Int. j. inf. tecnol. (June 2021) 13(3):1101–1110
https://doi.org/10.1007/s41870-021-00669-z

ORIGINAL RESEARCH

A predictive model to estimate effort in a sprint using machine learning techniques

Melvina Autar Ramessur[1] · Soulakshmee Devi Nagowah[1]

Received: 5 May 2020 / Accepted: 11 April 2021 / Published online: 28 April 2021

Abstract Effort estimation is an essential task in a software project as it helps to establish feasible plans for the implementation of a project. It largely influences success or failure of the project. Project planning becomes more efficient with accurate effort estimates, thus providing a number of benefits to the organization. Estimating effort in agile projects has been a challenging task for researchers. Several studies exist in that domain. While some have considered people-related factors, others have catered for project-related factors for estimating effort. Others have adopted machine learning (ML) techniques to produce an accurate estimation. This paper presents a model to estimate and predict effort in a sprint using ML techniques while considering various factors that affect a sprint. The model has been evaluated using various regression algorithms, namely linear regression, K-nearest neighbor, decision tree, polynomial kernel, radius basis function and multi-layer perception (MLP). The model has produced more reliable estimates, with low error values using MLP algorithm.

Keywords Regression algorithm · Prediction · Effort estimation · Agile project · Scrum

✉ Soulakshmee Devi Nagowah
s.ghurbhurrun@uom.ac.mu

Melvina Autar Ramessur
melvina.autar@gmail.com

[1] Department of Software and Information Systems, Faculty of Information, Communication and Digital Technologies, University of Mauritius, Reduit, Mauritius

1 Introduction

Effort estimation is considered as a component of high importance in software project management, more specifically for project planning and monitoring. A project is likely to be completed successfully if effort is estimated accurately. Underestimation may result to schedule and budget overruns, while overestimation may affect the organizational competitiveness [1]. Inaccuracies and inconsistencies exist between estimates due to a number of reasons. One of the reasons is that effort estimation is affected by various factors that can be related to the project, team or working environment. Many industries are increasingly adopting agile methodologies because of the concepts of adaptability and flexibility that allow a software to evolve with the changing needs of user [2]. A recent study has reported that effort estimation of around half of the agile teams are inaccurate by at least 25% [1]. Several methods for effort estimation in agile software development were introduced in past years such as planning poker, analogy, story points and expert judgment [3–8]. With the introduction of ML techniques in the field of software estimation, accurate estimations are possible due to the learning nature of algorithms based on previously completed projects. ML is a field of artificial intelligence that "allows systems to automatically learn and improve from experience without being explicitly programmed" [9]. ML algorithms are constructed in such a way that they can learn from data and output a prediction. Various research works have been conducted on effort estimation using ML techniques [10–12]. According to [13], ML techniques provide better software effort estimation than algorithmic/parametric estimation methods. The paper therefore proposes a predictive model to estimate effort in a sprint by using ML techniques based on specific factors that affect a sprint. This research work is carried out in a scrum environment and focuses on sprint estimation where the team takes the highest priority user story from the backlog and decomposes it into individual tasks that could be estimated. The rest of the paper has the following structure: Sect. 2 describes related work. Section 3 presents the methodology adopted to carry out the research work. Section 4 describes the different factors considered in effort estimation at the sprint level. Section 5 explains the steps used in the construction of a dataset. Section 6 describes the different regression algorithms that will be used for effort prediction. Section 7 shows the behavior of different regression algorithms on the proposed model and the ML algorithm selected for the model. The proposed model is evaluated using different seeds and datasets in Sect. 8. Recommendations and improvements are presented in Sect. 9. Finally, Sect. 10 concludes the paper.

2 Related work

In the past decades, a number of software effort estimation models have been proposed [14]. The popular ones were COnstructive COst MOdel (COCOMO) and Function Point Analysis (FPA). COCOMO is a well-known algorithmic model used for effort estimation [14]. It is based on a regression formula, which considers parameters derived from historical project data and characteristics of the current project [15]. Function Point Analysis (FPA) is another method for predicting application software effort by measuring the functional requirements of a system based on its complexities [14, 16]. Due to the fact that agile software estimation is an iterative process focusing on changing and evolving requirements, traditional estimation methods such as COCOMO and FPA are ineffective for estimating agile software projects [7]. This has given rise to other estimation methods namely analogy [8], expert judgement [17], planning poker [6] and story points [4] as pointed out earlier. However, these methods use subjective expert effort estimation [7]. Researchers have thus investigated on other agile effort estimation techniques to predict effort accurately. This section presents some studies that have performed effort estimation based on factors and those that have adopted ML techniques, more specifically regression techniques, for effort estimation.

2.1 Factor-based effort estimation

Various studies have investigated on factors affecting effort estimation of agile projects. Different algorithmic methods have been proposed to cater for the identified factors in effort estimation. Some methods are described as follows:

2.1.1 *Agile effort estimation using project-related factors*

In this research work, an algorithm known as *Constructive Agile Estimation Algorithm* (CAEA) that incorporates vital factors related to effort estimation of agile projects, is proposed [17]. These factors are typically related to the project namely project domain, project performance, configuration, data transaction, complex processing, operation ease, multiple sites and security. The researchers designed the algorithm by integrating these vital factors with different intensity levels. Small web based application, outsourced contract and critical or military projects with story points are considered as case studies to test the performance of the algorithm.

2.1.2 *Agile effort estimation using project-related and people-related factors*

This research work is based on the CAEA proposed by [17]. In this research work, the researchers demonstrate how project-related factors along with people-related factors impact the effort, duration and cost estimation of a project [2]. Different scenarios are analyzed wherein the level of factors are changed to observe the impact of changing values of factors on the effort, duration and cost estimation. The results show that there is a high dependence of the estimated values on the value of these factors. It is also deduced that estimation is not only affected by project related factors but people related factors like managerial skills, familiarity in team and communication skills are equally important to take into account when estimating a project.

2.1.3 *Generalized agile estimation method*

This research work provides a *Generalized Estimation Method* (GEM) along with an algorithm for estimation of agile projects [3]. This method tries to overcome the limitations of CAEA described by [17]. GEM is an iterative algorithmic approach that improves the estimates after the actual team efficiency and clarity in requirements are known. The GEM based algorithm considers various parameters such as project domain, vital factors, weights and intensity levels of vital factors and different uncertainty levels. Different cases with small projects of three project domains with varying intensity levels of vital factors and uncertainty levels are used to test GEM based algorithm.

2.2 Effort estimation based on ML

Various studies have adopted ML algorithms to produce accurate estimations [35–37]. ML algorithms can be categorized into unsupervised ML and supervised ML. Supervised techniques are divided into two categories, namely regression and classification [18]. Regression algorithms are used to make predictions about numbers. The output variable takes continuous values. Classification algorithms are used to predict group membership for data instances. The output variable takes class labels. Regression and classification algorithms are generally used for classification, learning and estimation problems [13]. In this section, some recent research works related to effort estimation using regression algorithms are discussed.

2.2.1 *Agile software effort estimation using SVR kernel methods*

In this research work, story point approach is used for effort estimation in agile projects [11]. The researchers use Support Vector Regression (SVR) technique to improve the estimates obtained from story points. Four SVR kernel methods namely *Sigmoid Kernel, Linear Kernel*, *Radial Basis Function Kernel* and *Polynomial Kernel* are used to implement the SVR-based model. The inputs of the model consist of total story points and the project velocity based on computation from user stories of 21 developed projects. The output is the effort, that is, the time required to complete the project. The outputs are generated using MATLAB. Mean Magnitude Relative Error (MMRE) and Prediction Accuracy (PRED) are used as performance measures to evaluate the models obtained from the different SVR kernel methods. Based on the generated results, radial basis function (RBF) kernel gives a very high squared correlation coefficient, indicating that a strong relationship between the inputs and the estimated effort. The SVR RBF kernel-based effort estimation model produces lower value of MMRE and higher prediction accuracy value than the other SVR kernel methods.

2.2.2 *Agile software effort estimation using ANN, SVR and RF techniques*

In this research work, size and effort of agile projects are estimated from story points using ML techniques in order to make the estimates more efficient and accurate [12]. Six software houses provided user stories from 21 developed projects [12]. These user stories are used to compute the total story points and their corresponding project velocity. The methods used are Artificial Neural Networks (ANN), SVR and Random Forest (RF). Different metrics like MMRE and Mean Square Error (MSE) are used for performance evaluation. Based on the results obtained, the performance of RF model has proved to be best compared to ANN and SVR models as it provides lower MMRE value.

2.2.3 *Effort estimation using ML techniques with robust confidence intervals*

This research work introduces a method based on ML techniques combined with robust confidence intervals to improve effort estimation accuracy of software projects [10]. The need for building robust confidence intervals is to have an indication on the predictions' reliability. Two datasets namely Desharnais and NASA were used to conduct experiments that compare the performance of different regression algorithms and to prove that robust confidence intervals can be established and built successfully. The regression methods used are Regression-based Trees, MLP and SVR. Bagging predictors were used to decrease the variance of the predictions. It was deduced that M5P/model trees achieves best performance in NASA dataset in terms of MMRE and Bagging and MLP outperforms previous methods in terms of PRED.

After analyzing the different systems, the following research question (RQ) is asked:

> RQ- Can a model be developed that estimates effort based on relevant factors affecting a sprint along with ML to predict effort accurately?

None of the existing systems have combined the factors proposed by the predictive model in this paper to aid in sprint effort estimation and none have investigated on which ML technique gives most accurate effort prediction based on these factors.

3 Methodology

This section presents the methodology that was adopted in an attempt to develop a model defined by the RQ in previous section.

3.1 Collection of sprint factors

A list of factors that influences sprint effort estimation is collected based on existing literature and a survey carried out in a Mauritian company.

3.2 Construction of dataset

Due to confidentiality of data, companies in Mauritius were reluctant to provide their datasets. A dataset was therefore simulated taking into consideration the identified sprint factors. This dataset is used to train different ML techniques for effort estimation.

3.3 Application of different ML algorithms

Different regression algorithms are then applied to the dataset using cross-validation approach. This approach is opted as it overcomes overfitting problems and makes the predictions more general. It divides the dataset into 2 segments: one for training the model and one for validating / testing the model. Ten-fold cross-validation, which is mostly used in data mining, is used on the dataset [19].

3.4 Performance Evaluation

Different experiments were carried out to determine which regression algorithm produces best prediction values based on the simulated dataset and factors.

3.5 Model selection

The model that gives the closest prediction values to the actual effort values was chosen.

3.6 Performance evaluation of selected model with different seeds and datasets

To ensure that the evaluation results of the selected model are not misleading, the selected model is evaluated on the same dataset using cross-validation with random seeds ranging from 1 to 10 and different datasets.

The different steps of the methodology are elaborated in the forthcoming sections.

4 Collection of factors

Several factors affect the effort estimation process in agile projects. [20] have described eighteen factors as listed follows: "Quality of requirements, Team dynamics and commitment, Stakeholders, Data transaction, Communication, Experience of previous project, Process, Responsibilities outside of the project, Staff experience and technical ability, Configuration, Security, Operation Ease, Defects and changes of previous implementation, Working Time, Complexity of requirements, Environmental Factors, Management and Volatility of requirements". A survey was carried out in an IT-based Mauritian company to find out about the extent to which these factors influence a sprint. Fifteen scrum masters have participated in the survey. Based on the survey, it was observed that among the eighteen factors, "Staff experience and technical ability" and "Complexity of requirements" are the two most dominating factors in all projects and thus largely influence effort estimation in a sprint [20]. On the other hand, factors such "Data Transaction, Hardware and Software requirements, Operation Ease, Process, Management, Environmental Factors and Working Time" have less influence on a project's sprint [20]. Therefore to carry out this study on effort estimation in a sprint, twelve dominating factors were taken into consideration namely "Staff experience and technical ability, Stakeholders, Team dynamics and commitment, Communication, Configuration, Experience of previous project, Responsibilities outside project, Volatility of requirements, Complexity of requirements, Defects and changes of previous implementation, Quality of requirements and Security" [20].

5 Construction of dataset

Though it was possible to conduct the survey in a particular company in Mauritius, the company could not provide us with real data due to confidentiality of the data and the company policy. A dataset of consisting of 2100 records is thus simulated using different combination of intensity level of twelve identified factors namely Stakeholders (S), Team dynamics and commitment (TC), Communication (C), Staff experience and technical ability (ST), Experience of previous project (E), Responsibilities outside project (R), Configuration (CF), Security (SC), Defects and changes of previous implementation (D), Quality of requirements (Q), Complexity of requirements (CX) and Volatility of requirements (VR). The dataset is presented in Table 1.

The inputs of the dataset are the number of user stories planned for a sprint, the project domain (PD), an uncertainty factor (UF), impact levels of the twelve identified factors and the simulated actual effort (SAE). The actual effort, in terms of duration in weeks, is simulated using the GEM based algorithm, which works as follows [3]:

1. Assign quantified intensity levels of factors.
2. Assign weights to factors (sum of all weights should not be greater than 1).
3. Calculate priority factor for each factor by multiplying the weight and intensity level of the factor.
4. Compute *Unadjusted Value* (UV) by summing the priority factor of all factors.
5. Calculate the *New Story Point* (NSP) of each story by summing the baseline story point, project domain and the product of UV and an uncertainty factor.
6. Compute size of the project which is the total number of NSP.
7. Calculate the project's duration by dividing the project's size by the team's velocity.

This algorithmic approach is chosen as it incorporates various important parameters that are essential in effort estimation. The dataset presented in Table 1 takes into consideration the following parameters:

Table 1 Dataset

Number of user stories	PD	UF	S	TC	C	ST	E	R	CF	SC	D	Q	CX	VR	SAE
5	1	0.2	1	1	1	1	1	1	1	1	1	1	1	1	41
10	4	0.4	4	9	4	1	4	9	4	1	4	1	9	4	130.24
15	1	0.4	1	1	1	4	4	1	4	1	1	4	4	4	137.52
5	1	0.4	1	1	1	1	1	1	1	1	9	4	4	4	45.72
5	1	0.8	9	4	1	1	1	9	1	1	4	9	9	4	60
10	1	0.2	4	1	1	1	1	1	4	1	1	1	4	1	83.92
10	1	0.2	4	1	1	1	1	1	1	1	4	1	4	1	83.92
5	1	0.2	4	1	1	1	1	1	1	1	1	1	1	1	41.18
15	1	0.2	4	1	1	1	1	1	4	1	4	1	1	1	124.62

5.1 Project domain (PD)

Web application, MIS project and military project are the three types of projects considered in the dataset. The efforts required for each type are quantified using square series, which is the most preferred series in agile estimation [17]. The values 1,4 and 9 were assigned for web application, MIS project and military project respectively. Fibonacci series (2, 3, 5, and so on) can also be used for this purpose.

5.2 Intensity levels (I)

An intensity level defines the impact of a particular factor in a project [8]. Each factor considered in the dataset is assumed to have three levels (Low, Medium and High), mapped with an intensity value using square series. Low level was assigned an intensity value of 1, medium level was assigned an intensity value of 4 and high level was assigned an intensity value of 9.

5.3 Vital factors (v) and weights

The vital factors are the twelve identified factors affecting effort estimation of a sprint in Sect. 4. A weight is assigned to each of the twelve factors. Weight is a value assigned to each factor based on its priority in a project [3]. Factors "Staff experience and technical ability" and "Complexity of requirements" are assigned higher weights than the other factors as they are considered as the most impacting factors [20]. A weight of 0.2 is thus assigned to these two factors and 0.06 to the ten other factors. The weights are assigned in such a way that the aggregation of weights of all factors for a project sums to 1 [3].

5.4 Uncertainty factor (UF)

One of the inputs of the dataset is an uncertainty factor. An uncertainty factor is used to quantify the risk involved in an agile project in terms of requirements, technology and resource aspects. The uncertainty level varies from 0 to 1 [3]. Four levels of uncertainty ranging from 0.2, 0.4, 0.6 and 0.8 are included in the dataset.

5.5 Other parameters

Other parameters that are necessary for sprint-level estimation are the number of user stories planned for the sprint and the baseline story point, which is the number of points for a user story of medium complexity and medium build time. All user stories are assumed to be of the equal size and the baseline story point is taken as 7 [3]. Different number of user stories (5, 10 and 15) are catered in the datasets. The formulas (1), (2) and (3) are used to calculate the actual effort of the dataset. The priority factor for each vital factor is calculated using formula (1). It is the product of the weight and intensity level assigned to each factor. The summation of all priority factors for each occurrence gives the Unadjusted Value using formula (2). The effort, that is, NSP is then computed using formula (3) where SP is the baseline story point, PD is the value set for the project domain, UF is the uncertainty factor of the project and UV is the unadjusted value. The total effort of the sprint is finally calculated as the number of user stories planned for the sprint multiplied by the NSP (N^o of user stories * NSP).

$$PF(vi) = wi * Ii \tag{1}$$

$$UV = \sum PF \underset{i=1}{\overset{n}{(vi)}} \tag{2}$$

$$NSPj = SPj + PD + UF * UV \tag{3}$$

6 Application of ML algorithms for effort prediction

Regression algorithms are used for effort prediction as the expected output of the model should be a number. Therefore, the following regression algorithms namely Linear

Regression (LR), K-Nearest Neighbors (KNN), Decision Tree, SVR and MLP are chosen for experiments on the simulated dataset in Sect. 5. LR makes use of regression line to determine the relationship between a dependent variable and one or more independent variables [21]. KNN is based on "the assumption of locality in data space" [22]. KNN algorithm sets an estimate by using the closest data points. It makes use of local information and forms nonlinear and adaptive decision boundaries for each data point. The algorithm searches for the k nearest neighbor in a training table for every data point to be scored [23]. Decision Tree uses the regression tree logic to create multiple trees in different iterations and selects the best one from all generated trees [24]. Support Vector Machine (SVM) is a learning system that produces prediction functions that are expanded on a subset of support vectors. SVR is a category of SVM. It is grounded on the computation of a linear regression function in a high dimensional feature space where the input data are mapped via a nonlinear function [23]. In order to achieve generalized performance, it attempts to minimize the generalization error bound instead of minimizing the observed training error [25, 26]. MLP network composes of a set of source nodes that consist of the input layer, one or more layers of hidden neurons and the output layer [27]. The model of each neuron in the network comprises of a differentiable nonlinearity at the output end [27]. The network produces a high degree of connectivity determined by the weights of the network. Neural network regression is mainly used for complex problems, which cannot be solved by a traditional regression model [27].

The tool that has been chosen for the proposed model implementation is Waikato Environment for Knowledge Analysis (WEKA). WEKA is a compilation of ML algorithms software that supports various data mining tasks, namely data processing, clustering, classification, regression, visualization and features selection [24]. It is an open source tool, which allows any organizations to access and use it freely. The tool is easy to use and has an intuitive interface. The dataset file was then converted into ARFF (Attribute-Relation File Format) which is a standard input format for a ML analysis tool [8]. The AREF is thus inputted in WEKA.

7 Performance evaluation and model selection

In this section, different experiments are carried out to determine which one of the six regression algorithms produces best prediction values based on the simulated dataset and factors. Cross-validation approach is used to evaluate and compare the different ML algorithms. This approach is opted as it overcomes overfitting problems and makes the predictions more general. It divides the dataset into 2 segments: one for training the model and one for validating / testing the model. Ten-fold cross-validation is mostly used in data mining [19]. Therefore, ten-fold cross-validation is used on the dataset. In ten-fold cross-validation, the data is partitioned into ten equally sized segments. Ten iterations of training and validation are executed. While one fold of the data is used for validation purpose, the outstanding k-1 folds are used for training. After applying the different regression algorithms to the dataset, five actual effort values with their corresponding predicted values from each algorithm are taken out at random from the dataset for analysis. Table 2 compares the actual effort with an estimated effort produced by each algorithm. It is observed that the MLP model gives better prediction values than the other models.

There exist several performance measures to evaluate the accuracy of an effort estimation model. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are two metrics that are regularly employed in model evaluation studies [28]. In recent literature, there have been a lot of debates regarding which of these two metrics should be preferred. However, the conclusions have mainly been based on empirical arguments [29]. WEKA uses both metrics for regression tasks. The results obtained for Correlation coefficient, MAE and RMSE from each regression algorithm are compared are examined in Table 2. MLP model and Decision Tree model have a nearly perfect positive correlation of values 0.9999 and 0.9961 respectively. This means that the relationship between the actual value and predicted value is very strong. They also both have lower error values in terms of MAE and RMSE than the other models. However, MLP model gives the lowest MAE and RMSE values.The graphs generated by each algorithm are also gathered for analysis. Figure 1 shows the actual effort values plotted against the predicted effort values for the different algorithms. The dispersion of data points in the graphs are compared and it is clearly visible that MLP model has higher correlation than other models as the data points are very less dispersed from the regression line. Based on the results and analysis, MLP model is selected for the proposed model as it outperforms the other five models. It exhibits the lowest error value and highest prediction accuracy value.

8 Performance evaluation of selected model

This section presents an evaluation of the selected model using different seeds and datasets.

8.1 Performance evaluation of selected model using different seeds

To ensure that the evaluation results of the selected model are not misleading, the MLP model is evaluated once again

Table 2 Predicted efforts and error values obtained from different ML algorithms

	Actual effort	LR	KNN/IBk	Decision tree (REPTree)	SVR polynomial kernel (PolyKernel)	SVR RBF kernel	MLP (ANN)
1	83.52	85.291	95.28	82.943	85.854	84.154	81.865
2	44.56	30.921	40.92	42.693	34.922	37.802	42.703
3	95.6	90.944	101.2	97.248	90.153	94.921	95.466
4	103.44	129.47	148.44	97.284	125.201	118.61	102.262
5	72.5	79.648	75	74.608	82.25	87.209	71.096
Correlation coefficient		0.9747	0.974	0.9961	0.9744	0.9887	0.9999
MAE		11.6035	10.3608	4.17	11.4369	7.4869	0.5268
RMSE		15.4234	15.6376	6.1169	15.6332	10.8469	0.7055

with the same dataset using cross-validation with random seeds ranging from 1 to 10. A random seed is a number that initializes a pseudorandom number generator [30]. Ten more models are generated and different MAE and RMSE values are obtained for each one. Table 3 shows the variation of MAE and RMSE produced by random seeds of the MLP model. The mean for each metric is calculated based on the error values produced by the random seeds as shown in Table 3. The MAE and RMSE values tend to be close to the mean as they have a standard deviation of 0.86 and 1.03 respectively. Therefore, it can be determined that MLP suits the proposed model best and produces estimates that are very close to the actual effort values, with a low MAE of 0.7402 and RMSE of 1.0538.

8.2 Performance Evaluation of selected model using different datasets

The proposed model is evaluated with different datasets of different weights of factors by using MLP model in order to observe the performance of the model with different datasets. By changing the weights of factors, different effort values are generated for the same combination of levels of factors as used in the previous dataset. Four scenarios as shown in Table 4 are used as experiments to see how the model behaves with different datasets of different weights of factors. Scenario 2 was used for model selection as described in Sect. 4 with a weight of 0.2 assigned to most impacting factors. Additional scenarios 1, 3 and 4 with a weight of 0.1, 0.3 and 0.4 assigned respectively to most impacting factors are used to create three additional datasets. These datasets are used as an experiment to see how the model behaves. Table 4 additionally presents the results obtained for each scenario and the mean value for each metric. It is observed that there is quite a consistency in the predictions of the model with different datasets. The mean value of MAE is 0.7589, which is approximately close to the mean value of MAE of the model that is 0.74021. The same conclusion is observed for RMSE, which has a value of 1.0340 close to that of the model that is 1.05375. The mean value obtained for correlation coefficient is 0.9999, which is same as that of the model, indicating the strength of the relationship between the actual values and predicted values. Therefore, it can be concluded that the prediction accuracy of the proposed model is consistent with different datasets, making it reliable.

9 Recommendations and future improvements

The proposed model is compared against related works adopting ML algorithms described in Sect. 2.2 in Table 5.

Table 5 shows a comparison of different features of existing effort estimation systems using ML techniques. It can be observed that all three existing models do not incorporate factors (people-related or project related) in their estimation. They make use of project velocity in their estimation without considering factors that decelerate the velocity of the team working on the project. Additionally, the studies make use of performance measures that are based on Absolute Error (ActualSP-EstimatedSP). 'ActualSP' is a story's actual effort and 'EstimatedSP' is the predicted effort of the story. The existing three systems have all made use of MMRE for model evaluation. However, several studies have found that MMRE and PRED have an inclination towards underestimation and are unreliable for comparing effort estimation models [31–34]. This paper has therefore proposed a model that incorporates relevant factors that affect effort estimation along with ML techniques. Twelve factors that affect sprints of agile projects are identified for model development based on a survey that have been conducted on 15 small-scaled projects in an agile environment. The estimated value of effort highly depends on the values of these factors. As future works, a survey can be conducted on large-scaled projects consisting of large distributed teams in order to identify other sprint factors such as time zones, work

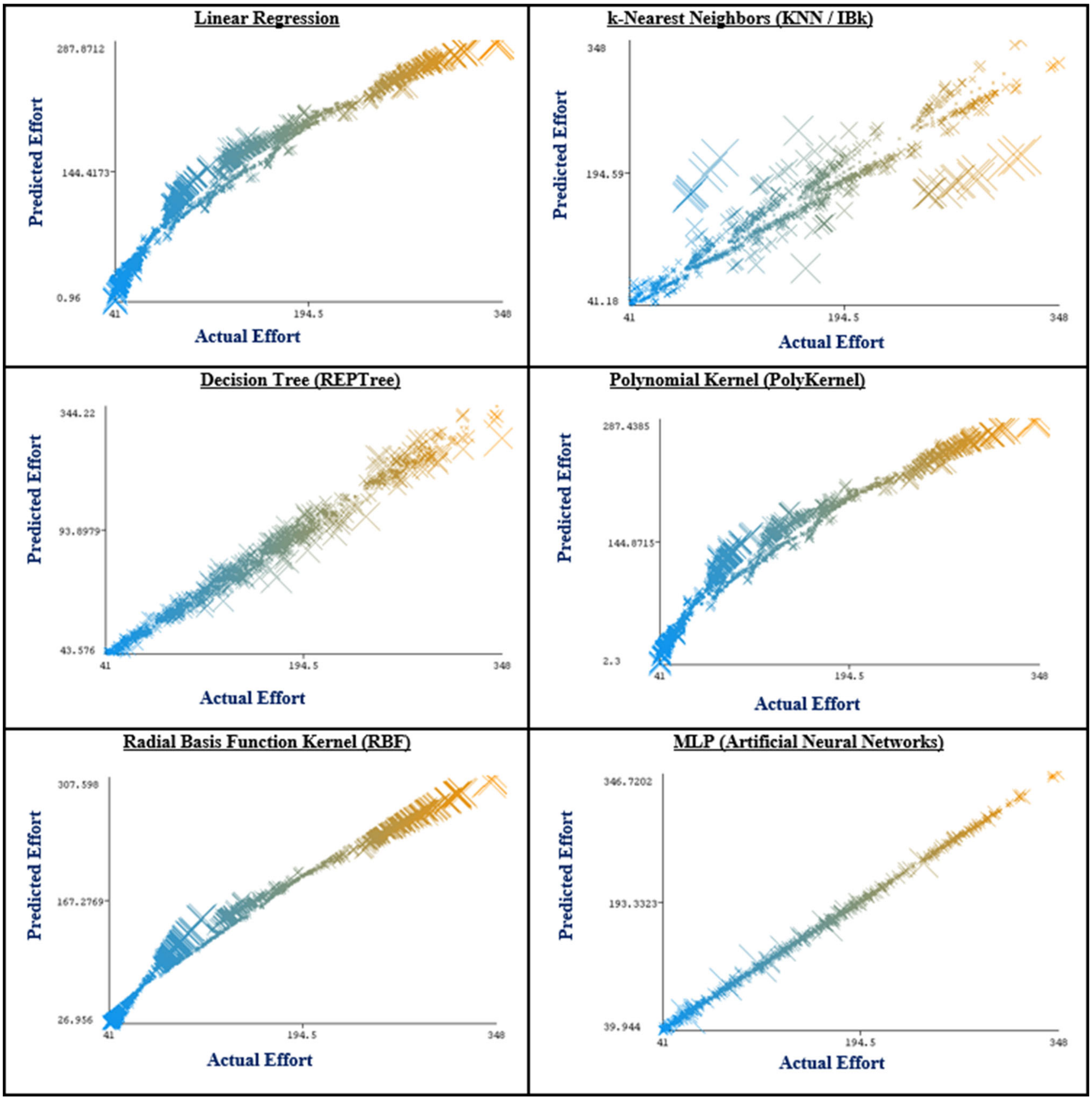

Fig. 1 Actual effort values plotted against the predicted effort values for the different algorithms

Table 3 Mean and Standard Deviation of MAE and RMSE for MLP model

Seed	1	2	3	4	5	6	7	8	9	10	Mean	Standard Deviation
MAE	0.5268	0.7882	0.9018	0.6851	0.5647	0.6241	0.6779	0.8991	1.0498	0.6846	0.7402	0.86
RMSE	0.7055	1.2122	1.3425	0.9346	0.7763	0.8563	0.9422	1.3134	1.4688	0.9857	1.0538	1.03

culture and so on. These factors can be included in the model so that it can produce reliable effort estimation for all sizes of projects.The model is trained and evaluated with data that are simulated based on GEM based algorithm. As further improvements, the model can be trained and evaluated by using real projects' data to ensure that the

Table 4 Factors and associated weights along with metrics

	Factor	Scenario 1 Weight	Scenario 2 Weight	Scenario 3 Weight	Scenario 4 Weight	Mean
1	Stakeholders	0.08	0.06	0.04	0.02	
2	Team dynamics and commitment	0.08	0.06	0.04	0.02	
3	Communication	0.08	0.06	0.04	0.02	
4	Staff experience and technical ability	0.1	0.2	0.3	0.4	
5	Experience of previous project	0.08	0.06	0.04	0.02	
6	Responsibilities outside project	0.08	0.06	0.04	0.02	
7	Configuration	0.08	0.06	0.04	0.02	
8	Security	0.08	0.06	0.04	0.02	
9	Defects and changes of previous implementation	0.08	0.06	0.04	0.02	
10	Quality of requirements	0.08	0.06	0.04	0.02	
11	Complexity of requirements	0.1	0.2	0.3	0.4	
12	Volatility of requirements	0.08	0.06	0.04	0.02	
Correlation coefficient		0.9999	0.9999	0.9999	0.9999	0.9999
MAE		0.7363	0.5268	0.9411	0.8312	0.7589
RMSE		1.0542	0.7053	1.3058	1.0706	1.0340

Table 5 Comparison of existing effort estimation systems with proposed model

Feature	[11]	[12]	[10]	Proposed model
Use of story points for effort estimation	✔	✔	✔	✔
Dataset	21 developed projects	21 developed projects from 6 software houses	2 software datasets (Desharnais and NASA)	4 simulated datasets
Consideration of factors for effort estimation	✘	✘	✘	✔
ML Approach	Cross-validation	Trainlm	Cross-validation	Cross-validation
ML Techniques	SVR Methods	ANN, RF, SVR Methods	M5P algorithm, MLP, SVR method	MLP
ML Tool	MATLAB	MATLAB	unknown	WEKA
Performance measures for model evaluation	MMRE, PRED	MMRE, MSE	MMRE, PRED	MAE, RMSE

prediction accuracy of the model is relevant. In this project, the size of user stories is assumed to be equal but in reality, it is not the case. The baseline story point needs to be adjusted accordingly with different sizes of user stories. Therefore, providing the baseline story point as an input variable to ML may generate better prediction values. This point is not catered in this paper due to limited access to real projects' data. Furthermore, a large volume of data is required to train a model in order to get a precise prediction. The future work can be further expanded by evaluating empirically the impact of the proposed model for effort estimation in practice.

10 Conclusion

This paper provides a predictive model that considers twelve sprint factors for effort estimation by using a ML technique. MLP algorithm is chosen for the model as it outperforms other regression algorithms such as Linear Regression, k-Nearest Neighbors, Decision Tree and Support Vector Regression on the provided datasets. It produces satisfactory and accurate estimates with low error values and high prediction accuracy. Different datasets are used to evaluate the proposed model and the results obtained are reliable. This model provides an automatic way of generating effort estimation with the help of ML, thus saving estimators' time. Estimators of a project can also have a visual representation of the accuracy of the estimates produced. The model additionally provides flexibility on the number of levels of factors and other inputs. Proper planning and monitoring of a project can therefore be achieved with a good initial sprint effort estimation using the proposed model.

References

1. Choetkiertikul M, Dam HK, Tran T, Pham T, Ghose A, Menzies T (2018) A deep learning model for estimating story points. IEEE T Softw Eng 45(7):637–656
2. Popli R, Chauhan N (2012) Research challenges of agile estimation. J Intell Comput Appl.
3. Bahlerao S, Ingle M (2011) Generalized agile estimation method. Int J Adv Sci Eng Inform Technol 1(3):262–267
4. Coelho E, Basu A (2012) Effort estimation in agile software development using story points. Int J Appl Inform Syst (IJAIS) 3(7).
5. Haugen NC (2006) An empirical study of using planning poker for user story estimation. In AGILE 2006 (AGILE'06) 9.
6. Mahnič V, Hovelja T (2012) On using planning poker for estimating user stories. J Syst Softw 85(9):2086–2095
7. Munialo SW, Muketha GM (2016) A review of agile software effort estimation methods. Int J Comput Appl Technol Res 5(9):612–618
8. Sapre AV (2012) Feasibility of automated estimation of software development effort in agile environments (Doctoral dissertation, The Ohio State University).
9. Das S, Dey A, Pal A, Roy N (2015) Applications of artificial intelligence in machine learning: review and prospect. Int J Comput Appl 115(9).
10. Braga PL, Oliveira AL, Meira SR (2007) Software effort estimation using machine learning techniques with robust confidence intervals. In: 7th International conference on hybrid intelligent systems (HIS 2007), pp 352–357.
11. Satapathy SM, Panda A, Rath SK, (2014) Story point approach based agile software effort estimation using various svr kernel methods.
12. Sree SR, Rao P, Mounika M (2017) An early stage software effort estimation in agile methodology based on user stories using machine learning techniques. Int J Res Appl Sci Eng Technol (IJRASET) 5 (XII).
13. Baskeles B, Turhan B, Bener A (2007) Software effort estimation using machine learning methods. In: 2007 22nd international symposium on computer and information sciences, pp 1–6.
14. Kuan S (2017) Factors on software effort estimation. Int J Softw Eng Appl 8(1):23–32
15. Merlo–Schett N, Glinz M, Mukhija A. Seminar on Software Cost Estimation WS 2002/2003. Department of Computer Science 3–19.
16. Borade JG, Khalkar VR (2013) Software project effort and cost estimation techniques. Int J Adv Res Comput Sci Software Eng 3(8).
17. Bhalerao S, Ingle M (2009) Incorporating vital factors in agile estimation through algorithmic method. IJCSA 6(1):85–97
18. Soofi AA, Awan A (2017) Classification techniques in machine learning: applications and issues. J Basic Appl 13:459–465
19. Refaeilzadeh P, Tang L, Liu H (2009) Cross-Validation. Encyclopedia of database systems 5.
20. Ramessur MA, Nagowah SD (2020) Factors affecting sprint effort estimation. In: Advanced Computing and Intelligent Engineering, Springer, Singapore 507–518.
21. Schneider A, Hommel G, Blettner M (2010) Linear regression analysis: part 14 of a series on evaluation of scientific publications. Dtsch Arztebl Int 107(44):776
22. Kramer O (2011) Dimensionality reduction by unsupervised k-nearest neighbor regression. In: 2011 10th International Conference on Machine Learning and Applications and Workshops 1: 275–278.
23. Xie L (2012) KNN classification and Regression using SAS. SD-09.
24. Kalmegh S (2015) Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news. Int J Innov Res Sci Eng Technol 2(2): 438–446.
25. Basak D, Pal S, Patranabis DC (2007) Support vector regression. Neural Inform Process Lett Rev 11(10):203–224
26. Smits GF, Jordaan EM (2002) Improved SVM regression using mixtures of kernels. In: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290) 3: 2785–2790.
27. Li S, Wunsch DC, O'Hair E, Giesselmann MG (2001) Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation. J Sol Energy Eng 123(4):327–332
28. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)?—arguments against avoiding RMSE in the literature. Geosci Model Dev 7(3):1247–1250
29. Brassington G (2017) Mean absolute error and root mean square error: which is the better metric for assessing model performance? EGU Gen Assem Conf Abstr 19:3574
30. Savicky P, Robnik-Šikonja M (2008) Learning random numbers: a Matlab anomaly. Appl Artif Intell 22(3):254–265
31. Foss T, Stensrud E, Kitchenham B, Myrtveit I (2003) A simulation study of the model evaluation criterion MMRE. IEEE T Softw Eng 29(11):985–995
32. Kitchenham BA, Pickard LM, MacDonell SG, Shepperd MJ (2001) What accuracy statistics really measure. IEE Proc Softw 148(3):81–85
33. Korte M, Port D (2008) Confidence in software cost estimation results based on MMRE and PRED. In: Proceedings of the 4th international workshop on Predictor models in software engineering, pp 63–70.
34. Port D, Korte M (2008) Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research. In: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pp 51–60.
35. Pospieszny P, Czarnacka-Chrobot B, Kobylinski A (2018) An effective approach for software project effort and duration estimation with machine learning algorithms. J Syst Softw 137:184–196
36. Singh AJ, Kumar M (2020) Comparative analysis on prediction of software effort estimation using machine learning techniques. Available at SSRN 3565822.
37. Kumar PS, Behera HS, Kumari A, Nayak J, Naik B (2020) Advancement from neural networks to deep learning in software effort estimation: Perspective of two decades. Comput Sci Rev 38:100288