Journal of Education and Science (ISSN 1812-125X), Vol: 31, No: 1, 2022 (80-92)

Software Development Effort Estimation Techniques: A Survey

Farah B. Ahmad[1*], Laheeb M. Ibrahim[2]

[1*,2]Department of Software Engineering, College of Computer Sciences & Mathematics, University of Mosul, Mosul, Iraq

E-mail: [1*]farah.20csp11@student.uomosul.edu.iq, [2]laheeb_alzubaidy321966@uomosul.edu.iq

(Received November 24, 2021; Accepted January 17, 2022; Available online March 01, 2022)

DOI: 10.33899/edusj.2021.129868.1156,

ABSTRACT:
Software Effort Estimation (SEE) is used in accurately predicting the effort in terms of (person–hours or person–months). Although there are many models, Software Effort Estimation (SEE) is one of the most difficult tasks for successful software development. Several SEE models have been proposed. However, software effort overestimation or underestimation can lead to failure or cancellation of a project.

Hence, the main target of this research is to find a performance model for estimating the software effort through conduction empirical comparisons using various Machine Learning (ML) algorithms. Various ML techniques have been used with seven datasets used for Effort Estimation. These datasets are China, Albrecht, Maxwell, Desharnais, Kemerer, Cocomo81, Kitchenham, to determine the best performance for Software Development Effort Estimation. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Squared were the evaluation metrics considered. Results and experiments with various ML algorithms for software effort estimation have shown that the LASSO algorithm with China dataset produced the best performance compared to the other algorithms.

Keywords: Software Effort Estimation (SEE); Machine Learning (ML); Random Forest; Decision Tree; Support Vector Machines (SVM).

تقنيات تقدير جهود تطوير البرمجيات:دراسه

فرح باسل احمد[1*]، لهيب محمد ابراهيم[2]

قسم البرمجيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العراق

الملخص:

يتم استخدام تقدير جهد البرمجيات (Software Effort Estimation (SEE)) للتنبؤ وبشكل دقيق بالجهد من حيث (عدد شهور العمل أو عدد ساعات العمل)، وعلى الرغم من وجود العديد من النماذج فإن تقدير جهد البرمجيات يعد من أصعب المهام لتطوير البرمجيات الناجحة. حيث تم اقتراح العديد من نماذج تقدير جهد البرمجيات ومع ذلك ، فإن الإفراط في تقدير جهد البرامج أو النقص في تقدير جهد البرامج يؤدي إلى إلغاء المشروع أو فشل المشروع .

إن الهدف الرئيسي لهذا البحث هو العثور على نموذج أداء لتقدير جهد البرمجيات من خلال إجراء دراسه و مقارنات تجريبية لخوارزميات التعلم الآلي. تم استخدام تقنيات التعلم الالي مع سبع مجموعة بيانات مستخدمة والتي تضمنت china، Albrecht، Desharnais، Maxwell، Kemerer، Cocomo81، Kitchenham ، وذلك لتحديد أفضل أداء لتقدير جهود تطوير

البرمجيات. حيث تم اعتبار الجذر التربيعي لمتوسط الخطأ (RMSE) ومتوسط الخطأ المطلق (MAE) و R-Squared كمقاييس للتقييم التي تم أخذها في الاعتبار. أظهرت النتائج والتجارب مع خوارزميات التعلم الالي المختلفة لتقدير جهد البرمجيات أن خوارزمية LASSO مع مجموعة بيانات china أنتجت أفضل أداء مقارنة مع خوارزميات التعلم الالي الأخرى.

الكلمات المفتاحية : تقدير جهد البرمجيات؛ التعلم الآلي؛ الغابة عشوائية؛ شجره القرار ،دعم آلات النواقل.

1. Introduction

Estimating software efforts is an important part of developing software applications because it allows application development teams to finish the process of development on time and within budget [1]. Despite the complexity of estimating the effort, when done correctly, effort estimating serves as a foundation for all following project planning and management stages. The correct progress of the development depends on the correct estimation. As a result, several studies and research have been conducted with the goal of improving the prediction process and obtaining more exact and dependable findings.

Software effort estimation (and other estimations like software size estimation, software cost estimation, and software schedule estimation) create some certainty and commitment and are used for planning and making decisions for the project (planning budget, investment, and pricing). Figure 1 shows the overlap between the types of software estimation process in the life cycle of software development. [2][23]

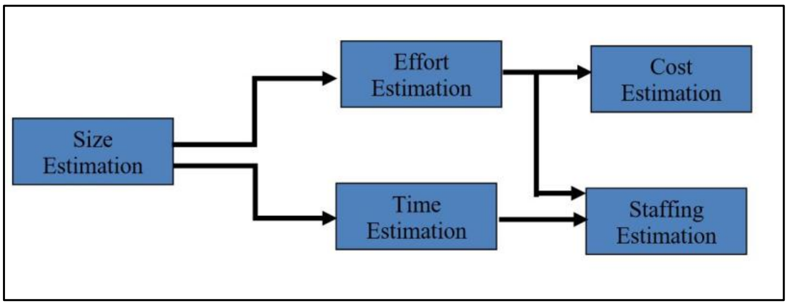

Figure1. Type of software estimation process. [2]

Software estimation consists of the following:

- **Cost Estimation:** Cost estimation plays a significant role in the entire software development cycle, so it is must be completed before the development cycle begins and can continue throughout the software's life cycle [3][24]
- **Effort Estimation:** One of the steps in a software development project that aims to produce high-quality software that can be delivered on time and on budget while meeting the project's requirements, from which the most realistic value of the effort required is calculated to develop or maintain software that is not yet developed on the basis of inputs. [4] [25]
- **Size Estimation:** Size Estimation is defined as determining the resulting size when developing software. Size Estimation is necessary for software project planning and management.[5] [27]
- **Staffing Estimation:** Staffing Estimation is manipulating a suitable model for software project planningto address the problem of human resource allocation [6] [26]
- **Time estimation:** time estimation is regarded as one of success factors in software development. It is associated with the cost of the project. If the time estimation of the project exceeds its deadline and the budget overruns, then we can consider it as failure of management that fails to deliver the project in the estimated time and cost. [7] [28]

Several models for estimating software effort have been proposed [1]. Initially, estimates of software effort are made using expert judgment [46], the use case point approach [49], user stories [50], function

point [51], and analogy-based estimations [48]. Later, for estimation, multiple Machine learning algorithms such as Linear regression, multiple linear regression, logistic regression, ridge regression, neural networks, lasso regression, decision tree, support vector machine, stepwise regression, Navie bayes, Elasticnet regression, random forest, and so on were used [8] [35][44]. The ensemble methods have gained a lot of attention and also produced more predictions than individual effort prediction algorithms [43]. This paper is a survey of techniques for estimating software development efforts.
The following is how this paper is organized. Section 2 is Effort Estimation materials and methods which provides an overview of the Dataset, Evaluation Metrics and Evaluation techniques used in evaluation. Section 3 explains the results and discussion; finally, Section 4 presents conclusions and future work perspectives.

2. Materials and Methods

This section explains Effort Estimation materials and methods which provide an overview of the Dataset, Evaluation Metrics, and Evaluation techniques used in evaluation of Estimation Effort of software.

2.1 Datasets

There are many databases used for estimating effort, which are: China, Kemerer, Cocomo81, Albrecht, Maxwell, Desharnais and Kitchenham. Table 1 displays the repository details of datasets, including the number of Records, attributes, datasets of records, and the output unit of each dataset.

Table 1. Software Effort Estimation [11]

Dataset Name	Records	Attributes	Unit of output
China [38]	499	16	Person-Unit of hours
Kemerer[37]	15	7	Person-Unit of months
Cocomo81 [39]	63	17	Person-Unit of months
Albrecht[41]	24	8	Person-Unit of months
Maxwell[36]	62	27	Person-Unit of hours
Desharnais[40]	81	12	Person-Unit of hours
Kitchenham[42]	145	9	Person-Unit of hours

2.2. Evaluation Metrics

The Evaluation Metrics used to Estimate efforts like Percentage of Predictions (Pred), Root Relative Absolute Error (RRAE), Mean-Square Error (MSE), and Correlation Coefficient (CC), selection and of evaluation metrics can affect the success of the search. Therefore, in this paper more general evaluation metrics are used in the majority of papers, like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R-Squared.[29] [33] [45]

1) **Mean Absolute Error (MAE) [9][32]** [34]
   The average sum of all absolute errors is referred to as MAE.
   Prediction error = (Actual value-Predicted value).
   Absolute error = |Prediction error|.
   $$MAE = \sum_{i=1}^{n} |Actual\ value_i\_Predicted\ value_i| \quad (1)$$

2) **Root Mean Squared Error (RMSE) [10][47]**
   The RMSE is a measure of the Standard Deviation (SD) of the Evaluated deviation [18] and it is calculated using eq (2).
   $$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(X_{obs,i}-X_{model,i}\right)^2}{n}} \quad (2)$$
   Where, X_{obs} _ the actual value and X_{model}_ the expected value[30].

3) **R-Squared [11] [31]**
R-squared is calculated by calculating the Remaining Sum of Squares (RSS) and dividing by the Total Sum of Squares (TSS) and subtracting from 1. It is computed using eq. (3).

$$R - Squared \ = 1 - \frac{RSS}{TSS} \tag{3}$$

2.3 Software Effort Estimation Evaluation Techniques

Estimating software effort is mostly carried out using numerous methodologies proposed by researchers over the last few decades. The function point-based model, the constructive cost model (COCOMO), and the Putnam program life cycle model (SLIM) were the first three approaches to estimate the efforts of widely used software, and they describe the estimation of effort using historical data in specific formula [21] [22].

Estimating the efforts of software by using machine learning (ML) has greatly been used, by applying these ML Algorithms, experts spend less time estimating the proposed project and more time on other functions of the software system that satisfies the client. In the last ten years, machine learning has been used by many researchers to measure effort.

A. Najm, et al. in 2012 used the Bayes networks method for estimating program effort and the StepWise Regression (SWR) and Naive Bayes Classifier (NBC) models [20].

E. Khatibi, et al. in 2015 developed a model using ABE and selective classification, the proposed model greatly improves performance measures besides improving accuracy [19].

A. Idri, et al. in 2015 conducted a Software Development Effort Estimation (ASEE) research published in (1990-2012), and revised based on an automated search in 4 electronic databases [9].

P. Rijwani. et al. in 2016 used a multilayer forward neural network to estimate effort using a back-propagation algorithm for training. They discovered that the neural network model provided better estimates than the COCOMOII computational model [14].

P. Pospieszny et al. in 2018 presented three effective ML algorithms (General Linear Model (GLM), Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP)) ensemble averaging to estimate effort. The results of the ensemble model show that they are very accurate when compared to other approaches used by other researchers [12].

M. Hammad, et al. in 2018 used four machine learning algorithms: K-star, ANN, SVM, and linear regression to estimation effort. The results demonstrated that a ML approach is better for software effort with a low MAE value. [16]

S. Mensah, et al. in 2018 classify the effort of 14 datasets into three categories (low, moderate, and high). They used the quantitative density function, and then built regression models for these datasets. To look for statistically significant differences in the prediction performances of the models they used Robust statistical tests [13].

Z. Abdelali, et al. in 2019 used the validation method. They compared the constructed RF model to the regression tree model and across 3 datasets Tukutuku, ISBSG, and COCOMO. Evaluation metrics used are Pred (0.25), MdMRE, and MMRE. The results showed that the RF model gives a better result [15].

M. Kumar, et al. in 2020 applied LR, MLP and RF algorithms with 12 attributes and the results showed that linear regression computes higher estimation results than other ML techniques [18].

Table 2 describes software effort estimation survey using different algorithms, and also describes the data sets used in each paper for estimation, evaluation metrics, the results from papers, and Table 3 describes the Advantages and Disadvantages of algorithms used to get best results in each paper for effort estimation.

Table 2. A survey of Software Effort Estimation using different algorithms

Papers	datasets	Algorithm	Evaluation measures	Results
P. Pospieszny et al. [12]	ISBSG	Ensemble averaging of three ML algorithms -General Linear model. -Support Vector Machines. -Multi-Layer Perceptron.	-RMSE -MAE -MMER -MSE -MMRE -PRED -MBRE	In ensemble were used multi-layer perceptron (MLP), Support Vector Machines (SVM), and general linear model (GLM) combined for effort estimation. The Ensemble model is better than the Single models [12].
S. Mensah et al. [13]	PROMISE: Desharnais China Cocomo Kemerer Cocomonasa1 Cosmic Maxwell Cocomonasa Kitchenham Industry: php_projects Github: Albrecht Miyazaki Telecom	Regression depended Effort Estimation techniques: -elastic net regression, -ordinary least Squares regression, -LASSO regression, -ridge regression, -stepwise regression.	-MAE -BMMRE -Adjusted R^2	For estimation they considered 14 datasets, and each dataset was grouped into three classes based on the effort attribute, namely low, medium, and high, which is considered the first output, and then six regression models were applied to predict accuracy, which is considered the second output. Elastic net regression outscored the other algorithms in the comparison. [13]
A. Idri et al. [9]	Maxwell ISBSG Kemerer Desharnais Albrecht Abran COCOMO Telecom	-(ASEE) Analogy Based Software Effort Estimation .	-MdMRE - Pred(25) - MMRE.	They compared Analogy Based Software Effort Estimation (ASEE) with 8 ML and Non-ML techniques, including expert judgment, the COCOMO model, Function point analysis, regression, artificial neural network, radial, decision trees, and vector regression support base function. ASEE outperformed eight other techniques and provided greater accuracy based on the evaluation measures used, Pred (25), MdMRE, and MMRE, then accuracy was improved when ML and Non-ML were combined with ASEE [9].
E. Khatibi, et al. [19]	ISBSG	A hybrid estimation model (Analogy Based Estimation) and a weighting system (Outperforms other algorithms) compared with other algorithms like: - Analogy Based Estimation, -Convolutional Neural Network, --Classification And Regression	- MRE -MMRE -PRED (25)	The findings of the suggested model, which included 448 software projects, were compared to those produced using various estimating approaches, showing that the proposed model greatly improves the performance criteria [19].

		Trees, - Multiple Linear Regression.		
P. Rijwani et al. [14]	COCOMO II.	-Multi layered Feed Forward Artificial neural network	- MMRE - MSE	ANN is used for multi-layer Feed forward with back propagation method. Whereas the Multi Layered feed forward Artificial neural network provided better accuracy in prediction efforts [14].
B. Marapelli [4]	COCOMO Nasa2, COCOMO81, COCOMO Nasa.	- K-nearest Neighb, -Linear Regression (LR).	−MSE −MMRE	On the data sets, the results show that the LR model is a good Estimator when compared to K-nearest neighbors: COCOMONASA_2 COCOMONASA,COCOMO81, by getting a higher Correlation Coefficient and a lower RRSE, RAE, RMSE, and MAE [4].
Z. abdelali et al. [15]	ISBSG R8, Tukutuku, COCOMO	-Regression Tree(RT) -Random Forest(RF)	-MdMRE -Pred(25) -MMRE	The results show that the RF model outperforms the RT model on all evaluation metrics, particularly COCOMO and ISBSG R8 [15].
M.Hammad, et al. [16]	PROMISE: Usp05-tf	-Kstar, -Artificial Neural Network, -Support Vector Machines, - Linear Regression	- MAE	SVM has the best prediction accuracy compared to other algorithms as it has the lowest MAE value [16].
M. Kumar, et al. [18]	Desharnais dataset(With all12 attributes) Desharnais dataset(With seven selected attributes)	-Linear Regression, -Random Forest, -Multi-layer perceptron .	−CC −MAE −RMSE −RRAE −RSE	The results by applying LR, MLP and RF algorithms with the 12 attributes showed that LR computed superior estimation results than other ML techniques. Using performance matrices such as CC, MAE, RMSE, RRAE and RSE for LR. When these techniques were used on the Desharnais dataset with only seven attributes selected, results showed that LR allowed a better estimate than MLP and RF [18].
A. Najm , et al.[20]	ISBSG	- forward Stepwise Regression (SWR), -Naive Bayes Classifier (NBC)	-Pred(25)	The NBC model was used to be as useful as the SWR model in terms of prediction accuracy using Pred (25) [20].

Table 3. Advantages and Disadvantages of Algorithms used to estimate software efforts

Papers	Algorithm	Advantages	Dis−advantages
P. Pospieszny et al. [12]	Ensemble averaging of three ML algorithms -General Linear model. -Support Vector Machines. -Multi-Layer Perceptron.	It reduces prediction error, is stable, and produces more accurate results.	They need more time in Computation and design.
S. Mensah et al. [13]	Regression dependent Effort Estimation techniques: -elastic net regression, -ordinary least Squares regression,	Elastic net regression is a simple method and more preferred than RR or LASSO regression.	These methods cannot solve NON-Linear problems but can only solve Linear problems and the computational cost is high.

	-LASSO regression, -ridge regression, -stepwise regression.		
A. Idri et al. [9]	-(ASEE) Analogy Based Software Effort Estimation.	It is similar to human thinking.	It has an inability to properly deal with categorical attributes.
E. Khatibi, et al. [19]	A hybrid estimation model (Analogy Based Estimation) and a weighting system (Outperforms other algorithms) compared with other algorithms like: - Analogy Based Estimation, -Convolutional Neural Network, --Classification And Regression Trees, - Multiple Linear Regression.	Hybrid methods combine multiple models into a best-assembled model, that improve the performance metrics.	Hybrid approaches are computationally more expensive. The proposed model is not appropriate for data sets with a nonlinear relationship between effort and independent project variables.
P. Rijwani et al. [14]	-Multi layered Feed Forward Artificial neural network	Suitable for complex data set and ANN learn from previous data, as well as suitable for both Linear and non-Linear functions, so the voltage prediction results are high.	The problems are overfitting and have slow convergence speed.
B. Marapelli [4]	- K-nearest Neighb, -Linear Regression (LR).	Fast, and it is the simplest way to find relationship between variables, and easy to implement.	This method is able to give a relation between just the linear dependent and independent variables.
Z. Abdelali et al. [15]	-Regression Tree (RT) -Random Forest (RF)	It is robust against overfitting, easy to use, and can handle huge data sets.	It uses black box method. It is time consuming and complicated.
M.Hammad, et al. [16]	-Kstar, -Artificial Neural Network, -Support Vector Machines, - Linear Regression	It can be applied to unstructured or semi-structured data, and it works even better with a lot of features.	It takes more time to predict in larger datasets.
M. Kumar, et al. [18]	-Linear Regression, -Random Forest, -Multi-layer perceptron.	It is a simple algorithm to finding the relationship between many variables that will have designed Many applications with less size or effort, which will reduce the complexity of the software.	This algorithm is able to give a linear relationship between the dependent and independent variables.
A. Najm , et al.[20]	- forward Stepwise Regression (SWR), -Naive Bayes Classifier (NBC)	Naïve Bayes is a simple algorithm to implement; it produces better results when the input variables are independent, and the ability to integrate data with expert knowledge.	This method is always based on the assumption that Input variables are independent and this is not always true.

3. Results and Discussion

Table 4 displays the prediction accuracy results of many machine learning techniques used Albrecht dataset with describing the performance metrics MAE, RMSE and R-Squared for ML algorithms. According to the interpretation from Table 4, the Random Forest algorithm produces better results with lower MAE, RMSE, and higher value of R-Squared.

Table 4. Estimated effort performance metrics of machine learning Algorithms using the Albrecht_dataset[11]

Dataset 1-Albrecht			
Algorithms names used	MAE	RMSE	R-squared
Random Forest	0.1940703	0.2273109	0.4732279
Support vector machines	0.271210	0.291869	0.1315232
Decision Tree	0.2299442	0.3140069	0.0052189
Neuralnet	0.2208348	0.2676081	0.2699029
Ridge	0.2495593	0.274339	0.232714
LASSO	0.2672776	0.2952834	0.1110852
ElasticNet	0.2522743	0.2771694	0.2168
Deepnet	0.2702398	0.3222748	-0.05885081

Table 5 shows performance metrics for ML algorithms that used China dataset with describing the performance metrics MAE, RMSE and R-Squared for ML algorithms. According to the interpretation from Table 5, the LASSO algorithm produces better results with lower MAE, RMSE, and higher value of R-Squared.

Table 5. Estimated effort performance metrics of machine learning Algorithms using the China_dataset[11]

Dataset 2-China			
Algorithms names used	MAE	RMSE	R-squared
Random Forest	0.03832486	0.0651549	0.8028527
Support vector Machines	0.04872961	0.109964	0.4384377
Decision Tree	0.0222997	0.05380968	0.8655325
Neuralnet	0.01775594	0.0439852	0.9101517
Ridge	0.01977087	0.03703651	0.9362975
LASSO	0.01344521	0.02381411	0.9736631
ElasticNet	0.01406866	0.02462756	0.9718331
Deepnet	0.09730134	0.1497018	-0.0407606

Table 6 shows performance metrics for ML algorithms used in Desharnais dataset, and representation of MAE, R-Squared, and RMSE for ML algorithms for the Desharnais dataset. According to Table 6, the results differ because the Neuralnet algorithm produces lower MAE, the Ridge algorithm produces lower RMSE, and the LASSO algorithm produces higher value of R-Squared.

Table 6 Estimated effort performance metrics of machine learning Algorithms using the Desharnais_dataset[11]

Dataset 3-Desharnais			
Algorithms names used	MAE	RMSE	R-squared
Random Forest	0.1161946	0.1744573	0.38059
Support Vector machines	0.1052762	0.1993475	0.1912367
Decision Tree	0.1067227	0.1724028	0.3950932
Neuralnet	0.08394905	0.1508566	0.3950932
Ridge	0.08810373	0.1482106	0.5529472
LASSO	0.08731722	0.1455049	0.5691211
ElasticNet	0.08874032	0.1463786	0.563931
Deepnet	0.1913845	0.2396687	-0.169022

Table 7 represents the performance indicators for ML methods when using Kemerer dataset. Table 7 also represents MAE, R-Squared, and RMSE for ML approaches. According to Table 7, the findings are different because the LASSO method produces lower MAE, while the Random Forest technique produces lower RMSE and greater value of R-Squared.

Table 7. Estimated effort performance metrics of machine learning Algorithms using the Kemerer_dataset [11]

Dataset 4-Kemerer			
Algorithms names used	MAE	RMSE	R-squared
Random Forest	0.2076936	0.2357751	0.6032962
Support vector machines	0.219517	0.2635159	0.5044536
Decision Tree	0.3709271	0.3954624	-0.1160428
Neuralnet	0.2179616	0.3219353	0.2603814
Ridge	0.1971335	0.2810334	0.4363801
LASSO	0.183967	0.256731	0.5296434
ElasticNet	0.1890074	0.26093	0.5141316
Deepnet	0.4011895	0.4142278	-0.2244724

Table 8 describes the performance metrics for ML methods using the Maxwell dataset as an example. For the Maxwell dataset, Table 8 shows MAE, R-Squared, and RMSE vs ML approaches. According to Table 8, the LASSO technique gives better outcomes by producing lower values of MAE, RMSE, and a greater value of R-Squared.

Table 8 Estimated effort performance metrics of machine Learning Algorithms using the Maxwell_dataset[11]

Dataset 5-Maxwell			
Algorithms names used	MAE	RMSE	R-squared
Random forest	0.2330224	0.3098055	0.109088
Support vector machines	0.311288	0.4006657	-0.4901189
Decision Tree	0.2896955	0.3897197	-0.4098123
Neuralnet	0.2245069	0.2914014	0.2117937
Ridge	0.2162983	0.2901234	0.2186921
LASSO	0.2106617	0.2859893	0.2407999
ElasticNet	0.211274	0.2865395	0.2378761
Deepnet	0.2901784	0.3571974	-0.1843314

Table 9 demonstrates performance metrics for ML algorithms using a Kitchenham dataset. Table 9 displays MAE, R-Squared, and RMSE vs ML approaches. According to the interpretation of Table 9, the LASSO method generates better results because it produces lower MAE, whereas the Neuralnet algorithm produces better results because it produces lower RMSE and greater value of R-Squared.

Table 9. Estimated effort performance metrics of Machine learning Algorithms using the Kitchenham-dataset[11]

Dataset 6-Kitchenham			
Algorithms names used	MAE	RMSE	R-squared
Random forest	0.1047124	0.1715596	0.3928102
Support Vector machines	0.1189372	0.2379881	-0.1684357
Decision Tree	0.1174008	0.2013018	0.1640318
Neuralnet	0.03921013	0.0739	0.8873365
Ridge	0.04342892	0.08774596	0.8411641
LASSO	0.03876806	0.07402251	0.8869626
ElasticNet	0.03961122	0.07542022	0.8826536
Deepnet	0.1400501	0.2242306	-0.037252

Table 10 shows performance metrics for ML methods using the Cocomo81dataset as an example. For the Cocomo81 dataset, Table 8 shows a representation of MAE, R-Squared, and RMSE vs ML approaches. According to Table 10, the Neuralnet algorithm offers better results, with lower values of MAE, RMSE, and a greater value of R-Squared.

Table 10 Estimated effort performance metrics of machine learning Algorithms using the Cocomo81_dataset [11]

Dataset 7-Cocomo81			
Algorithms names used	MAE	RMSE	R-squared
Random forest	0.06047924	0.1402015	0.6062928
Support vector machines	0.07500071	0.1976564	0.2174898
Decision Tree	0.08838886	0.1869692	0.2998227
Neuralnet	0.05387155	0.09447618	0.8212226
Ridge	0.07912894	0.1592412	0.4920992
LASSO	0.08097285	0.1766792	0.3747712
ElasticNet	0.07980254	0.1747383	0.3884329
Deepnet	0.2548906	0.285657	-0.6343981

Table 11 shows the ML algorithms that get the best results for each dataset's performance measures. Figure 2 shows a visualization of the best findings for the MAE, RMSE, and R-Squared for each data set, using ML methods. According to Table 11, the LASSO algorithm gives better results in the China data set when compared to other ML algorithms and datasets because it produces lower MAE, RMSE, and a greater value of R-Squared.

Table11. Estimated effort performance metrics of best machine learning algorithms using different dataset[11]

Dataset used	machine learning Algorithms	MAE	RMSE	R-squared
Albrecht	Random Forest	0.1940703	0.2273109	0.4732279
China	LASSO	0.01344521	0.02381411	0.9736631
Desharnais	Neuralnet	0.08394905		
	Ridge		0.1482106	
	LASSO			0.5691211
Kemerer	Random Forest		0.2357751	0.6032962
	LASSO	0.183967		
Maxwell	LASSO	0.2106617	0.2859893	0.2407999
Kitchenham	Neuralnet		0.0739	0.8873365
	LASSO	0.03876806		
Cocomo81	Neuralnet	0.05387155	0.09447618	0.8212226

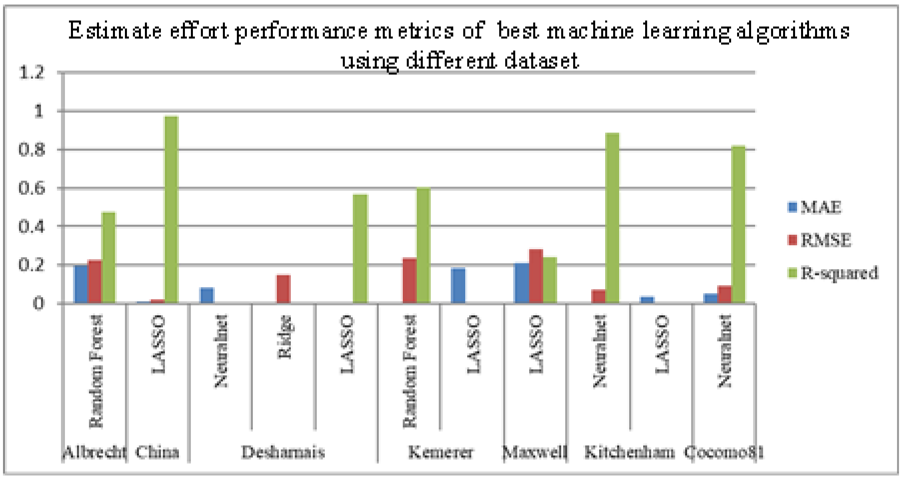

Figure 2. Estimated effort performance metrics of best machine learning algorithms using different datasets

4. Conclusion

Several machine learning algorithms, such as Support Vector Machines, Random Forest, LASSO, Neuralnet, Decision Tree, Ridge, , ElasticNet, and Deepnet, are studied and compared in this paper using the Kitchenham, China, Maxwell, Albrecht, Kemerer, Desharnais, and Cocomo81 datasets for Software Effort Estimate (SEE) predicting the amount of time it takes to develop software in terms of (Person _ hour, or Person _ month). Based on a comparison of multiple ML methods, it was discovered that the Random Forest technique outperforms all other measures in the Albrecht dataset, and in

Desharnais, Neuralnet algorithm produce lower values for MAE, where Ridge algorithm provides lower values of RMSE, and LASSO algorithm produces higher value of R-Squared, in Kemerer different as LASSO algorithm produces lower values for MAE, and Random Forest algorithm produces lower values for RMSE, and produces higher R-Squared value. In Maxwell, the case is different as LASSO algorithm produces lower values for MAE, and Random Forest algorithm produces lower values of RMSE, and produces higher R-S, In Kitchenham, the LASSO algorithm provides better results because it produces lower values of MAE, and the Neuralnet algorithm provides better results because it produces lower values of RMSE and produces higher value of R-Squared, and the Neuralnet algorithm provides better results in Cocomo81 because it produces lower values of RMSE, MAE and produces higher value of R-Squared, where the LASSO algorithms in the China dataset outperforms others. Evaluation metrics considered are MAE, RMSE and R-Squared.

References

[1] S.M.Saif, "Software Effort Estimation for Successful Software Application Development." Tools and Techniques for Software Development in Large Organizations: Emerging Research and Opportunities. IGI Global, 2020. 45-97.

[2] A. Abdulmajeed, M. Al-Jawaherry and T. Tawfeeq, "Predict the required cost to develop Software Engineering projects by Using Machine Learning", *Journal of Physics: Conference Series*, vol. 1897, no. 1, p. 012029, 2021. Available: 10.1088/1742-6596/1897/1/012029.

[3] S.Bilgaiyan, S. Sagnika, S.Mishra, and M. Das, "A Systematic Review on Software Cost Estimation in Agile Software Development." Journal of Engineering Science & Technology Review 10.4 (2017).

[4] B. Marapelli, "Software Development Effort Duration and Cost Estimation using Linear Regression and K-Nearest Neighbors Machine Learning Algorithms", *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2, pp. 1043-1047, 2019. Available: 10.35940/ijitee.k2306.129219.

[5] R. Silhavy, P. Silhavy and Z. Prokopova, "Using Actors and Use Cases for Software Size Estimation", *Electronics*, vol. 10, no. 5, p. 592, 2021. Available: 10.3390/electronics10050592.

[6] W. Chen and J. Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler", *IEEE Transactions on Software Engineering*, vol. 39, no. 1, pp. 1-17, 2013. Available: 10.1109/tse.2012.17.

[7] S. Warda, " Time Estimation as Critical Factor of Software Failure: A Systematic Literature Review Protocol", *i-manager's Journal on Software Engineering*, vol. 13, no. 4, p. 10, 2019. Available: 10.26634/jse.13.4.15775.

[8] A. Varshini, "Predictive analytics approaches for software effort estimation: A review", *Indian Journal of Science and Technology*, vol. 13, no. 21, pp. 2094-2103, 2020. Available: 10.17485/ijst/v13i21.573.

[9] A. Idri, F. Amazal and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review", *Information and Software Technology*, vol. 58, pp. 206-230, 2015. Available: 10.1016/j.infsof.2014.07.013.

[10] A. Nassif, M. Azzeh, A. Idri and A. Abran, "Software Development Effort Estimation Using Regression Fuzzy Models", *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1-17, 2019. Available: 10.1155/2019/8367214..

[11] P. A G, A. K and V. Varadarajan, "Estimating Software Development Efforts Using a Random Forest-Based Stacked Ensemble Approach", *Electronics*, vol. 10, no. 10, p. 1195, 2021. Available: 10.3390/electronics10101195.

[12] P. Pospieszny, B. Czarnacka-Chrobot and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms", *Journal of Systems and Software*, vol. 137, pp. 184-196, 2018. Available: 10.1016/j.jss.2017.11.066..

[13] S. Mensah, J. Keung, M. Bosu and K. Bennin, "Duplex output software effort estimation model with self-guided interpretation", *Information and Software Technology*, vol. 94, pp. 1-13, 2018. Available: 10.1016/j.infsof.2017.09.010.

[14] P. Rijwani and S. Jain, "Enhanced Software Effort Estimation Using Multi Layered Feed Forward Artificial Neural Network Technique", *Procedia Computer Science*, vol. 89, pp. 307-312, 2016. Available: 10.1016/j.procs.2016.06.073.

[15] Z. abdelali, H. Mustapha and N. Abdelwahed, "Investigating the use of random forest in software effort estimation", Procardia Computer Science, vol. 148, pp. 343-352, 2019. Available: 10.1016/j.procs.2019.01.042.

[16] M.Hammad, and A.Alqaddoumi, "Features-level software effort estimation using machine learning algorithms." 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). IEEE, 2018.

[17] P. Singal, A. Kumari and P. Sharma, "Estimation of Software Development Effort: A Differential Evolution Approach", *Procedia Computer Science*, vol. 167, pp. 2643-2652, 2020. Available: 10.1016/j.procs.2020.03.343.

[18] M. Kumar and A. Singh, "Comparative Analysis on Prediction of Software Effort Estimation Using Machine Learning Techniques", *SSRN Electronic Journal*, 2020. Available: 10.2139/ssrn.3565813.

[19] E. Khatibi and V. Khatibi Bardsiri, "Model to estimate the software development effort based on in-depth analysis of project attributes", *IET Software*, vol. 9, no. 4, pp. 109-118, 2015. Available: 10.1049/iet-sen.2014.0169.

[20] S. Elyassami, "Investigating Effort Prediction of Software Projects on the ISBSG Dataset", *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 2, pp. 121-132, 2012. Available: 10.5121/ijaia.2012.3210.

[21] A. Najm, A. Zakrani and A. Marzak, "Systematic Review Study of Decision Trees based Software Development Effort Estimation", *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020. Available: 10.14569/ijacsa.2020.0110767.

[22] A. Ali and C. Garvin, "A systematic literature review of software effort prediction using machine learning methods", Journal of Software: Evolution and Process, vol. 31, no. 10, 2019. Available: 10.1002/smr.2211.

[23] "Advances in Machine Learning and Computational Intelligence" , Springer Science and Business Media LLC, 2021

[24] D. Puja, "Intelligent Method for Cost Estimation during Software Maintenance", *International Journal Of Engineering And Computer Science*, 2017. Available: 10.18535/ijecs/v6i5.11.

[25] A. Sharma and D. Kushwaha, "Estimation of Software Development Effort from Requirements Based Complexity", *Procedia Technology*, vol. 4, pp. 716-722, 2012. Available: 10.1016/j.protcy.2012.05.116.

[26] N. Mohamed, A. Al-Qasmi, S. Al-Lamki, M. Bayoumi and A. Al-Hinai, "An estimation of staffing requirements in primary care in Oman using the Workload Indicators of Staffing Needs method", *Eastern Mediterranean Health Journal*, vol. 24, no. 09, pp. 823-829, 2018. Available: 10.26719/2018.24.9.823.

[27] X. MA, Y. WANG and Z. ZHENG, "Network Size Estimation Method Based Semantic Attraction", *Journal of Software*, vol. 23, no. 3, pp. 662-676, 2012. Available: 10.3724/sp.j.1001.2012.03990.

[28] J. Ning and K. Bandeen-Roche, "Estimation of time-dependent association for bivariate failure times in the presence of a competing risk", *Biometrics*, vol. 70, no. 1, pp. 10-20, 2013. Available: 10.1111/biom.12110.

[29] A. Priya Varshini, K. Anitha Kumari, D. Janani and S. Soundariya, "Comparative analysis of Machine learning and Deep learning algorithms for Software Effort Estimation", *Journal of Physics: Conference Series*, vol. 1767, no. 1, p. 012019, 2021. Available: 10.1088/1742-6596/1767/1/012019.

[30] Y. Mahmood, N. Kama and A. Azmi, "A systematic review of studies on use case points and expert-based estimation of software development effort", *Journal of Software: Evolution and Process*, vol. 32, no. 7, 2020. Available: 10.1002/smr.2245.

[31] M. Nazari and S. Shamshirband, "The particle filter-based back propagation neural network for evapotranspiration estimation", *ISH Journal of Hydraulic Engineering*, vol. 26, no. 3, pp. 267-272, 2018. Available: 10.1080/09715010.2018.1481462.

[32] A. Baareh, "Optimizing Software Effort Estimation Models Using Back-Propagation Versus Radial Base Function Networks", *Journal of Computer Science*, vol. 15, no. 3, pp. 321-331, 2019. Available: 10.3844/jcssp.2019.321.331.

[33] A. Kusiak and Z. Zhang, "Control of wind turbine power and vibration with a data-driven approach", *Renewable Energy*, vol. 43, pp. 73-82, 2012. Available: 10.1016/j.renene.2011.11.024.

[34] E. ERTUĞRUL, Z. BAYTAR, Ç. ÇATAL and Ö. MURATLI, "Performance tuning for machine learning-based software development effort prediction models", *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, pp. 1308-1324, 2019. Available: 10.3906/elk-1809-129.

[35] M. Hammad and A. Alqaddoumi, "Features-Level Software Effort Estimation Using Machine Learning Algorithms," *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2018, pp. 1-3, doi: 10.1109/3ICT.2018.8855752.

[36] Boetticher, G., Menzies, T., Ostrand, T. PROMISE Repository of empirical software engineering data http://promisedata.org/ repository, West Virginia University, Department of Computer Science.

[37] S. J. Shirabad, and T. J. Menzies, 2005. "The PROMISE Repository of Software Engineering Databases," School of Information Technology and Engineering, University of Ottawa, Canada. Available: http://promise.site.uottawa.ca/SERepository

[38] "DASE/440_evaluationInSoftEng.Rmd at master · danrodgar/DASE", *GitHub*, 2021. [Online]. Available: https://github.com/danrodgar/DASE/blob/master/440_evaluationInSoftEng.Rmd. [Accessed: 08- Dec- 2021].

[39] *Promise.site.uottawa.ca*, 2021. [Online]. Available: http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff. [Accessed: 08- Dec- 2021].

[40] "Desharnais dataset", *Kaggle.com*, 2021. [Online]. Available: https://www.kaggle.com/toniesteves/desharnais-dataset. [Accessed: 08- Dec- 2021].

[41] "UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set", *Archive.ics.uci.edu*, 2021. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset. [Accessed: 08- Dec- 2021].

[42] *Promise.site.uottawa.ca*, 2021. [Online]. Available: http://promise.site.uottawa.ca/SERepository/datasets/Kitchenham.arff. [Accessed: 08- Dec- 2021].

[43] O. Malgonde and K. Chari, "An ensemble-based model for predicting agile software development effort", Empirical Software Engineering, vol. 24, no. 2, pp. 1017-1055, 2018. Available: 10.1007/s10664-018-9647-0.

[44] H. De Carvalho, R. Fagundes and W. Santos, "Extreme Learning Machine Applied to Software Development Effort Estimation", *IEEE Access*, vol. 9, pp. 92676-92687, 2021. Available: 10.1109/access.2021.3091313.

[45] M. Sadiq, M. Asim, J. Ahmed, V. Kumar and S. Khan, "Prediction of Software Project Effort Estimation: A Case Study", *International Journal of Modeling and Optimization*, pp. 37-43, 2011. Available: 10.7763/ijmo.2011.v1.7.

[46] M. Jorgensen, B. Boehm and S. Rifkin, "Software Development Effort Estimation: Formal Models or Expert Judgment?", IEEE Software, vol. 26, no. 2, pp. 14-19, 2009. Available: 10.1109/ms.2009.47.

[47] R. Mehdi, "Software Effort Estimation Using Adaptive Fuzzy-Neural Approach", *International Journal of Computer Applications Technology and Research*, vol. 6, no. 07, pp. 329-332, 2017. Available: 10.7753/ijcatr0607.1011.

[48] J. Li, G. Ruhe, A. Al-Emran and M. Richter, "A flexible method for software effort estimation by analogy", *Empirical Software Engineering*, vol. 12, no. 1, pp. 65-106, 2006. Available: 10.1007/s10664-006-7552-4.

[49] M. ManzoorKirmani and A. Wahid, "Use Case Point Method of Software Effort Estimation: A Review", *International Journal of Computer Applications*, vol. 116, no. 15, pp. 43-47, 2015. Available: 10.5120/20415-2835.

[50] J. Choudhari and U. Suman, "Story Points Based Effort Estimation Model for Software Maintenance", *Procedia Technology*, vol. 4, pp. 761-765, 2012. Available: 10.1016/j.protcy.2012.05.124.

[51] S. Lee, J. Gang and J. Park, "Software Development Effort Estimation Using Function Point", *The KIPS Transactions:PartD*, vol. 9, no. 4, pp. 603-612, 2002. Available: 10.3745/kipstd.2002.9d.4.603.