International Journal of Computer Theory and Engineering, Vol. 15, No. 4, November 2023

Machine Learning Based Effort Estimation of Web Applications Using ISBSG Dataset

Manpreet Kaur* and Kanwalvir Singh Dhindsa

Abstract—**The web projects that are completed on time and within budget ascertain a commendable position in the rapidly growing economic web development market. Web Effort Estimation (WEE) estimates the time *t* will take to develop a web application in person-hours or months Expert Opinion algorithmic models, e.g., Constructive Cost Model (COCOMO), and machine learning are the primarily used effort estimating techniques. As current effort estimating techniques face many shortcomings, accurate effort prediction has become a challenging task. To improve prediction accuracy, this work proposes a hybrid approach based on Machine Learning. This approach is validated through an empirical evaluation of the International Software Benchmark Software Group, ISBSG Release 19 dataset. The ISBSG R19 dataset is first pre-processed using machine learning-based linear regression. Secondly, Support Vector Regression (SVR), Decision Tree Regression (DTR), Random Forest regression (RFR), and Ridge Regression (RR) techniques are employed to predict the web effort. The performance of the examined models is evaluated using two commonly used evaluation metrics, Mean Magnitude Relative Error (MMRE) and Prediction accuracy at level 25%, i.e., *Pred(25)*. Then, the statistical significance of effort predicting model producing the highest accuracy and lowest error rates is verified using the Mann-Whitney U test. The performance of the proposed models is also compared with the existing effort estimation models. The results show that the Ridge regression-based model produces exceptionally improved prediction accuracy for web projects in this work.**

Index Terms—**Web development, machine learning, support vector regression, ridge regression**

I. Introduction

Although calculating the effort necessary to develop a web application is difficult, precise estimations of the development effort are critical for the successful management of web-based projects. According to Retail sales reports from 2014 through 2023, there were $3.53 trillion in 2019, with e-commerce revenues expected to reach $6.54 trillion by 2022 [1]. With the growth of practicing web applications, reliable effort estimates are needed to ensure that web projects are completed and delivered on time while remaining within budget [2]. The necessity for employing techniques, standards, and best-practice guidelines to design applications that are delivered on time and under budget grows along with the demand for larger and more complex Web applications.

the demand for larger and more complex Web applications.

In the field of effort estimation for conventional software projects, several methods have been developed, tested, and successfully implemented. However, developing Web applications is different from conventional software projects. Expert Judgement, Algorithmic Models, and Machine learning are the common methods used to predict the effort required to complete a web application project. Most of the Web developers use previous similar project experiences or expert judgment for effort estimates. Examples of algorithmic models are the Constructive Cost Model (COCOMO), and the Software Lifecycle Management Model (SLIM), whereas machine learning techniques are being used in aggregation or as replacements for algorithmic models. These methods include Neuro-fuzzy, Genetic Algorithms, Neural Networks, Fuzzy Logic and Regression trees, Case-based reasoning, Reasoning by Analogy, etc. [3].

Despite numerous effort prediction models in the literature, developing effort estimation models for web applications has become a challenging task. There is an urgent need to improve the prediction accuracy attained by existing models. One of the common confronts while constructing estimating models is incomplete data in historical datasets [4]. The absence of data values in various key project attributes consistently appears which might lead to inaccurate conclusions about the model's precision and analytical ability [5]. However, statisticians also fail to give accurate analytical conclusions due to omitted data from the dataset under statistical analysis [6]. The ISBSG R19 dataset description reveals that the various data fields have a substantial number of missing values. The performance of the effort estimation model may decline due to the lack of complete and consistent datasets. The International Software Benchmark Software Group (ISBSG) provides the dataset with heterogeneity [7] (combining data from several sources), irrelevant data fields, and missing values [8], which might lead to findings that are deceptive about the model's precision and propensity for prediction [4]. In the empirical software engineering literature, the imputation methods for missing values have garnered some practical attention [4, 5]. The default method suggested by existing researchers is to remove data fields with missing values, resulting in a relatively smaller dataset [9]. The quality of the prediction accuracy may be enhanced further if the threat of missing entries is filled by applying specific missing data management strategies [10]. Pre-processing of the dataset also requires attention to maximize the data retention and robustness of effort estimation models [11].

In this paper, the dataset used is an industrial dataset reported to the International Software Benchmarking Standards Group (ISBSG) [12]. In the first step, the pre-processing of the dataset is performed through a sequence of

Manuscript received September 19, 2022; revised December 12, 2022; accepted July 21, 2023.

Manpreet Kaur is with the I. K. Gujral Punjab Technical University, Punjab, India and Department of Computer Science, Hindu College, Punjab, India.

K. S. Dhindsa is with Department of Computer Science and Engineering, Baba Banda Singh Bahadur Engineering College, Punjab, India. E-mail: kdhindsa@gmail.com (K.S.D.)

*Correspondence: manprit.k.dhaliwal@gmail.com (M.K.)

DOI: 10.7763/IJCTE.2023.V15.1347

steps, i.e., Data filtering, Data division, Data Normalization, and Data Scaling. The filtered dataset attained is then preserved to transform the partial dataset into a whole dataset. Mean Imputation is used to handle a slight absence of data, but a high number of missing values in a data field are treated using a machine learning-based approach called linear regression. The effort estimation model based on four machine learning techniques viz., support vector regression, decision tree, random forest, and ridge regression that can contribute significantly to improving the prediction accuracy for developing web applications is employed on the training dataset. Finally, the performance of the proposed effort estimation models is assessed on the testing dataset using various evaluation metrics MMRE and *Pred(25).* In this way, we contribute to exhibiting the low error values and highest accuracy with the implementation of the Ridge regression technique The prediction accuracy of the proposed model is also compared with other existing models. All the experiments conducted in this work exhibit better prediction accuracy as compared to existing outcomes

To analyze the performance of the proposed work systematically, the following research questions are raised:

RQ1. Which ML technique among Support Vector Regression, Decision Tree, Random Forest, and Ridge Regression results in the lowest MMRE error rate?

In the case of web effort estimation models, the prediction accuracy is inversely proportional to the MMRE. This indicates that models with the highest accuracy have the lowest mean magnitude relative error.

RQ2. Which ML techniques employed for Web effort estimation outperform with the highest prediction accuracy in terms of Pred(25)?

The prediction accuracy is directly proportional to the *Pred(25).* This indicates that the predictive models with the highest estimation accuracy have the highest value of *Pred(25).*

RQ3. Does the performance of proposed models largely differ from one another?

The four proposed techniques are employed to evaluate the estimated effort involved in developing web applications using machine learning techniques. It is to be checked if there is some significant difference in the performance of each model.

The rest of the paper is organized as follows: Section II contains a literature review. Section III presents the details of the dataset, and discusses the data pre-processing and the implementation of the model. Section IV describes the results of the evaluation metrics used, discusses the results obtained, and highlights the comparison of the proposed models with other models. The conclusion of this work is provided in Section V, followed by suggestions for further research.

II. Literature Review

Machine learning techniques have been effectively used to compute the development effort for software projects. The related literature projects numerous ML-based models to predict the development effort in the area of web application project management [13].

Researchers conducted a study to predict the web effort by implementing the Case-Based Reasoning (CBR) approach and regression-based statistical methods [6]. It is shown that the stepwise regression resulted in a better approach with the lowest Mean magnitude relative error value, i.e., MMRE= 1.50.

According to the study carried out by Idril *et al.*, a combined effort estimation model, i.e., FRBFNN comprising of Fuzzy Radial Basis Function (FRBF) and Neural Network (NN) with the prediction accuracy of 50.94% was emphasized [14]. A nature-inspired technique Particle Swarm Optimization (PSO) technique effectively optimized the Statistical Regression Equation (SRE) and thus evaluated the development effort for software projects using the repository ISBSG R18 [15]. However, it is found that the issue of incompleteness of the dataset is not addressed at all, despite numerous missing values existing in the dataset.

Shukla and Kumar [8] employ the Extreme Learning Machine (ELM) model for effort prediction of software projects and compare it with Support Vector Regression and Multi-layer perceptron using the dataset ISBSG R19. However, it is observed that the ELM model outperforms other models with a significant difference in all evaluation metrics.

Ordinary Least Squares Regression (OLSR) technique is used for web effort prediction in different empirical studies [11, 16, 17]. However, it is investigated that the machine learning technique Support Vector Regression (SVR) was used for effort prediction of web projects [18], and different kernels of Support vector regression were compared with Stochastic Gradient Boosting (SGB) using the ISBSG R12 dataset. However, the missing data values of the ISBSG R12 dataset are eliminated instead of applying missing data handling techniques [10].

Qamar *et al*. [19] also drew attention to the use of non-algorithmic models and proposed an Artificial Neural Network (ANN) based effort estimation model for web applications. The related work confronts a common limitation of using a small dataset which does not allow effort estimating models as generalized models for other datasets.

Nassif *et al.* [20] compared four effort estimation models: Sugeno linear FL, Sugeno constant FL, Mamdani FL, and MLR. Models were trained and tested using four datasets extracted from ISBSG. Then, the performance of the models was analyzed by applying various unbiased performance evaluation criteria and statistical tests that included: Mean Absolute Error (MAE), Mean Balance Relative Error (MBRE), Mean Inverted Balance Relative Error (MIBRE), Standardized Accuracy (SA), and Scott-Knott. Then, outliers were removed, and the same tests were repeated to draw a conclusion about superior models. The inputs for all models were software size in terms of Adjusted Function Points (AFP), team size, and resource level, while the output was software effort.

Deng and MacDonell [21] conclude that there is a need for greater clarity in describing and justifying the pre-processing, discarding, and retention of data from software engineering data sets. The authors have illustrated how such clarity can be achieved through an example, filtering, formalising, and refining the data in Release 9 of the ISBSG repository in line with an intent to build a predictive model of project-level development effort for FPA-sized projects. The authors emphasize the need to be transparent and to retain as much data as possible.

Palaniswamy *et al*. [22] used the ISBSG dataset for constructing the stacking ensemble model. It is a heterogeneous dataset consisting of software project data from different countries and organizations. The dataset contains 8261 instances and 252 attributes with varying degrees of quality. However, the dataset has many missing values, outliers, categorical data, correlated, and irrelevant features, and needs to be pre-processed before machine learning algorithms can be applied for knowledge extraction. The data pre-processing increases the generalization accuracy of any ML model by a significant degree. The missing values in the ISBSG dataset should be dealt with by Missing Data Treatment (MDT).

Priyavarshini *et al*. [23] conducted a study to compare the random forest algorithm with other machine learning and deep learning algorithms for software effort estimation. Researchers have shown that the Random Forest delivers the best due to its robustness and ability to handle large dataset. Researchers have found the Random Forest as the best performing technique with performance metrics, i.e., Mean Absolute Error (0.20), Root Mean Squared Error (0.25), Mean Square Error (0.067), and R-Squared (0.1280).

This work suggests that there is an utmost requirement to explore machine learning techniques to improve prediction accuracy in the research area of web effort estimation. Researchers in an overview investigated the various machine learning techniques adopted for effort estimation of software projects [7]. They also highlighted the lack of data for analysis in the software development industry. There are only few studies investigating the imputation of missing data in the effort estimation field. Researchers insisted on conducting an empirical investigation of the robustness and accuracy in handling the missing data [7]. The summary of a few studies utilizing the ISBSG dataset is given in Table I which uses major datasets but does not address the issue of missing data values.

TABLE I: SUMMARY OF FEW LITERATURE STUDIES NOT USING IMPUTATION TECHNIQUES ISBSG DATASET

Authors, Year	Dataset Used	Type of Projects	Handling Missing Values	Total Projects	Selected Projects	ML Techniques Used
Mariana *et al*., 2020 [12]	ISBSG R18	Software-based	No	6394	2094	PSO- SRE
Shukla and Kumar, 2021 [13]	ISBSG R19	Software-based	No	9178	4444	ELM
Satapathy and Rath, 2016 [5]	ISBSG R12	Web-based	No	6006	879	SVR, SGB

III. PROPOSED WORK

A. Dataset Description

The ISBSG repository provides the largest accessible cross-company dataset consisting of software projects and web-based projects [23]. The ISBSG R19 includes data on 9178 projects corresponding to 253 data fields. Effort prediction models built using all available variables in the dataset are difficult to construct and unstable to use. According to Deng and MacDonell [21], "the single most important tool in selecting a subset of variables for use in a model is the analyst's knowledge of the area under study". Pre-processing, discarding, and retention of data from software engineering datasets require more clarity in defining and justifying them. The intent is to build a predictive model of development effort. Hence, feature subset selection and pre-processing of the ISBSG R19 dataset are advised to extract the relevant data fields. Table II shows the summary of selected features of the ISBSG R19 dataset. The chosen variables are classified as follows.

Filtering Variable: These variables are used only for pre-processing and filtering the ISBSG R19 dataset.

Independent Variable: These are the variables or factors that affect the value of a dependent variable.

Dependent Variable: Predicted effort which is an outcome of the model, is to be compared with Normalized effort to find the accuracy of the model.

TABLE II: SUMMARY OF SELECTED FEATURES ISBSG R19 DATASET

Parameter Type	Parameter	Meaning	Data Type
Filtering Variables	DQR	Data Quality Rating	Categorical
	UFPRat	Unadjusted FP Rating	Categorical
	DT	Development type	Categorical
	CA	Count Approach	Categorical
	WebD	Web Development	Categorical
Independent Variables	AG	Application Group	Categorical
	DP	Development Platform	Categorical
	LT	Language Type	Categorical
	PPL	Primary Programming language	Categorical
	RL	Resource Level	Categorical
	DBMS	DBMS Used	Categorical
	MDR	Manpower Delivery Rate	Nominal
	PPDR	Productivity Rate	Nominal
	AFP	Adjusted Function points	Nominal
	S_Del	Speed of Delivery	Nominal
	PET	Project Elapsed Time	Nominal
Dependent Variable	NWE	Normalized work effort	Nominal

B. *Methodology*

The proposed methodology is shown in Fig. 1. The proposed work follows several steps to extract high- quality data from the dataset and to evaluate the predicted effort for Web applications using Machine learning techniques. The ISBSG R19 dataset is first loaded. ISBSG R19 dataset includes data on 9178 projects representing 'Records' corresponding to 253 data fields acting as 'Columns'. *Data Filtering* and *Data Division* are performed to select appropriate data fields. A total of 9178 projects (both software and web-based) have been recorded in the ISBSG R19 dataset. Pre-processing of the dataset is performed to address the data quality and to improve the efficiency of the machine learning-based effort estimation model.

Step 1. Data filtering: Web-based projects are the area under study. High-quality projects with ratings "A" and "B" are preferred. All the projects with the same functional size measurement method, i.e., IFPUG 4+ are selected as shown in Table III.

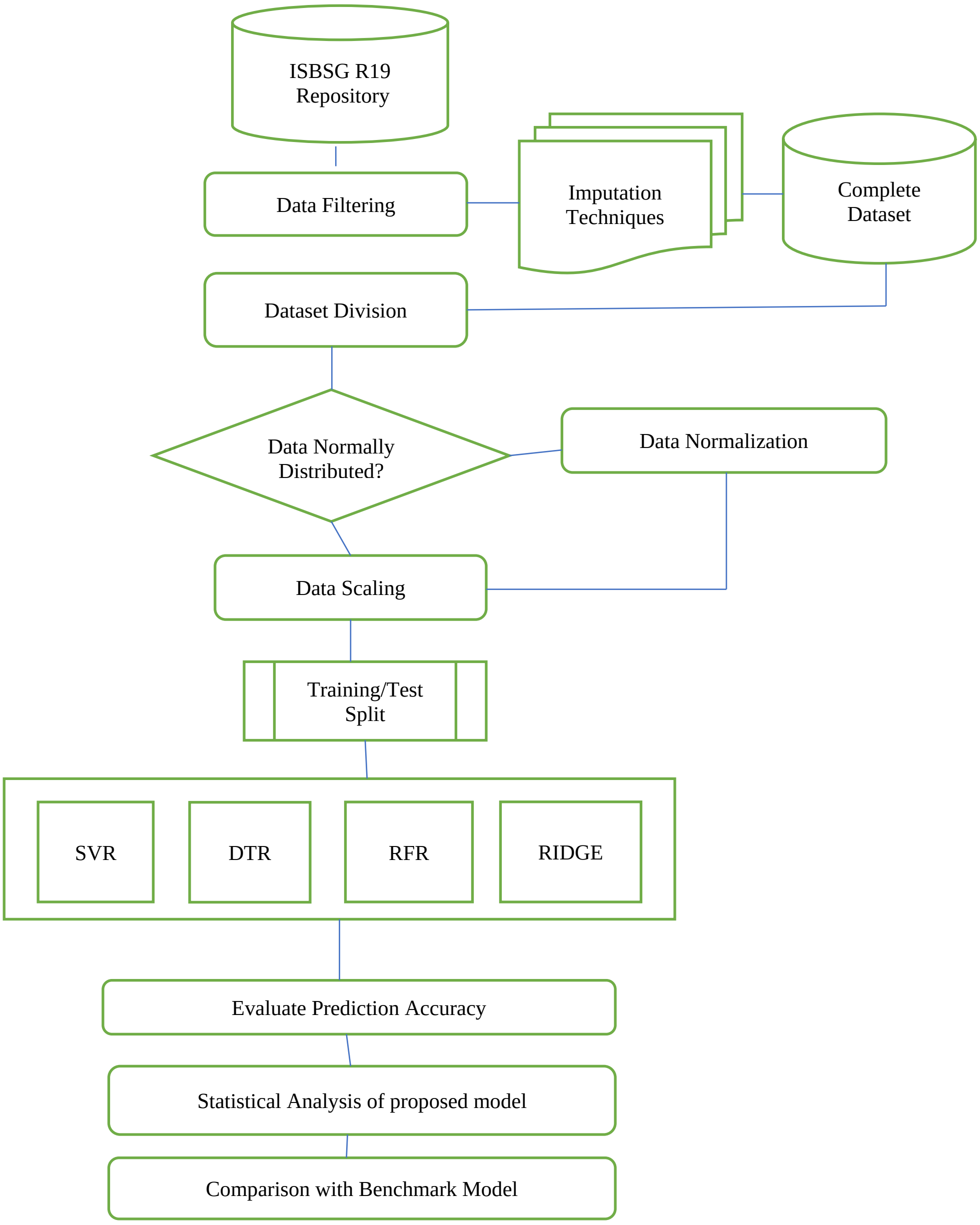

Fig. 1. Proposed methodology for web effort estimation using ML models.

TABLE III: INVESTIGATION OF FILTERING VARIABLES FOR PRE-PROCESSING PROCESS

Filtering Variable	Values	Accepted	Rejected
Web Development (WebD)	Web	1059	8119
Data Quality Rating (DQR)	A or B	1020	39
Unadjusted FP Rating (UFP)	A, B, or C	861	159
Count Approach (CA)	IFPUG 4+	860	1

Step 2. Handling missing data: This work aims to understand the variables and their relationship present in the ISBSG R19 dataset. To handle omitted data present in the dataset, missing data handling techniques depending on the type of data field are implemented. Missing data values are handled depending on the extent of missing data and the

relationship among different data fields.

If a data field with missing values is strongly correlated with another data field, then record/rows are imputed using Linear Regression, a supervised machine learning algorithm.

For non-correlated data fields, with missing data values less than 10–30%, mean imputation is implemented, whereas for data fields with more than 40% omitted values, the data field is discarded for better learning of predictive models.

Step 3. Data division: The division of the filtered dataset is performed based on parameters development type and productivity rate. Redevelopment web projects have been excluded due to their very low count of only 12 web projects as depicted in Table IV.

Software effort to software size is a measure of productivity [19]. Even with the same size metric, productivity in the ISBSG dataset differs unpredictably [23]. For example, the value of productivity varies between 0.4 and 155.7 for projects with the same metric size as IFPUG. For instance, if a project is 100 units in size, the effort needed to develop it can range from 40 (assuming productivity is 0.4) to 15,570 hours (assuming productivity is 155.7) [10]. As shown in Table V, the original dataset is divided into three subsets DATASET1, DATASET2, and DATASET3 depending on the productivity value of each new development and enhancement web project to address this issue.

TABLE IV: DIVISION OF DATASET BASED ON DEVELOPMENT TYPE

Filtering Variable	Type of web project	Accepted/ Rejected	No. of studies
Development Type	New Development	Accepted	273
	Enhancement	Accepted	575
	Redevelopment	Rejected	12

TABLE V: DIVISION OF DATASET BASED ON PRODUCTIVITY RATE (PPDR)

Development type	Subset	Productivity Rate (PPDR)	#Web projects
New Development Web projects	DATASET1	PPDR ≥0 and ≤6.9	115
	DATASET2	PPDR ≥7 and ≤14.9	110
	DATASET3	PPDR ≥15	52
Enhancement Web Projects	DATASET1	PPDR ≥0 and ≤6.9	266
	DATASET2	PPDR ≥7 and ≤14.9	207
	DATASET3	PPDR ≥15	105

TABLE VI: STATISTICAL PROFILE OF ISBSG RELEASE 19

Dataset	Minimum	Maximum	Mean	Skewness	Kurtosis	SD
New Dataset1	83.0	12259.0	1491.1578	3.383	12.903	2193.784
New Dataset2	664.0	18314.0	3765.1192	2.182	4.935	3406.041
New Dataset3	784.0	60826.0	7720.4313	3.882	17.347	9958.740
Enhancement Dataset1	8.0	15712.0	632.4076	7.883	83.274	1261.937
Enhancement Dataset2	28.0	17400.0	648.5482	4.110	23.263	2065.32
Enhancement Dataset3	188.0	21700.0	3644.5196	2.700	8.431	3986.938

Step 4. Data normalization: Based on the values of skewness and kurtosis as shown in Table VI, the data provided in the three subsets of new development and enhancement web projects appear not to be normally distributed. To normalize the data, a logarithmic transformation has been applied to the three subsets of both *New datasets* and *Enhancement datasets*.

Step 5. Scaling of dataset: The scaled values of the input vectors within the range [0,1] are generated by applying *standardization*. For the complete dataset X and data value x, the normalized value x' can be calculated as Eq. (1).

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (1)$$

where x' is the Normalized value of x, which is within the range [0,1], $\min(X)$ is the minimum value of X, $\max(X)$ is maximum value of X.

Step 6. Employ machine learning based effort estimation techniques to compute effort for web applications: The effort estimation model for web applications is constructed using four machine learning techniques viz., Support vector regression, Decision tree regression, Random Forest regression, and Ridge regression are deployed on the pre-processed ISBSG R19 dataset. At a time, only one ML technique can be used to evaluate the effort involved in developing a Web application.

a) Support Vector Regression (SVR): To forecast effort, Support Vector Regression (SVR) uses the Support Vector Machine (SVM, a classification technique) algorithm. It fits the best line within a predetermined or threshold error value, as opposed to conventional linear regression models that aim to minimize the error between the predicted and actual value. SVR divides all prediction lines into two categories: those that cross the error boundary and those that do not. The lines that cross are considered as a potential support vector to forecast an unknown value.

b) Decision Tree Regression (DTR): This model develops a tree-based model for classification as well as regression problems**.** The main idea of the method is to predict the value of the target variable based on the decision rules generated by the attributes. This method divides the dataset into smaller subsets and develops a related decision tree at the time of division. The tree will be generated by recursive partitioning of each node.

c) Random Forest Regression (RFR): The RF regression algorithm is an extension of the Decision Tree algorithm. One of the DT algorithm's main problems is that they are very computationally expensive with the risk of overfitting. Also, they are very sensitive to the training data samples. On changing the training data, the predictions will be different.

So, the RF model combines various decision trees into one to overcome the disadvantages of the decision tree model.

d) Ridge Regression (RR): Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This algorithm works by penalizing the magnitude of coefficients of features along with minimizing the error between predicted and actual observations. It performs L2 regularization in which it adds a penalty equivalent to the square of the magnitude of the coefficients. It evaluates the minimum error using Eq. (2).

$$Minimization\ Objective = Least\ squares\ objective + \alpha \times (sum\ of\ squares\ of\ coefficients) \quad (2)$$

Step 7. To validate the proposed method using evaluation metrics: The performance of effort estimation models is highly dependent on the evaluation accuracy metrics which play a key role in determining the effectiveness of that model underestimation. MMRE is utilized in a large portion of the research work as an assessment standard because of its independence of units characteristics like person-hours, person-months, and so on [24]. MMRE is an important metric used to outline measurements and assess a predictive model. On the other hand, *Pred(25)* is the percentage of estimates that are within 25% of the actual effort [7]. To assess the predicted effort, performance metrics are used as follows.

a) Mean Magnitude of Relative Error (MMRE): The magnitude of relative error (MRE) is defined as given in Eq. (3).

$$MRE = \frac{|E_{actual} - E_{pred}|}{E_{actual}} \quad (3)$$

MMRE is evaluated using the mean of MRE. A good prediction should have MMRE ≤0.25, to denote that the mean estimation error should be less than 25%.

b) Pred(l): Pred(l) is also known as Prediction at level *l*%. It measures the percentage of estimates whose error is less than *l*%, where *l* is set at 25%. It can be defined as explained in Eq. (4).

$$Pred(25) = k/N \quad (4)$$

where k is the number of observations whose MRE is less than or equal to 0.25, and N is the total number of observations. A good prediction model should present a *Pred(25)* ≥0.75, meaning that at least 75% of the predicted values should fall within 25% of actual values [7].

IV. Result and Discussion

A. Performance Comparison

1) Mean Magnitude Relative Error (MMRE)

The mean MRE threshold value of less than 0.25 is considered appropriate [7]. For all three datasets namely DATASET1, DATASET2, and DATASET3 of new development web-based projects, the application of various proposed effort estimation methods yields MMRE values less than 0.25, as shown in Table VII. In other words, for all web-based projects, the difference between the actual effort and the estimated effort w.r.t. actual effort is as small as possible. The mean MRE is inversely proportional to the accuracy of an estimation model.

According to Table VII, the lower value of Mean magnitude relative error for RR-EE explains the reason for less deviation of estimated effort from their corresponding actual effort. The lowest MMRE obtained is 0.0433 only. In general, it is harder to predict effort for enhancement web projects than New development and Re-development projects [19]. Although another effective prediction model using Random Forest regression produces a least mean MRE of 0.12401 for DATASET1 of enhancement type web projects, Ridge regression shows a considerable improvement in prediction accuracy for enhancement web projects of DATASET1 and DATASET2 with a mean MRE of 0.2660 and 0.16859, respectively.

TABLE VII: Mean Magnitude Relative Error (MMRE) for New Development and Enhancement Web Projects

Dataset Type	Dataset	SVR-EE	DTR-EE	RF-EE	RR-EE
New Development	DATASET1	0.1966	0.252	0.1715	0.1711
	DATASET2	0.1134	0.1516	0.1131	0.0822
	DATASET3	0.09041	0.13445	0.0938	0.0433
Enhancement	DATASET1	0.18785	0.17401	0.12401	0.3182
	DATASET2	0.2818	0.33858	0.31446	0.2660
	DATASET3	0.17984	0.24274	0.20594	0.1685

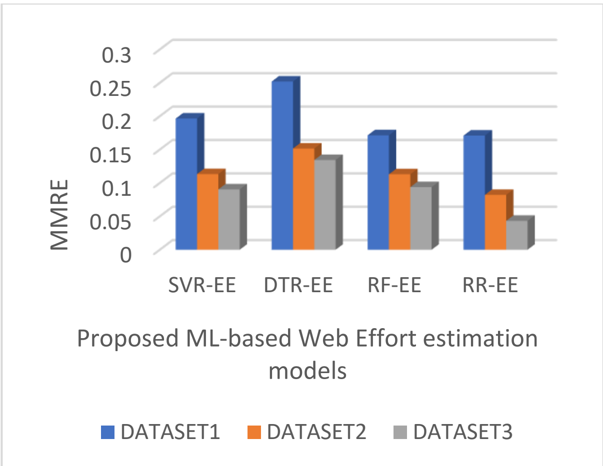

Fig. 2. Comparison of proposed models using mean magnitude relative error for new development projects.

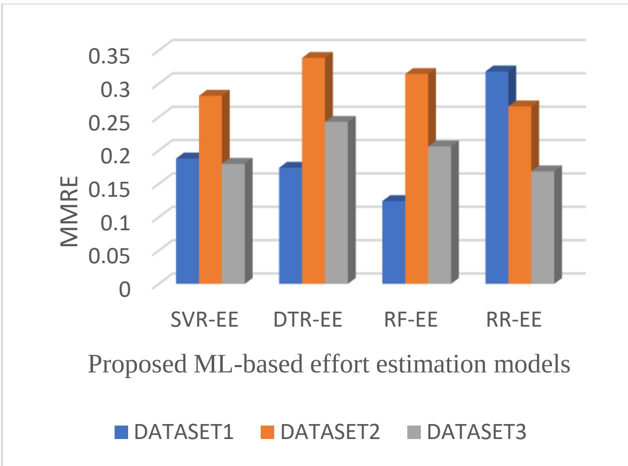

Fig. 3. Comparison of proposed models using mean magnitude relative error for enhancement web projects.

The bar plots in Figs. 2 and 3 show the comparison of mean MRE generated after implementing machine learning-based effort estimation models for new development and enhancement web projects respectively. Results show a significant difference in the performance of the proposed models.

2) Pred(25)

Pred is a crucial evaluation indicator to gauge the effectiveness of regression models. This metric counts the proportion of projects for which an MRE of 0.25 or less was projected. *Pred(25)* greater than or equal to 0.75 is considered to be an acceptable threshold value. For different datasets of New development web projects, Table VIII shows *Pred(25)* values for the Ridge regression-based predictive model point toward higher prediction accuracy. An estimation model's accuracy is typically directly correlated with its *Pred(25)*. The more constructive *Pred(25)* is, the more accurate the predictive model is.

TABLE VIII: PRED (25) FOR NEW DEVELOPMENT AND ENHANCEMENT WEB PROJECTS

DATASET	DATASET	SVR-EE	DTR-EE	RF-EE	RR-EE
New Development	DATASET1	85.88	80.88	83.111	89.55
	DATASET2	100	87.87	100	100
	DATASET3	92.86	89.497	92.368	98.94
Enhancement	DATASET1	79.487	76.923	85.897	57.69
	DATASET2	69.13	62.64	63.56	74.13
	DATASET3	82.12	73.21	71.11	89.98

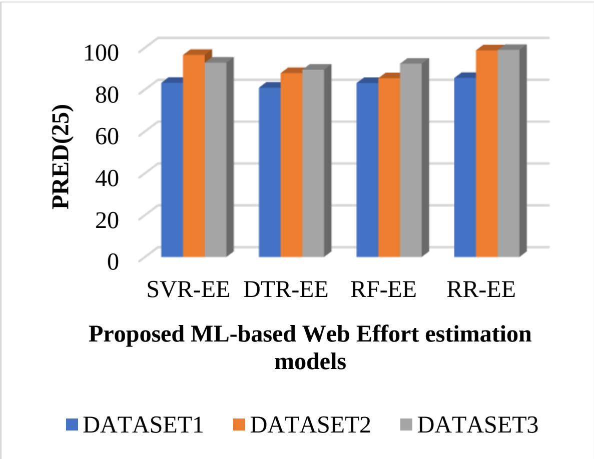

Fig. 4. Comparison of proposed models for New development projects using *Pred(25)*.

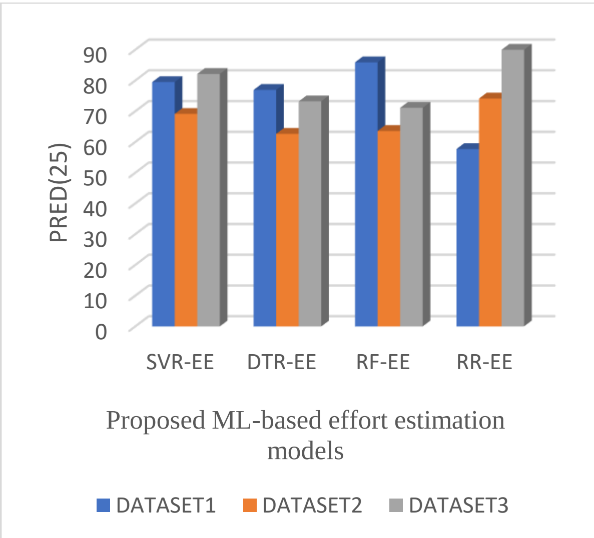

Fig. 5. Comparison of proposed models for Enhancement projects using *Pred(25)*.

Fig. 4 and Fig. 5 demonstrate that the Ridge regression-based effort prediction model performs better for all subsets of new development web projects. For DATASET1, DATASET2, and DATASET3 new development web projects, the greatest prediction accuracy attained by running the RR-EE model is 89.55, 100, and 98.94, respectively. Ridge regression also greatly improves prediction accuracy in the case of enhancement web projects, with values of 74.13 and 89.98 for DATASET2 and DATASET3 respectively, whereas Random Forest regression shows remarkable results for DATASET1.

B. Discussion

In this section, the results obtained in Section IV-A have been discussed and the research questions listed in Section I have been tried to answer.

RQ1. Which ML technique among Support Vector Regression, Decision Tree, Random Forest, and Ridge Regression results in the lowest MMRE error rate?

Fig. 6 demonstrates the performance of proposed models for the three subsets of new development and enhancement web projects using the evaluation metric MMRE. The results indicate that all the new development web projects possess an error rate of less than 0.25. However, in the case of enhancement web projects, all the MMRE values for DATASET1 and DATASET3 reach below the threshold value of less than or equal to 0.25, whereas DATASET2 shows the MMRE value little beyond the threshold.

RQ2. Which ML techniques employed for Web effort estimation outperform with the highest prediction accuracy in terms of Pred(25)?

According to Fig. 7, all the proposed models represent a high *Pred(25)* value of nearly 100%, which means that all datasets possess MRE less than 0.25 for new development web projects, whereas the *Pred(25)* ranges from 69% to 90% for different enhancement web projects. Hence, it can be concluded that the RR-EE model results in the highest *Pred(25)* for new development projects, whereas all models perform alike in the case of enhancement web projects.

RQ3. Does the proposed model with the highest prediction accuracy largely differ from other models?

Fig. 7 remarks the Ridge Regression based model as the highest predictive accuracy model. A non-parametric Mann-Whitney *p*-value test is performed to check if the Ridge regression-based effort estimation model is significantly different from the other three models for all the subsets of the dataset. The null and the alternative hypothesis of the Mann-Whitney U test of the RR-EE model are as follows:

H_0: The RR-EE is not statistically different from models SVR-EE, DTR-EE, and RF-EE.

H_1: The RR-EE model is statistically different from models SVR-EE, DTR-EE, and RF-EE.

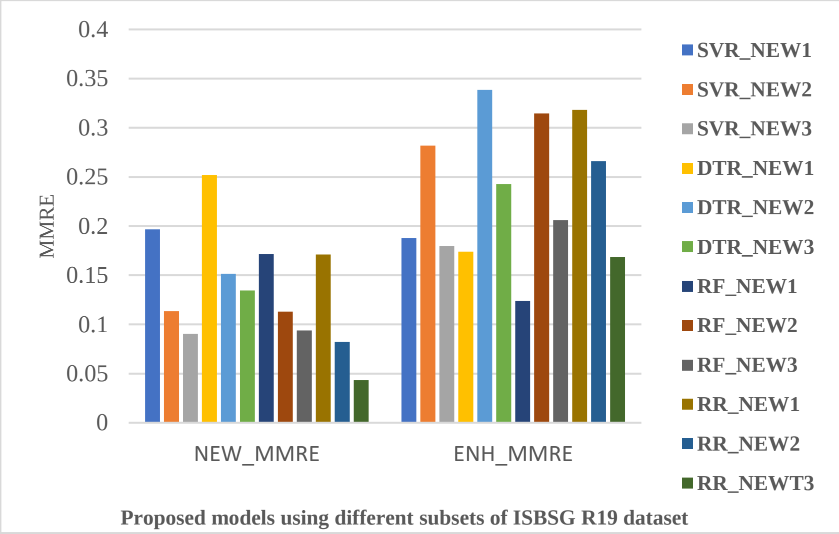

Fig. 6. MMRE of proposed models for New Development and Enhancement web projects.

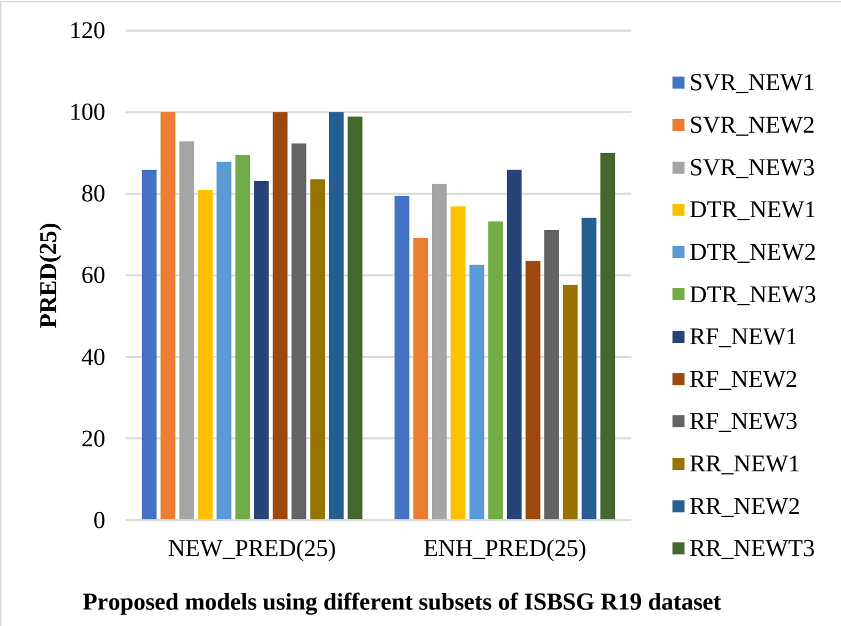

Fig. 7. *Pred(25)* of proposed models for New Development and Enhancement web projects.

TABLE IX: COMPARISON OF STATISTICAL SIGNIFICANCE OF PROPOSED MODELS

Proposed Model	**New Development Web Projects Mann Whitney *p*-value**	**Enhancement Web Projects Mann Whitney *p*-value**
SVR1 vs. RR1	0.03176	0.3534
DTR1 vs. RR1	0.00153	0.9449
RF1 vs. RR1	0.02913	0.1128
SVR2 vs. RR2	0.0586	0.0416
DTR2 vs. RR2	0.0231	0.0666
RF2 vs. RR2	0.3178	0.2537
SVR3 vs. RR3	0.1184	0.0787
DTR3 vs. RR3	0.5637	0.3186
RF3 vs. RR3	0.3363	0.1935

Based on Table IX, a *p*-value less than 0.05 rejects the null hypothesis. The performance of the Ridge regression-based model is statistically different. Thus, the RR-EE model remarkably outperforms DATASET1 and DATASET2 in new development projects and performs identically to other models for DATASET3. However, a p-value not less than 0.05 fails to reject the null hypothesis. Table IX indicates that in the case of Enhancement web projects, RR-EE is not statistically different as compared to SVR-EE, DTR-EE, and RF-EE. Thus, RR-EE performs similarly to other models for most of the enhancement web projects. It can be concluded that all four models based on machine learning techniques perform well for different subsets of the ISBSG R19 dataset.

The results clearly show that implementation of all the proposed models based on ML techniques, i.e., Support Vector Regression, Decision tree regression, Random Forest, and Ridge Regression possess lower MMRE and higher *Pred(25)* concerning the threshold values. However, the lowest error rate and highest prediction accuracy achieved using Ridge regression affirms RR-EE to be the best-performing model in this work.

C. Comparison with Existing Web Effort Estimation Models

In the literature, different dataset repositories like ISBSG R12 [10], tukutuku [18], and freelancing [19] have been used for the effort estimation of web applications. Table X shows the summary of the most commonly used evaluation metric, i.e., Mean Magnitude Relative Error (MMRE) and *Pred(25)* of a few existing effort estimation models proposed by various researchers.

TABLE X: SUMMARY OF MMRE AND PRED(25) VALUES OF RELATED STUDIES

Study	Authors	EE model	Dataset	# of projects	MMRE	Pred(25)
[17]	Corazza *et al*., 2011	SVR-RBF	tukutuku	195	1.10	42.0
[5]	Satapathy and Rath, 2016	SVR-RBF	New Development Dataset1	140	5.6497	36.53
			New Development Dataset2	124	0.5897	68.87
			New Development Dataset3	104	0.3020	39.42
			Enhancement Dataset1	247	1.6863	32.38
			Enhancement Dataset2	163	0.2763	69.32
			Enhancement Dataset3	104	0.5130	30.69
[5]	Satapathy and Rath, 2016	Decision Tree	New Development Dataset1	140	6.3851	26.4286
			New Development Dataset2	124	0.8133	59.3548
			New Development Dataset3	104	1.8570	25.9615
			Enhancement Dataset1	247	1.3448	32.3887
			Enhancement Dataset2	163	0.3957	59.5092
			Enhancement Dataset3	104	1.1192	47.5248
[5]	(Satapathy and Rath, 2016)	Random Forest	New Development Dataset1	140	8.4046	28.8462
			New Development Dataset2	124	2.4085	27.1429
			New Development Dataset3	104	1.0307	50.8065
			Enhancement Dataset1	247	3.0233	26.7206
			Enhancement Dataset2	163	1.6007	38.6503
			Enhancement Dataset3	104	2.1201	38.6139
[5]	Satapathy and Rath, 2016	Stochastic Gradient Boosting	New Development Dataset1	140	5.2902	38.5714
			New Development Dataset2	124	0.6649	76.6129
			New Development Dataset3	104	0.7949	28.8462
			Enhancement Dataset1	247	3.2330	32.3887
			Enhancement Dataset2	163	1.6007	60.1227
			Enhancement Dataset3	104	2.1201	44.5545
[19]	Martino *et al*., 2016	SLR	Italian Company	25	0.29	68.0
[25]	Mendes, 2007	Bayesian network	Tukutuku	150	34.26	-
[26]	Mendes, 2008	MSWR	Tukutuku	65	0.73	10.77
[27]	Lee *et al*., 2016	Bayesian network	-	-	1.90	15.38
[18]	Qamar *et al*., 2018	Neuro-web	freelancers	164	9.92	-
[12]	Tamez *et al*., 2020	PSO-SRE	ISBSG R18(software projects only)	2094	0.61	42.0

No single existing model generates an optimal prediction accuracy for web effort. According to results produced in an existing study [10], the MMRE varies between 0.3020 and 8.4046, whereas *Pred(25)* ranges between 25.96 and 76.61. Mendes in her work [23, 24] has implemented a machine learning-based Bayesian network model resulting in *Pred(25*) of 15.38. The mean MRE achieved by deploying the Neuro-web model is 9.92 which is far higher than the accepted threshold value of 0.25 as indicated in the study [24], whereas the algorithmic model Simple Linear Regression (SLR) results in MMRE of 0.25 and *Pred(25)* of 68.0. However, SLR cannot be assumed to produce an improved result for machine learning-based models due to the small size of the dataset used in the study [16]. Manual Stepwise Regression (MSWR) used in [21] results in a low prediction accuracy of 10.77 to predict the effort involved in web project development.

The performance of effort estimation models is highly dependent on the evaluation accuracy metrics which play a key role in determining the effectiveness of that particular model under consideration. MMRE is utilized in a large portion of the research work as an assessment standard because it is independent of unit characteristics like person-hours, person-months, and so on. MMRE is an important instrument used to outline measurements and assess the web effort estimation model. Table XI shows an enormous percentage decrease in the most widely used statistic metric MMRE by the proposed study. The maximum percent decrease in MMRE value is 97.85%.

TABLE XI. COMPARISON OF MMRE OF PROPOSED MODELS WITH A BENCHMARK STUDY

EE Model	Dataset	MMRE (Benchmark Study [5])	MMRE Proposed Model	Percentage Decrease in MMRE
SVR vs. SVR-EE	New Dataset1	5.6497	0.1966	96.52%
	New Dataset2	0.5897	0.1134	80.76%
	New Dataset3	0.3020	0.09041	70.06%
	Enhancement Dataset1	1.6863	0.18785	88.85%
	Enhancement Dataset2	0.2763	0.2818	1.99%
	Enhancement Dataset3	0.5130	0.17984	64.94%
Decision Tree vs. DTR-EE	New Dataset1	6.3851	0.2520	96.05%
	New Dataset2	0.8133	0.1516	81.35%
	New Dataset3	1.8570	0.13445	92.75%
	Enhancement Dataset1	1.3448	0.17401	87.06%
	Enhancement Dataset2	0.3957	0.33858	14.43%
	Enhancement Dataset3	1.1192	0.24274	78.31%
Random Forest vs. RF-EE	New Dataset1	8.4046	0.17150	97.95%
	New Dataset2	2.4085	0.11310	95.30%
	New Dataset3	1.0307	0.09383	90.89%
	Enhancement Dataset1	3.0233	0.12401	95.89%
	Enhancement Dataset2	1.6007	0.31446	80.35%
	Enhancement Dataset3	2.1201	0.20594	90.28%

Another largely applicable evaluation metric is *Pred(25)* which is the percentage of effort predicted that is within 25% of the actual effort [23] The comparison of *Pred(25)* acquired using proposed models with the benchmark levels marked in the literature is shown in Table XII. The maximum value of *Pred(25)* achieved in the existing study is 69.32% which is far less than the maximum *Pred(25)* of 100% attained using the proposed models. The proposed models give a significant percentage increase in prediction accuracy for all three subsets of New development and Enhancement web projects.

TABLE XII: COMPARISON OF PRED (25) OF PROPOSED MODELS WITH A BENCHMARK STUDY

EE Model	Dataset	*Pred(25)* (Benchmark Study [5])	*Pred(25)* Proposed Model	Percentage Increase in *Pred(25)*
SVR vs. SVR-EE	New Dataset1	36.53	85.88	135.09%
	New Dataset2	68.87	100	45.20%
	New Dataset3	39.42	92.86	135.56%
	Enhancement Dataset1	32.38	79.4892	145.48%
	Enhancement Dataset2	69.32	69.13	0.27%
	Enhancement Dataset3	30.69	82.42	168.55%
Decision Treevs. DTR-EE	New Dataset1	26.4286	92.86	251.36%
	New Dataset2	59.3548	80.88	36.26%
	New Dataset3	25.9615	87.87	238.46%
	Enhancement Dataset1	32.3887	76.9231	137.49%
	Enhancement Dataset2	59.5092	62.64	5.26%
	Enhancement Dataset3	47.5248	73.21	54.04%
Random Forestvs. RF-EE	New Dataset1	28.8462	83.111	188.11%
	New Dataset2	27.1429	100	268.42%
	New Dataset3	50.8065	92.368	81.80%
	Enhancement Dataset1	26.7206	85.8974	221.46%
	Enhancement Dataset2	38.6503	63.56	64.44%
	Enhancement Dataset3	38.6139	71.111	84.15%

The performance comparison of proposed models and benchmark study is also illustrated graphically as represented in Figs. 8 and 9. Fig. 8 demonstrates an enormous fall of error produced in the most widely used statistic metric MMRE by the proposed models, whereas Fig. 9 presents a significant increase in prediction accuracy for all the three subsets of New development and Enhancement web projects. The results obtained in this work show that non-functional requirements of software other than the size of the web project such as Project Elapsed time, Speed of delivery of project, and Team experience, considered in this work ominously recuperate the predicted effort. Also, there is a rigorous need to uncover suitable imputation techniques for handling the missing data values present in the dataset, as large and complete datasets favorably surge the potential of machine learning techniques in estimating the effort of web application development.

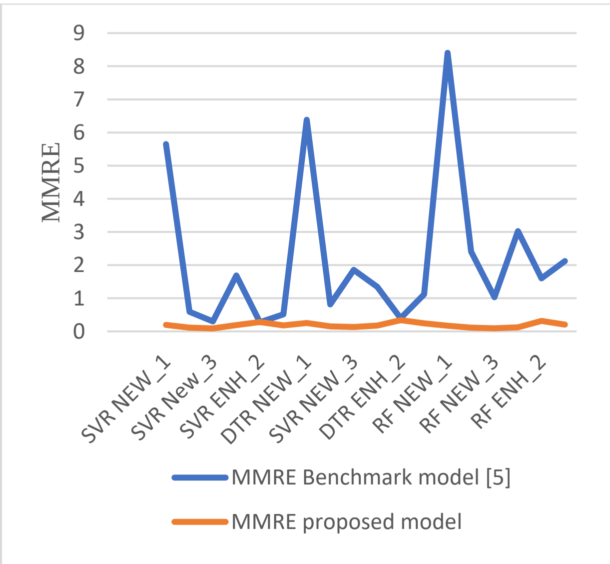

Fig. 8. Comparison of MMRE metric for proposed models with a benchmark study [5].

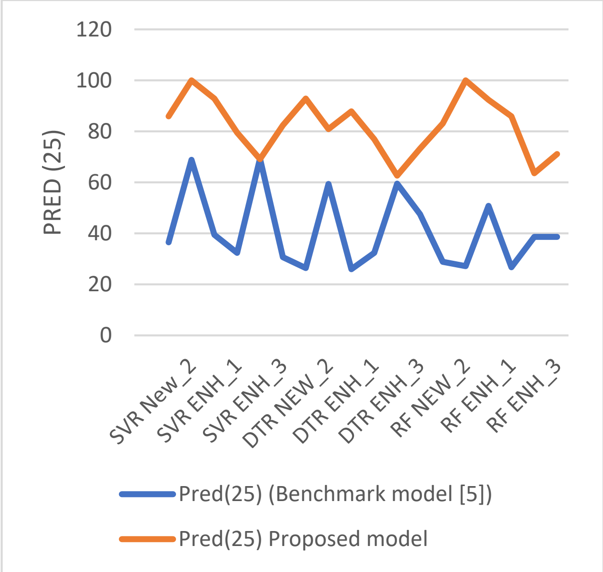

Fig. 9. Comparison of *Pred(25)* for proposed models with a benchmark study.

V. Conclusion

Estimating effort plays a vital role in publicizing a web application on schedule and within budget to the market. Researchers proposed numerous algorithmic models for predicting the effort involved in developing web-based projects amid innumerable challenges. Most of the researchers instigated their models using small-sized single-company datasets. The concerns of pre-processing the dataset and handling missing data values present in the dataset are rarely addressed. The literature does not deliver any assistance for predictors to employ an appropriate effort prediction model. In this research work, effort predictive models viz., SVR-EE, DTR-EE, RFR-EE, and RR-EE based on four machine learning techniques Support vector regression, decision tree, random forest, and ridge regression have been proposed. The largest available cross-company dataset ISBSG R19 is used to implement the proposed models. Pre-processing of the dataset is accomplished through *feature subset selection, data division, data normalization, and data transformation.* Appropriate imputation methods are employed to handle the incompleteness of the dataset. The proposed models were implemented on three subsets of new development and enhancement web projects. The performance of the models is compared using two majorly applied evaluation metrics, i.e., MMRE and *Pred(25).* The results obtained confirm a significant advancement in the performance of the proposed models. However, among various effort estimation models employed, Ridge regression outcomes as the best effort prediction model with the lowest error rates and highest prediction accuracy in this research work.

In the future, this research work will be extended to different work areas. The finest proposed model RR-EE based on Ridge regression will be extended to an automated effort estimation tool. Machine learning algorithms for feature subset selection and handling missing data values present in the datasets will be explored. The proposed models will also be implemented for software projects of the ISBSG dataset.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The authors contributed their sincere efforts to conduct this research work. Manpreet Kaur analyzed the literature from different sources on the web and discussed it thoroughly with K. S. Dhindsa. Manpreet Kaur implemented the proposed algorithms on the dataset provided by the ISBSG community. K. S. Dhindsa evaluated and cross-checked the performance of algorithms and provided valuable feedback. Everyone approved the final version of this research work.

Acknowledgment

The authors are extremely obliged to IK Gujral Punjab Technical University, Kapurthala, Punjab, India for giving this great opportunity to conduct this research work. Also, we are thankful to the ISBSG community that provided us with the largest available dataset based on web projects.

References

[1] R. A.-Q. M. Muhairat and S. Al-Daajeh, "The impact of global software development factors on effort estimation methods," *Eur. J. Sci. Res.*, vol. 46, no. 2, pp. 221–232, 2010.

[2] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A comparative study of cost estimation models for web hypermedia applications," *Empir. Softw. Eng.*, vol. 8, no. 2, pp. 163–196, 2003. doi: 10.1023/A:1023062629183

[3] D. J. Reifer, "Web development : Estimating quick-to-market software," *IEEE Software*, vol. 17, pp. 57–64, 2000.

[4] A. Idri, I. Abnane, and A. Abran, "Missing data techniques in analogy-based software development effort estimation," *J. Syst. Softw.*, vol. 117, pp. 595–611, 2016. doi: 10.1016/j.jss.2016.04.058

[5] P. Sentas and L. Angelis, "Categorical missing data imputation for software cost estimation by multinomial logistic regression," *J. Syst. Softw.*, vol. 79, no. 3, pp. 404–414, Mar. 2006. doi: 10.1016/J.JSS.2005.02.026

[6] E. Mendes and N. Mosley, "Further investigation into the use of CBR and stepwise regression to predict development effort for Web hypermedia applications," in *Proc. 2002 International Symposium on Empirical Software Engineering*, 2002, pp. 79–90. doi: 10.1109/ISESE.2002.1166928

[7] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation," *Softw.-Pract. Exp.*, vol. 52, no. 1, pp. 39–65, 2022. doi: 10.1002/spe.3009

[8] S. Shukla and S. Kumar, "An extreme learning machine based approach for software effort estimation," in *Proc. International Conference on Evaluation of New Approaches to Software Engineering*, 2021, pp. 47–57. doi: 10.5220/0010397700470057

[9] S. H. Arbain, N. A. Ali, and N. H. Mustaffa, "Adoption of machine learning techniques in software effort estimation: An overview," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 551, no. 1, 2019. doi: 10.1088/1757-899X/551/1/012074

[10] S. M. Satapathy and S. K. Rath, "Effort estimation of web-based applications using machine learning techniques," in *Proc. International Conference on Advances in Computing, Communications and Informatics*, 2016.

[11] F. Ferrucci, C. Gravino, and S. D. Martino, "A case study using web objects and COSMIC for effort estimation of web applications," in *Proc. 2008 34th Euromicro Conference Software Engineering and Advanced Applications*, 2008, pp. 441–448. doi: 10.1109/SEAA.2008.60

[12] ISBSG, Guidelines for use of the ISBSG data, 2019.

[13] M. Kaur and S. Sood, "Web effort estimation technique: A systematic literature review," *COMPUSOFT, An International Journal of Advanced Computer Technology*, vol. 8, no. 11, pp. 3462–3471, 2019.

[14] A. Idril, A. Zakranil, M. Elkoutbil, and A. Abran, "Fuzzy radial basis function neural networks for web applications cost estimation," in *Proc. 4th International Conference on Innovations in Information Technology, IIT '07*, 2008, pp. 576–580.

[15] M. D. Alanis-Tamez, "Particle swarm optimization for predicting the development effort of software projects," *J. Math. MDPI*, vol. 8, pp. 1–21, 2020.
[16] F. Ferrucci, C. Gravino, and S. D. Martino, "Estimating web application development effort using Web-COBRA and COSMIC: An empirical study," in *Proc. EUROMICRO*, 2009, pp. 306–312. doi: 10.1109/SEAA.2009.47
[17] S. D. Martino and C. Gravino, "Estimating web application development effort using COSMIC-FFP method," *Int. J. Comput. Appl.*, vol. 31, no. 3, pp. 153–158, 2009. doi: 10.1080/1206212X.2009.11441936
[18] A. Corazza, S. D. Martino, F. Ferrucci, C. Gravino, and E. Mendes, "Investigating the use of support vector regression for web effort estimation," *Empir. Softw. Eng.*, vol. 16, no. 2, pp. 211–243, 2011. doi: 10.1007/s10664-010-9138-4
[19] N. Qamar, F. Batool, and K. Zafar, "Efficient effort estimation of web-based projects using neuro-web," *Int. J. Adv. Appl. Sci.*, vol. 5, no. 11, pp. 33–39, 2018.
[20] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Comput Intell Neurosci.*, 2019. doi: 10.1155/2019/8367214
[21] K. Deng and S. G. MacDonell, "Maximising data retention from the ISBSG repository," in *Proc. 12th Int. Conf. Eval. Assess. Softw. Eng. EASE 2008*, 2008. doi: 10.14236/ewic/ease2008.3
[22] S. K. Palaniswamy and R. Venkatesan, "Hyperparameters tuning of ensemble model for software effort estimation," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 6579–6589, 2021.
[23] S. D. Martino, F. Ferrucci, C. Gravino, and F. Sarro, "Web effort estimation: Function point analysis vs. COSMIC," *Inf. Softw. Technol.*, vol. 72, pp. 90–109, 2016. doi: 10.1016/j.infsof.2015.12.001
[24] D. P. P. B. Marapelli, "Performance tuning for machine learning-based software development effort prediction models," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 21, no. 10, pp. 189–196, 2020.
[25] E. Mendes, "The use of a bayesian network for web effort estimation," *Web. Eng.*, pp. 90–104, 2007.
[26] E. Mendes, "The use of bayesian networks for web effort estimation: Further investigation," in *Proc. 2008 Eighth International Conference on Web Engineering*, Jul. 2008, pp. 203–216. doi: 10.1109/ICWE.2008.16
[27] T. K. Lee, K. T. Wei, and A. A. A. Ghani, "Systematic literature review on effort estimation for Open Sources (OSS) web application development," in *Proc. 2016 Future Technologies Conference (FTC)*, Dec. 2016, pp. 1158–1167. doi: 10.1109/FTC.2016.7821748