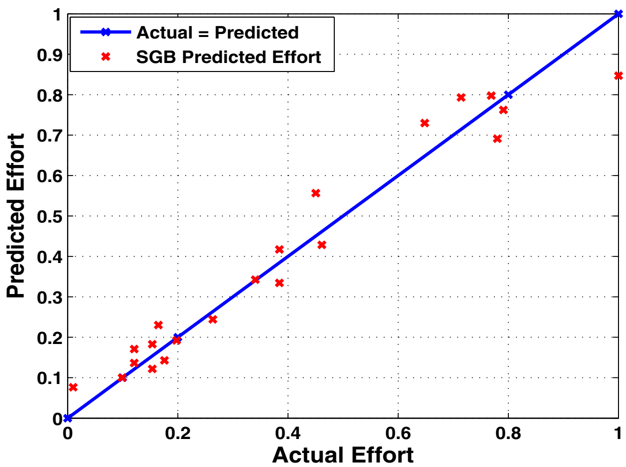

Fig. 5 Deviation of predicted effort from actual obtained using SGB technique

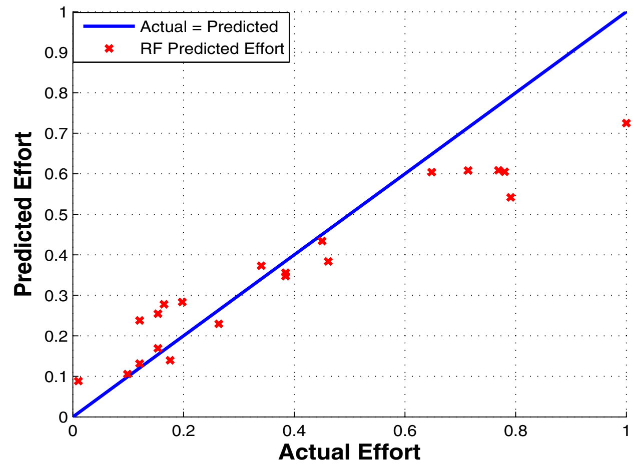

Fig. 6 Deviation of predicted effort from actual obtained using RF technique

- Stochastic Factor: 0.5
- N-Fold Cross-validation: tenfold

The comprehensive depiction of these parameters was described by Jerome H Friedman [9]. The values of these parameters are decided by picking proper combinations in order to produce results with maximum accuracy considering SGB-based model for estimating web development effort.

Figure 5 depicts the variation of values indicating predicted effort from actual by applying SGB-based effort estimation model for softwares developed using agile methodologies. From the figure, it is noticed that there is barely less deviation between the computed effort and the original values. Hence, the accuracy is observed to be quite high.

7.3 Model design using random forest technique

The Brieman's algorithm has been applied by a good number of authors to implement the random forest technique [3]. The steps considered in the algorithm help in building each tree. In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally during the runtime. RF has arbitrariness in input value and node splitting. Subsequently, if there should be an occurrence of RF procedure, at first a self-assertive irregular vector is chosen to give arbitrariness in input value and to begin the usage procedure. At that point, the data are separated utilizing this discretionary arbitrary vector. Results of RF system are obtained after estimation change has been indicated by random vector. So an assessment function, i.e. $(1 - \text{MMER} + \text{PRED}\ (x))$, is utilized to locate the random vector, which gives ideal quality to the assessment function. It is considered as the actual random vector to be used for implementation. Then, by using this final random vector, results are being predicted and assessed.

Table 2 Comparison of proposed results with existing work

	SGB	RF	Regression [43]
PRED (25)	47.6190	66.6667	57.14

This deviation is unmistakably noticeable from Fig. 6. The figure likewise demonstrates the deviation of the computed result from the actual one acquired utilizing RF system. A number of different data objects are produced by RF model, which need to be premeditated when application of RF technique is considered. The outcomes got from these data objects may be assessed for evaluating the performance accomplished.

8 Comparative analysis

While using the MMER, MdMER and PRED (x) in evaluation, significantly convincing results are implied by lower esteem of the MMER, MdMER and higher esteem of the PRED (x). Algorithms such as stochastic gradient boosting (SGB) algorithm and Decision Tree Forests algorithm exhibit functional similarity. The DT which uses recursive partitioning method to split each node is split into two nodes with the help of a splitting variable. SGB makes an ensemble of trees, and it furthermore utilizes randomization amid the manifestations of the trees. It creates a series of trees, and the accuracy of computation is figured by nourishing the outcome got starting with one tree then onto the next tree in the arrangement. In any case, RF manufactures trees in parallel and furthermore utilizes voting strategy during the process of prediction.

In Table 2, results obtained have been compared with the results obtained using existing algorithms available in the

Table 3 Comparison of computation results obtained by employing the DT, SGB and RF techniques

	MMER	MdMER	PRED (25)	PRED (50)	PRED (75)	PRED (100)
Decision tree	0.3820	0.2896	38.0952	71.4286	80.9524	90.4762
Stochastic gradient boosting	0.1632	0.1151	85.7143	95.2381	95.2381	95.2381
Random forest	0.2516	0.2033	66.6667	80.9524	90.4762	95.2381

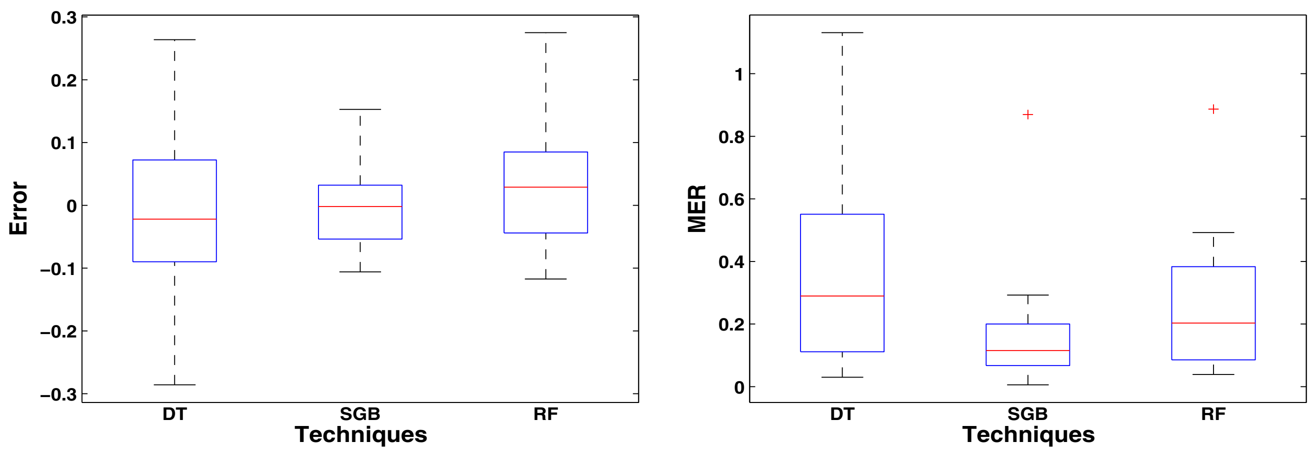

Fig. 7 Boxplot of error and MER values for SPA using DT, SGB and RF techniques

literature. It can be seen that the DT model performs poorly than the existing technique, where as the SGB and RF models outperformed over the existing method proposed by Zia et al. [43].

Table 3 demonstrates the comparison of MMER, MdMER and PRED (x) values for the DT, SGB and RF techniques. This comparative study helps in accurately assessing the performance obtained using these two techniques and proves that the resultant values gather from SGB technique-based effort estimation model which outperforms the values obtained using other models.

Figure 7 displays the box plot of Error and MER values for SPA using 21 project dataset, respectively. These figures help to illustrate the spread and differences of samples, with the help of their corresponding error values generated using DT, SGB and RF models.

In order to affirm the robustness of the proposed models, the effect size test [35] such as Cohen's d and Glass's Δ test between diverse proposed models is processed considering absolute residuals as demonstrated in Table 4 for SPA using 21 project. From the results provided in Table 4, it is evident that the effect size is always small for all the cases.

9 Threats to validity

This section aims to highlight some of the points, which need further analysis.

Table 4 Comparison of effect size test of proposed models for SPA using 21 project dataset

	Cohen's d effect size	Glass's Δ
DT versus SGB	0.0207	0.0215
SGB versus RF	0.1609	0.1396
DT versus RF	0.1450	0.1294

– The different models proposed in this work in order perform effort estimation have been developed by assuming that the value of project velocity of the considered projects is already available. The value of project velocity is computed from the past projects developed by the same group in comparable working conditions. But for a project having no past record of development, it is very difficult to compute the initial project velocity.
– In this study, twenty-one project record dataset from [43] is used for implementation. But there is no clear information available regarding the type of project considered by 43. With a specific end goal to get a substantial result, the proposed work is coveted to be founded on data, which considers various types of softwares obtained using different agile methodologies.
– The dataset considered for implementation is of very small size, i.e., only 21 $\times$ 3, that means the dataset contains only the information about three parameters, i.e., number of story point, project velocity and actual effort values for twenty-one number of records. Hence, the number of record available for testing purpose is very

less. Hence, obtaining optimal results by considering this small dataset is very difficult to be ensured.

10 Conclusion

From the literature review, it is observed that effort estimation of softwares developed using agile methodology is a very tedious task and needs proper analysis due to its changeability feature. Hence, there is a necessity of developing proper estimation techniques for agile software development process. The story point approach is considered by several authors and used for an effective effort estimation of softwares developed using agile methodologies. This approach mostly considered for Scrum model. In this paper, an attempt has been made to apply story point approach in order to estimate the effort required to develop a software using agile methodologies. The results of story point approach are further improved by employing DT, SGB and RF techniques over the story point dataset. The obtained results are then validated and compared with the existing result obtained from Zia et al. [43]. The results demonstrate that the SGB technique outperforms other used machine learning techniques for considered dataset. Extension to this procedure might be made by applying other machine learning techniques such as Extreme Learning Machine and Bayesian Networks on the SPA-related dataset.

References

1. Abrahamsson P, Fronza I, Moser R, Vlasenko J, Pedrycz W (2011) Predicting development effort from user stories. In: Empirical software engineering and measurement (ESEM), 2011 international symposium on, IEEE, pp 400–403
2. Azzeh M, Nassif AB, Minku LL (2015) An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation. J Syst Softw 103:36–52
3. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
4. Britto R, Mendes E, Borstler J (2015) An empirical investigation on effort estimation in agile global software development. In: Global Software Engineering (ICGSE), 2015 IEEE 10th International Conference on. IEEE, IEEE, pp 38–45
5. Coelho E, Basu A (2012) Effort estimation in agile software development using story points. Int J Appl Inf Syst (IJAIS) 3(7)
6. Cohen D, Lindvall M, Costa P (2004) An introduction to agile methods. Adv Comput 62:1–66
7. Foss T, Stensrud E, Kitchenham B, Myrtveit I (2003) A simulation study of the model evaluation criterion mmre. IEEE Trans Softw Eng 29(11):985–995
8. Fowler M, Highsmith J (2001) The agile manifesto. Softw Dev 9(8):28–35
9. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
10. Garg S, Gupta D (2015) PCA based cost estimation model for agile software development projects. In: Industrial Engineering and Operations Management (IEOM), 2015 International Conference on, IEEE, pp 1–7
11. Grapenthin S, Poggel S, Book M, Gruhn V (2014) Facilitating task breakdown in sprint planning meeting 2 with an interaction room: an experience report. In: Software engineering and advanced applications (SEAA), 2014 40th EUROMICRO Conference on, IEEE, pp 1–8
12. Hamouda AED (2014) Using agile story points as an estimation technique in cmmi organizations. In: Agile conference (AGILE), 2014, IEEE, pp 16–23
13. Hearty P, Fenton N, Marquez D, Neil M (2009) Predicting project velocity in xp using a learning dynamic bayesian network model. IEEE Trans. Softw. Eng. 35(1):124–137
14. Hussain I, Kosseim L, Ormandjieva O (2013) Approximation of cosmic functional size to support early effort estimation in agile. Data Knowl Eng 85:2–14
15. Kang S, Choi O, Baik J (2010) Model-based dynamic cost estimation and tracking method for agile software development. In: Computer and information science (ICIS), 2010 IEEE/ACIS 9th International Conference on, IEEE, pp 743–748
16. Keaveney S, Conboy K (2006) Cost estimation in agile development projects. In: ECIS, pp 183–197
17. Kitchenham B, Pickard L, MacDonell S, Shepperd M (2001) What accuracy statistics really measure [software estimation]. IEE Proc Softw 148(3):81–85. doi:10.1049/ip-sen:20010506
18. Lenarduzzi V, Lunesu I, Matta M, Taibi D (2015) Functional size measures and effort estimation in agile development: a replicated study. In: Agile processes, in software engineering, and extreme programming. Springer, Berlin, pp 105–116
19. Mahnic V (2011) A case study on agile estimating and planning using scrum. Elektron ir Elektrotech 111(5):123–128
20. Mahnic V, Zabkar N (2012) Measuring progress of scrum-based software projects. Elektron ir Elektrotech 18(8):73–76
21. Menzies T, Chen Z, Hihn J, Lum K (2006) Selecting best practices for effort estimation. IEEE Trans Softw Eng 32(11):883–895
22. Moreira ME (2013) Working with story points, velocity, and burndowns. In: Being Agile, Springer, Berlin, pp 187–194
23. Morgan JN, Messenger RC (1973) THAID: a sequential analysis program for the analysis of nominal scale dependent variables. Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor
24. Morgan JN, Sonquist JA (1963) Problems in the analysis of survey data, and a proposal. J Am Stat Assoc 58(302):415–434
25. Nassif AB, Capretz LF, Ho D, Azzeh M (2012) A treeboost model for software effort estimation based on use case points. In: Machine learning and applications (ICMLA), 2012 11th international conference on, IEEE, vol 2, pp 314–319
26. Nassif AB, Azzeh M, Capretz LF, Ho D (2013a) A comparison between decision trees and decision tree forest models for software development effort estimation. In: Communications and Information Technology (ICCIT), 2013 Third International Conference on, IEEE, pp 220–224
27. Nassif AB, Ho D, Capretz LF (2013b) Towards an early software estimation using log-linear regression and a multilayer perceptron model. J Syst Softw 86(1):144–160
28. Popli R, Chauhan N (2014) Cost and effort estimation in agile software development. In: Optimization, reliability, and information technology (ICROIT), 2014 international conference on, IEEE, pp 57–61
29. Raslan AT, Darwish NR, Hefny HA (2015) Towards a fuzzy based framework for effort estimation in agile software development. Int J Comput Sci Inf Secur 13(1):37
30. Satapathy SM, Acharya BP, Rath SK (2014a) Class point approach for software effort estimation using stochastic gradient boosting technique. ACM SIGSOFT Softw Eng Notes 39(3):1–6
31. Satapathy SM, Panda A, Rath SK (2014b) Story point approach based agile software effort estimation using various svr kernel

methods. In: The twenty-sixth international conference on software engineering and knowledge engineering, SEKE, pp 304–307
32. Satapathy SM, Acharya BP, Rath SK (2016) Early stage software effort estimation using random forest technique based on use case points. IET Softw 10(1):10–17
33. Schmietendorf A, Kunz M, Dumke R (2008) Effort estimation for agile software development projects. In: 5th Software measurement European Forum, pp 113–123
34. Schweighofer T (2016) How is effort estimated in agile software development projects? In: Fifth workshop on software quality analysis, monitoring, improvement, and applications SQAMIA 2016, p 73
35. Shepperd M, MacDonell S (2012) Evaluating prediction systems in software project estimation. Inf Softw Technol 54(8):820–827
36. Sobiech F, Eilermann B, Rausch A (2016) Using synergies between user stories in scrum. Lect Notes Softw Eng 4(2):91
37. Tanveer B (2016) Hybrid effort estimation of changes in agile software development. In: International conference on agile software development. Springer, Berlin, pp 316–320
38. Trendowicz A, Jeffery R (2014) Software project effort estimation. In: Foundations and best practice guidelines for success, constructive cost model–COCOMO, pp 277–293
39. Ungan E, Cizmeli N, Demirors O (2014) Comparison of functional size based estimation and story points, based on effort estimation effectiveness in scrum projects. In: Software engineering and advanced applications (SEAA), 2014 40th EUROMICRO conference on, IEEE, pp 77–80
40. Usman M, Mendes E, Weidt F, Britto R (2014) Effort estimation in agile software development: A systematic literature review. In: Proceedings of the 10th international conference on predictive models in software engineering, ACM, pp 82–91
41. Usman M, Mendes E, Börstler J (2015) Effort estimation in agile software development: a survey on the state of the practice. In: Proceedings of the 19th international conference on evaluation and assessment in software engineering, ACM, p 12
42. Wu L (1997) The comparison of the software cost estimating methods. http://www.computing.dcu.ie/~renaat/ca421/LWu1.html
43. Zia ZK, Tipu SK, Zia SK (2012) An effort estimation model for agile software development. Adv Comput Sci Appl 2(1):314–324