Software Project Estimation with Machine Learning

Noor Azura Zakaria[1], Amelia Ritahani Ismail[2], Afrujaan Yakath Ali[3]
Nur Hidayah Mohd Khalid[4], Nadzurah Zainal Abidin[5]
Department of Computer Science
Kulliyyah of Information and Communication Technology
International Islamic University Malaysia, Kuala Lumpur, Malaysia

***Abstract*—This project involves research about software effort estimation using machine learning algorithms. Software cost and effort estimation are crucial parts of software project development. It determines the budget, time and resources needed to develop a software project. One of the well-established software project estimation models is Constructive Cost Model (COCOMO) which was developed in the 1980s. Even though such a model is being used, COCOMO has some weaknesses and software developers still facing the problem of lack of accuracy of the effort and cost estimation. Inaccuracy in the estimated effort will affect the schedule and cost of the whole project as well. The objective of this research is to use several algorithms of machine learning to estimate the effort of software project development. The best machine learning model is chosen to compare with the COCOMO.**

Keywords—Software effort estimation; project estimation; constructive cost model; COCOMO; machine learning

I. Introduction

Problems are created for software professionals, their clients, and stakeholders from the impractical project strategy and budget overruns. Despite many studies and numerous attempts to learn from experience, the problem of inaccurate often happen and has not been solved yet [1]. Software cost, effort, and resources are estimated at the beginning of the development. That information will be used by developers and clients to estimate the budget and time needed to finish develop an application or a system. Techniques and models were invented to assist the developers in estimating budget and effort. However, the problem of inaccuracy in estimation still becomes one of the problems for the developers and stakeholders. Even the emergence of one of a well-established project estimation model in the 1980s, COCOMO model, does not solve the problem of inaccuracy in software project estimation. Therefore, in this research, machine learning algorithms are used to estimate the effort of a software project that is more accurate compared to the COCOMO model. COCOMO model datasets are used to build machine learning models.

Although the COCOMO model has many advantages, it has some weaknesses too. One of it is its estimation varies as time progress [2]. Furthermore, the COCOMO model works depends on historical project data which are not available at all times [3]. COCOMO model cannot be used to estimate in all Software Development Life Cycle phases [3]. A large amount of data required for the COCOMO model to work [4]. A user has to insert input of 15 effort multipliers in order to get output from the COCOMO model. Thus, it will consume a lot of time for industry that has to estimate a large number of projects. COCOMO has difficulty in learn and identify data patterns [4] which is an important element in the regression model such as COCOMO.

Our main three objectives are to pre-process and analyze the COCOMO dataset. Second, is to apply several algorithms and to predict the output based on the COCOMO dataset and to evaluate the performances of the selected algorithms with the COCOMO method.

An application called SOFREST Estimator is developed to demonstrate how the estimation work and what are the inputs that needed to produce the outcome. The application will require user to insert five inputs about a project, which are number of Lines of Code (LOC), Database Size (DATA), Required Software Reliability (RELY), Execution Time Constraint (TIME) and Main Storage Constraint (STOR). The output of the application will be estimation of effort that needed for that particular project in person-months unit.

This project is significant because it concerns the accuracy in predicting the budget and time needed to develop a whole project. By doing the project cost and time estimation using machine learning, higher accuracy of cost and effort of a software project estimation will be produced since, in machine learning, we build a prediction model by train and test the dataset. By having machine learning as the project cost and effort estimator, money and time can be saved as it will need less human effort. This project will be useful for every software development company for them to estimate the cost and effort of a project. The best algorithm to be used in this project cost and time estimation can be determined based on the highest classification accuracy in machine learning.

II. Related Work

A. Software Project Estimation

Project estimation is an essential part of completing a project. Projects are planned in terms of cost, effort, and budget at the beginning phase of development. Precise effort estimation of software development plays a main task to predict how much workforce should be prepared during the works of a software project so that it can be completed on time and with the budget that planned without ignoring the quality of a software [5]. Accuracy of development cost estimation is a key factor in the success of a construction project and influenced the decision-making by the stakeholders of a software project [6] and to bid a contract with them [7].

The capacity of a budget estimating model is determined by calculating its bias, stability, and precision. Measures of bias, stability, and precision are concerned with the difference in the average between the actual costs and the estimated costs, considering both the degree of variation around the average and the combination with bias and consistency. By far, the most popular evaluation criteria used involve statistics such as mean, standard deviation, and coefficient of variation [6].

Identifying and calculating software metrics are important for various reasons, including estimating programming execution, measuring the effectiveness of software processes, estimating required efforts for processes, reduction of defects during software development, and monitoring and controlling software project executions [8]. An example of the wrong cost estimation that happened recently was in estimating the budget of international arrivals facility that being built at Seattle-Tacoma International Airport in Seattle, Washington, USA. Initially, in 2013 the budget was estimated at US$ 300 million but then the budget increased up to US$ 968 million in September 2018 [9]. Research shows that usually projects seem to be unclear at the beginning and become less vague as they progress [10].

One of the software metrics that used to estimate the cost and effort is called lines of code (LOC) metric and is considered basic software metric [11] as it is used in most software project estimation techniques.

B. *Project Estimation Techniques*

It is hard to quickly and accurately predict the development budget at the planning stage because the documentation is generally incomplete. For this reason, various procedures have been created to accurately predict construction costs with the limited project data available in the early phase [6]. There are three known models that have been used to estimate the project effort, cost and resources which are the Constructive Cost Model (COCOMO), Analogy-based Model, Use Case Points model.

1) COCOMO Model: COCOMO (Constructive Cost Model) is a screen-oriented, interactive software package that assists in budgetary planning and schedule estimation of a software project [12]. The intermediate COCOMO model used 15 drivers to estimate the cost of a project. The drivers are classified into four attributes; Product attributes, Hardware attributes, Personnel attributes and Project attributes [7].

Table I shows the intermediate driver of the COCOMO model. Each driver has its own multipliers (refers to Table II) that divided into six categories which are Very low (VL), Low (L), Neutral (N), High (H), Very High (VH), Extra High (XH) [7].

2) Analogy-based model: The core of the Analogy-based model is to differentiate the projects that will be estimated with all the software project's former data. Dataset will be from the company itself or that available publicly. The comparison will be carried out to identify which former projects are similar to the current project that its cost and effort will be estimated. Similar projects will be chosen to be reworked so that the estimated effort of the new project can be identified. Similarity measures, how near the distance between project, will be measured on each type of attribute and using three measurement methods; Euclidean, Manhattan and Minkowski distance [5].

3) Use-case points model: Use Case Points (UCP) is a notable size estimate designed mainly for object-oriented projects that use the use case diagram to estimate the size of projects at the beginning development phase. Other software sizing methods that depend on functional requirements, called Function Point, was what encouraged the idea of UCP [13].

TABLE I. INTERMEDIATE COCOMO DRIVERS

Category	***Drivers***
Product Attributes	Required Software Reliability (RELY)
	Database Size (DATA)
	Product Complexity (CPLX)
Hardware Attributes	Execution Time Constraint (TIME)
	Main Storage Constraint (STOR)
	Virtual Machine Volatility (VIRT)
	Computer Turnaround Time (TURN)
Personnel Attributes	Analyst Capability (ACAP)
	Application Experience (AEXP)
	Programmer Capability (PCAP)
	Virtual Machine Experience (VEXP)
	Programming Language Experience (LEXP)
Project Attributes	Modern Programming Practices (MODP)
	Use of Software Tools (TOOLS)
	Required Development Schedule (SCED)

TABLE II. INTERMEDIATE COCOMO MULTIPLIERS

	VL	***L***	***N***	***H***	***VH***	***XH***
RELY	0.75	0.88	1.00	1.15	1.40	-
DATA	0.94	1.00	1.08	1.16	-1.23	-
CMPL	0.70	0.85	1.00	1.15	1.30	1.65
TIME	1.00	1.11	1.30	1.66	-1.30	1.66
STOR	1.00	1.06	1.21	1.56	-1.21	1.56
VIRT	0.87	1.00	1.15	1.30	-1.49	-
TURN	0.87	1.00	1.07	1.15	-1.32	-
ACAP	1.46	1.19	1.00	0.86	0.71	-
AEXP	1.29	1.13	1.00	0.91	0.82	-
PCAP	1.42	1.17	1.00	0.86	0.70	-
VEXP	1.21	1.10	1.00	0.90	1.34	-
LEXP	1.14	1.07	1.00	0.95	1.20	-
MODP	1.24	1.10	1.00	0.91	0.82	-
TOOL	1.24	1.10	1.00	0.91	0.83	-
SCED	1.23	1.08	1.00	1.04	1.10	-

C. Machine Learning

A computer becomes much more intelligent with their ability that can think by using Artificial Intelligence (AI). One of the subfields of AI is Machine Learning (ML). The computer intelligence is developed through various methods of learning. Thus, there are many types of Machine Learning which are Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement, Evolutionary Learning and Deep Learning [14]. Machine Learning models are build based on learning the dataset using algorithms such as Regression Tree, Linear Regression, Neural Network, Naïve Bayes, Support Vector Machine, K-Nearest Neighbor, Random Forest, etc. The training and testing will be carried out to the dataset to build the ML models.

Previously, ML has offered self-driving cars, speech recognition, systematic web explores, and improved realization of the human generation. Today machine learning is available everywhere that one can possibly use it many times a day without knowing it. A lot of researchers consider it as an excellent way of moving towards human-level as machine learning are advanced to the extent that it can recognize speech like human [15].

Machine learning is a subfield of computer science. It allows machines such as computers to build analytical models of data and find hidden perceptions by learning the data itself. It has been applied to a variety of aspects in modern society, ranging from Deoxyribonucleic Acid sequences classification, credit card fraud detection, robot locomotion, to natural language processing. It can be used to solve many types of tasks such as classification and prediction. Software project estimation is one of the tasks that machine learning is capable of [16].

Machine Learning is important as it is always up to date with the current environment and the model keeps improving its performance itself by learning data or experience. Human efforts and mistakes can be reduced by using Machine Learning.

The traditional software effort performance criterion is not accurate and not satisfying. There were many metrics and a number of techniques in cost estimation have been proposed. Unfortunately, most of them have lacked one or both of two characteristics which are sound conceptual, theoretical bases and sstatistically significant experimental validation.

Most performance criterion metrics have been defined by an individual and then tested in a very limited environment [17]. So, there is a need design optimization algorithm for correct, precise and reliable effort estimation [18].

Data mining is about acquiring perception in the data in order to detect useful patterns that imply information. Data mining has a record of success in business and more recently in scientific applications. Data mining is usually carried out using process models and employs tens of techniques that span a wide spectrum of interdisciplinary fields including statistics, machine learning, and pattern recognition. The use of data mining in software project prediction has recently gained remarkable popularity inspired by a large amount of error in traditional estimation methods and the continuous improvements of machine learning algorithms which could help to provide more accurate prediction [7].

Machine Learning has been used to predict the software project cost and effort estimation since late 1980 [6]. Measuring the performance of estimation of machine learning models is accomplished by calculating the metrics including Sum Squared Errors (SSE), Root Mean Square Error (RMSE), Mean Magnitude of Relative Error (MMRE), Mean Absolute Error (MAE), etc.

They are the well-known parameters that are used for the performance evaluation of methods [19]. The evaluation consists of comparing the accuracy of the estimated effort with the actual effort. There are many evaluation criteria for software effort estimation and among them, the most frequent one is the Magnitude of Relative Error (MRE) [20]. Linear regression and Multi-perceptron are the most popular machine algorithms for software development effort estimation [21].

In this research, four Machine Learning algorithms that are used are Random Forest (RF), Linear Regression (LR), Regression Tree (RT) and Support Vector Machine (SVM). Each model's performance was measured by the Mean Magnitude Relatives Error (MMRE). Among the four algorithms, the best one is chosen to build a Machine Learning model that can predict the cost and time of a software project.

D. Literature Analysis

This section contributes to the knowledge of previous studies on software project estimation by using the state-of-art machine learning techniques available. Alongside with the limitation mentioned in the subsection C, most of previous literature studies addressed to measure the project and cost estimation using a single machine learning algorithm which reveals numerous limitations of the particular algorithm. Besides, most previous related work have inadequately present an extensive comparison and evaluation towards their proposed solution. Therefore, this paper aims to extend the evaluation with any other similar machine learning algorithm with four different datasets to further investigate the performance of the most sophisticated algorithm compared to COCOMO model.

III. Methodology

This research uses an experiment as a methodology to develop the prediction model for software project cost and effort estimation using selected machine learning algorithms. The experiment procedure is illustrated in Fig. 1. There are four selected machine learning algorithms which are Support Vector Machine, Linear Regression, Regression Tree and Random Forest.

A. Data Collection

In this experiment, software project measurement datasets use for developing the prediction model using machine learning. All datasets can be accessed publicly from http://promise.site.uottawa.ca/serepository/datasets-page.html and a study conducted by Kaushik et al, 2012. Sources of the datasets are from COCOMO NASA 1, COCOMO NASA 2, COCOMO81 and Kaushik et al. Details of the datasets used in this experiment are tabulated in Table III.

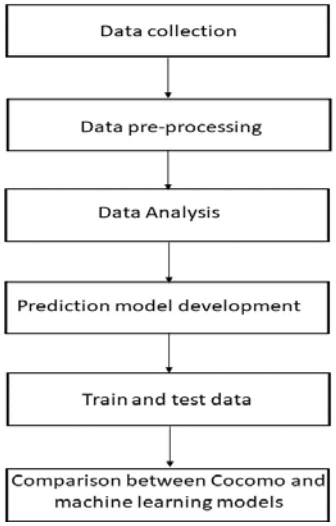
Data collection
Data pre-processing
Data Analysis
Prediction model development
Train and test data
Comparison between Cocomo and machine learning models

Fig. 1. Methodology.

TABLE III. DATASET USED IN THE EXPERIMENT

Dataset	*Attributes*	*Number of projects*
COCOMO NASA 1	17	60
COCOMO NASA 2	24	93
COCOMO81	15	63
Kaushik et al, 2012	16	15

B. Data Pre-processing

The data is pre-processed in order to calculate the effort estimation. In this experiment, the data is imported in r studio. Mice package is used to check the missing values, the datasets contain no missing values. The value of the drivers is in numerical weight converted to numerical values due to avoid bias during constructing the machine learning model. The mode constant is assigned based on the COCOMO predefined values.

C. Prediction Model Development

Four regression machine learning algorithms are used for this experiment.

1) Linear regression model: The linear regression model summarizes a relationship between two variables, independent and dependent variables. The practical use of linear regression in this experiment is to find the approximate prediction as a predictive model. The relationship of the prediction and the actuals data is then observed from the best fit line. The best fit line is where the total error prediction is as small as possible.

2) Support vector machine: Support vector machine model is a linear model for classification and regression problems. Linear and non-linear problems can be solved by support vector machine model. The aim of this model is to create a hyperplane and separate the data into classes. In support vector machine model, between the data points and the hyperplane we can find maximum margin to reduce misclassifications. Also, it can be used to solve unbalanced data problem.

3) Regression tree algorithm: egression tree is a type of decision tree and it is a method that can create and visualize prediction models from the data. The output of this model is numeric output, and the average value is assigned to the leaves of tree. The decision making in regression tree is easier compared to other method because the undesired data will be filtered and reduces the work on the data as it goes deeper in the tree. The regression tree is used due its ability to reduce ambiguity in decision-making.

4) Random forest algorithm: Random Forest model is model made up of many decision trees that each tree depends random vectors values. This model called random because during building trees it uses random sampling for training data points and during splitting nodes it uses random subsets of features considered. Each tree in random forest learns from a random sample of the data points. The random forest is used due to it can produce high accurate classifier.

5) Metrics: Mean Squared Error (MSE), the mean squared error or mean squared deviation of an estimator measures the average of the squares of the errors that is, the average squared difference between the estimated values and what is estimated. The lower the value of MSE, the better accuracy.

$$\text{MSE} = \frac{1}{\text{N}} \sum |\text{Effort}_{\text{estimated}} \text{- Effort}_{\text{actual}}| \quad (1)$$

Root Means Squared Error (RMSE). It represents the sample standard deviation of the differences between predicted values and observed values (called residuals).

$$\text{RMSE}=\frac{\sqrt{\sum (|\text{Effort}_{\text{estimated}} \text{- Effort}_{\text{actual}}|)^2}}{\text{N}} \quad (2)$$

Mean Absolute Error (MAE) is the average of the absolute difference between the predicted values and observed value. The MAE is a linear score which means that all the individual differences are weighted equally in the average.

$$MAE = \frac{1}{\text{N}} \sum |\frac{\text{Effort}_{\text{estimated}} \text{- Effort}_{\text{actual}}}{\text{Effort}_{\text{actual}}}| \quad (3)$$

Mean Absolute Percentage Error (MAPE) to measure prediction accuracy as percentage, or also known as the average absolute percent error for each predicted minus actual value and then divided by actuals values. The lower the lower of MAPE, the better the accuracy.

$$\text{MAPE}=\frac{100\%}{\text{N}} \sum |\frac{\text{Effort}_{\text{estimated}} \text{- Effort}_{\text{actual}}}{\text{Effort}_{\text{actual}}}| \quad (4)$$

The accuracy of the cost estimation models is evaluated by Magnitude of Relative Error (MRE) and the Mean Magnitude of Relative Error (MMRE). The optimum value of MRE and MMRE is closest to zero.

$$\text{MRE}=|\frac{\text{Effort}_{\text{estimated}} \text{- Effort}_{\text{actual}}}{\text{Effort}_{\text{actual}}}| \quad (5)$$

Mean Magnitude of Relative Error (MMRE) is the measure of predicted effort and actual effort value relative to the actual effort value.

$$MMRE=\frac{\Sigma=\left|\frac{Effort_{estimated}-Effort_{actual}}{Effort_{actual}}\right|}{N} \quad (6)$$

Min-Max Accuracy is a good metric to see how much they are close that considers the average between the minimum and the maximum prediction. The higher the value of Min-max accuracy the better the accuracy.

Correlation Accuracy is the correlation between predicted and actuals used as an accuracy measure. The Pearson product-moment correlation coefficient is used to measure the strength of predicted and actuals value of the experiment. The predicted and actuals value has similar directional movements when the correlation accuracy is high.

P-Value also known as calculated probability is to determine the significance of the experiments results. The P-Value is lower than 0.05 shows strong prove against the null hypothesis, thus the null hypothesis is rejected. The smaller the P-Value, the stronger the evidence to reject the null hypothesis.

Null hypothesis of this project is, the population correlation coefficient is not significantly different from zero. There is no significant linear correlation between control and experimental values in the population.

Alternative hypothesis of this project is, the population correlation coefficient is significantly different from zero. There is a significant linear relationship between control and experimental values in the population [22].

Vargha and Delaney A (VDA) measure is one of the examples to measure effect size, differentiate between two samples of observations, control and experimental sample. The range of VDA is from 0 to 1. VDA is calculated there is no effect result in a value of 0.50 [23]. Interpretation of A measure:

- A value around 0.56 = small effect;
- A value around 0.64 = medium effect;
- A value around 0.71 = big effect;

Wilcoxon Rank Sum Test is nonparametric test is used to compare two related samples on a single sample to see if their population ranks differ. The null hypothesis is difference between the two samples has equal medians. The alternative hypothesis is there is no difference between the two samples. If the p-value is larger than 0.05, we must accept the null hypothesis because there is enough evidence to conclude. The null hypothesis is rejected, there is sufficient evidence to conclude the sample has no identical distributions [24].

D. Data Training and Testing

Training dataset are 80% and testing dataset are 20% for COCOMO81, COCOMO NASA 1. Training dataset are 70% and testing dataset are 30% for COCOMO NASA 2, and Kaushik et al.

IV. Data Analysis

In this project, correlation matrix uses to evaluate the correlation of the two variables. The dependent variable is actual effort attribute, while the 15 cost drivers and the line code are independent variable. From Fig. 2, 3, and 4, there are five attributes that high positive correlation towards actual effort attribute, LOC, DATA, TIME, TOOL and STOR for COCOMO81, COCOMO NASA 1, COCOMO NASA 2. While, other attributes show negative correlations towards actual effort attribute.

Then, the project builds predictive machine learning models with COCOMO81, COCOMO NASA 1, COCOMO NASA 2 and Kaushik et al datasets, using all attributes. The predictive machine learning models are Support Vector Machine, Linear Regression, Regression Tree and Random forest. The result record is in the Table IV. The project evaluates the result and records in a table.

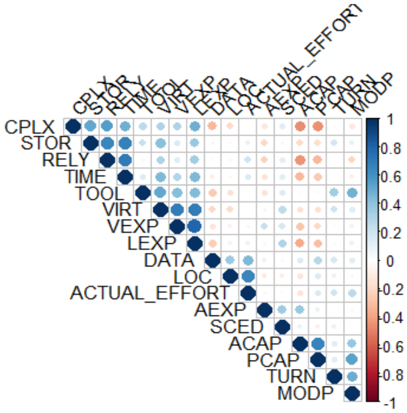

Fig. 2. Correlation of COCOMO81 Attributes.

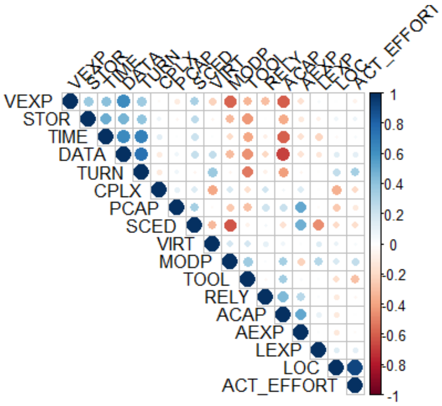

Fig. 3. Correlation of COCOMO NASA 1 Attributes.

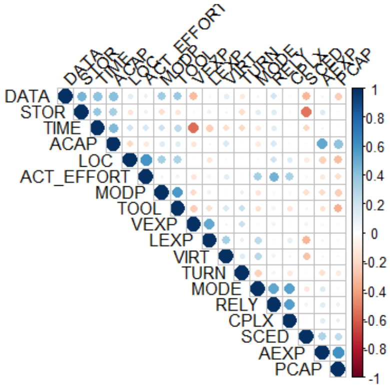

Fig. 4. Correlation of COCOMO NASA 2 Attributes.

TABLE IV. COMPARISON OF MACHINE LEARNING MODELS WITH COCOMO81 DATASET

Algorithm	*Support Vector Machine*	*Linear Regression*	*Regression Tree*	*Random Forest*
MAE	1002.844	1006.062	943.3075	928.3318
MSE	7675315	6939266	5193249	5769025
MAPE	4.721069	5.151246	4.803107	7.575327
RMSE	2770.436	2634.249	2278.87	2401.880
Min Max Accuracy	0.2742493	0.2647441	0.2758899	0.2895891
Correlation Accuracy	0.8218062	0.5933549	0.7427998	0.8671952
P-value	0.00568	0.03254	0.003628	0.001236
Significant Value	Significant	Not significant	Significant	Significant

Table IV explains the experiment is made on COCOMO81 with training dataset of 80 percent and the testing dataset is 20 percent. Based on the Table IV, Support Vector Machine, Regression Tree and Random Forest are significant due to the p-value which is less than 0.005 except Linear Regression model which p-value is 0.03254.

Table V explains the experiment that made on COCOMO NASA 1 with training dataset of 80 percent and the testing dataset is 20 percent. Based on the Table V, Support Vector Machine, Linear Regression, Random Forest are significant due to the p-value which is less than 0.005. The best MAE value among the four models is Support Vector Machine model which is the lowest, 31.5403163. The second-best MAE goes to Random Forest model with 36.8215429, followed up by Regression Tree with MAE 44.7018519. The worst MAE which has the highest value goes to Linear Regression with 47.7723733.

For the experiment that is made on COCOMO NASA 2 with training dataset of 80 percent and the testing dataset is 20 percent. Based on the Table VI, Support Vector Machine, Linear Regression, Random Forest and Regression Tree models are significant due to the p-value which is less than 0.005.

The experiment is made on Kaushik et al with training dataset of 70 percent and the testing dataset is 30 percent. Based on the Table VII, Support Vector Machine, Linear Regression, Random Forest are significant due to sufficient evidence to reject null hypothesis as the p-value which is less than 0.005; except Regression Tree model which p-value is 0.18 higher than 0.005, fail to reject null hypothesis.

From the evaluation above, Support Vector Machine and Random Forest show consistent and statistically significant between the two variables, predicted effort value and actual effort values; with the four datasets. On the other hand, Linear Regression and Regression Tree show inconsistent result and fail to reject null hypothesis. However, we cannot strongly prove that the statistically significant result of the research hypothesis is correct (100% certainty) and we need to calculate the effect size of the control vs experimental values.

TABLE V. COMPARISON OF MACHINE LEARNING MODELS WITH COCOMO NASA 1 DATASET

Algorithm	*Support Vector Machine*	*Linear Regression*	*Regression Tree*	*Random Forest*
MAE	31.5403	47.77237	44.701852	36.82154
MSE	2755.54	5078.1734	2719.832	2421.112
MAPE	0.290164	0.4632032	0.5614929	0.4032879
RMSE	52.49332	71.2613	52.15201	49.20480
Min Max Accuracy	0.8019366	0.5631169	0.645906	0.7328254
Correlation Accuracy	0.956159	0.95610	0.933626	0.9400819
P-value	3.67e-05	1.193e-06	9.069e-06	5.498e-06
Significant Value	Significant	Significant	Significant	Significant

TABLE VI. COMPARISON OF MACHINE LEARNING MODELS WITH COCOMO NASA 2 DATASET

Algorithm	*Support Vector Machine*	*Linear Regression*	*Regression Tree*	*Random Forest*
MAE	262.6466	412.9564	305.56240	281.0156
MSE	158576.63	318435.7	360594.68	187571.8
MAPE	2.101547	3.674328	1.61244	1.363585
RMSE	398.2165	564.3010	600.49536	433.0956
Min Max Accuracy	0.452203	0.332110	0.530223	0.5227043
Correlation Accuracy	0.712561	0.697992	0.568656	0.727761
P-value	0.0006161	0.008902	0.01108	0.004126
Significant Value	Significant	Significant	Significant	Significant

TABLE VII. COMPARISON OF MACHINE LEARNING MODELS WITH KAUSHIK ET.AL, 2012

Algorithm	*Support Vector Machine*	*Linear Regression*	*Regression Tree*	*Random Forest*
MAE	22.28604	14.77559	54.408	36.4
MSE	683.8073	468.77943	5247.853	2366.2582
MAPE	0.1721194	0.09580776	0.5615453	0.2335138
RMSE	26.14971	26.65131	72.4420	48.64420
Min Max Accuracy	0.82788	0.90449	0.64094	0.766486
Correlation Accuracy	0.9902	0.98828	0.7098	0.99224
P-value	3.67e-05	0.00152	0.18	0.008192
Significant Value	Significant	Significant	Not significant	Significant

In this research further experiment is to analysis the regression models with selected attributes COCOMO dataset using A measure and Wilcoxon Rank Sum test.

From Table VIII, the project calculates the accuracy metrics of COCOMO NASA 1, train and test Support Vector Machine and Random Forest with all attributes of COCOMO NASA 1, also with 5 selected attributes and one attribute only, LOC. From the result, COCOMO NASA 1, has the highest correlation accuracy compared to other experiments with machine models. There is sufficient evidence to reject the null hypothesis as the p-value of COCOMO NASA 1 is smaller than 5 percent, thus COCOMO NASA 1 is statistically significant. The MMRE of COCOMO NASA 1 is smaller compared to machine learning models' MMRE. The smaller the MMRE indicates the more accurate of the estimation [25].

As for Vargha and Delaney A measure there is no effect of difference between actual effort and the predict value of COCOMO NASA 1 model, to support the statement rank sum p-value is used to measure the distribution between the control and experimental sample. It shows that the p-value of Wilcoxon rank sum test is higher than 5 percent, the null hypothesis is fail to reject, the two samples has identical distributions.

To compare the trained machine learning models with all attributes and machines learning models only with 5 selected attributes, the five selected attributes performance better in producing more accurate results. The correlation accuracy of the five selected attributes have higher relationship between the actual effort and the predicted effort values compared to the all attributes. MMRE of five selected attributes has lower values compared to all attributes, this show that five selected attributes has more accurate estimations between the actual and predicted effort values. For Vargha and Delaney A measure, there are only slight differences between the all attributes and the five selected attributes, Support Vector Machine with all attributes has no effect differences compared to Random Forest model. The rank sum of all attributes and five selected attributes are statistically significant, there are enough evidence to support null hypothesis and to reject the alternative hypothesis.

Then experiment is using only one attribute, LOC. The correlation accuracy of machine models are increased compared to five attributes and all attributes. The p-value of Pearson's correlation also show the models are statistically significant. The MMRE of Random Forest is lower than Support Vector Machine, 67.6706. The Vargha and Delaney A measure of Support Vector Machine and Random Forest has no effect in difference, this mean the distribution of actual and predicted effort values are identical. The Wilcoxon rank sum test of Support Vector Machine and Random Forest are statistically significant where, there are enough evidences to support null hypothesis and reject the alternative hypothesis since the p-value of both machine learning models is higher than 5 percent.

To conclude, the machine learning models learn and prove that not all attributes are needed to trained and needed. From this experiment, using one attribute, LOC can have closer MMRE towards the COCOMO prediction model, higher correlation accuracy and identical distribution of actual and predicted effort values.

From the Table IX, the project calculates the accuracy metrics of COCOMO NASA 2, train and test Support Vector Machine and Random Forest with all attributes of COCOMO NASA 2, also with 5 selected attributes and one attribute only, LOC. COCOMO NASA 2 has 93 projects and highest number projects compared to other COCOMO datasets. From the result, COCOMO NASA 2 has the lower correlation accuracy compared to experiments with machine learning models that used all attributes. There is no sufficient evidence to reject the null hypothesis as the p-value of COCOMO NASA 1 is larger than 5 percent, thus COCOMO NASA 2 is not statistically significant. The MMRE of COCOMO NASA 2 is smaller compared to machine learning models' MMRE.

The smaller the MMRE indicates the more accurate of the estimation (Malhotra, 2014). As for Vargha and Delaney A measure there is intermediate differences between actual effort and the predict value of COCOMO NASA 2 model, however the Wilcoxon rank sum p-value is used to measure the distribution between the control and experimental sample. It shows that the p-value of Wilcoxon rank sum test is higher than 5 percent, the null hypothesis is fail to reject, the two samples has identical distributions. The COCOMO NASA 2 is still statistically significant according to rank sum test.

TABLE VIII. COMPARISON OF COCOMO NASA 1 DATASET WITH DIFFERENT ATTRIBUTES

		All Attributes		***5 Attributes***		***LOC Only***	
Algorithm	***COCOMO NASA 1***	***Support Vector Machine***	***Random Forest***	***Support Vector Machine***	***Random Forest***	***Support Vector Machine***	***Random Forest***
Correlation Accuracy	0.97578	0.8157	0.6623	0.7636	0.8642	0.7772	0.85521
P-value	6.295e-08	0.00122	0.01895	0.003842	0.002882	0.002933	0.000391
Significant Value	Significant	Significant	Significant	Significant	Significant	Significant	Significant
MMRE	29.61	127.5024	110.7032	109.684	108.4127	66.98639	67.6706
A Measure	0.5 (No effect)	0.4514 (No effect)	0.4097 (Small)	0.42361 (Small)	0.42361 (Small)	0.45833 (No effect)	0.4861 (No effect)
Rank Sum	1	0.7074	0.4704	0.5443	0.5443	0.7508	0.931
Significant Value of Rank Sum	Significant	Significant	Significant	Significant	Significant	Significant	Significant

TABLE IX. COMPARISON OF COCOMO NASA 2 WITH DIFFERENT ATTRIBUTES

		All Attributes		***5 Attributes***		***LOC Only***	
Algorithm	***COCOMO NASA 1***	***Support Vector Machine***	***Random Forest***	***Support Vector Machine***	***Random Forest***	***Support Vector Machine***	***Random Forest***
Correlation Accuracy	0.4388394	0.8142797	0.7077862	0.6552592	0.7265169	0.4203878	0.4567935
P-value	0.06016	2.202e-05	0.0006982	0.002324	0.004269	0.07311	0.04929
Significant Value	Not Significant	Significant	Significant	Significant	Significant	Not Significant	Significant
MMRE	150	281.5356	201.6638	226.7969	334.0268	205.99	133.2845
A Measure	0.1952663 (Large)	0.318559 (Medium)	0.3490305 (Small)	0.3268698 (Medium)	0.2908587 (Medium)	0.3434903 (Small)	0.3822715 (Small)
Rank Sum	0.1611	0.05771	0.1149	0.07025	0.02854	0.102	0.22
Significant Value of Rank Sum	Significant	Significant	Significant	Significant	Not Significant	Significant	Significant

In the experiment of machine learning models are using all attributes, there are large differences result obtained from Support Vector Machine models and Random Forest model. The correlation accuracy of the Random Forest is lower than Support Vector Machine however Random Forest has higher MMRE value compared to Support Vector Machine. Moreover, Vargha and Delaney A measure, Support Vector Machine and large difference between the predicted and actual effort values, while Random Forest has closer accuracy of control and experimental sample. The Wilcoxon rank sum test for Support Vector Machine and Random Forest model are statistically significant where there are enough evidence to support null hypothesis and reject the alternative hypothesis, as the p-value of both machine learning is higher than 5 percent.

Then, the experiments are evaluated between all attributes and five selected attributes. The correlation accuracy of the five selected attributes have higher relationship between the actual effort and the predicted effort values compared to the all attributes. However the MMRE of Random Forest for the five selected attributes has higher value compared to Random Forest for the all attributes, while for Support Vector Machine is vice versa.

For Vargha and Delaney A measure, the selected five attributes for both machine learning show medium differences between the actual and predicted effort value. The rank sum of all attributes and five selected attributes are statistically significant, there are enough evidence to support null hypothesis and to reject the alternative hypothesis.

Then experiment is using only one attribute, LOC. The correlation accuracy of machine models drop compared to five attributes and all attributes. The p-value of Pearson's correlation only show the Random Forest model is statistically significant. The MMRE of Random Forest is lower than Support Vector Machine compared to all attributes and COCOMO NASA 2 model. The Vargha and Delaney A measure of Support Vector Machine and Random Forest have small in difference, this mean the distribution of actual and predicted effort values are closely identical.

The Wilcoxon rank sum test of Support Vector Machine and Random Forest are statistically significant where, there are enough evidences to support null hypothesis and reject the alternative hypothesis since the p-value of both machine learning models is higher than 5 percent.

To conclude, the machine learning models learn and prove that not all attributes are needed to trained and needed. From this experiment, using one attribute, LOC can have closer MMRE towards the COCOMO prediction model, higher correlation accuracy and identical distribution of actual and predicted effort values.

From the Fig. 5, the machine learning model support vector machine and random forest show similar pattern towards to the actual effort, while the calculation of COCOMO deviates at 100, 14, 302,113, 350 and 339 lines of codes. Support Vector Machine and Random Forest has proximity predicted effort values compared to effort prediction of COCOMO models. Refer to appendices for development of machine learning model.

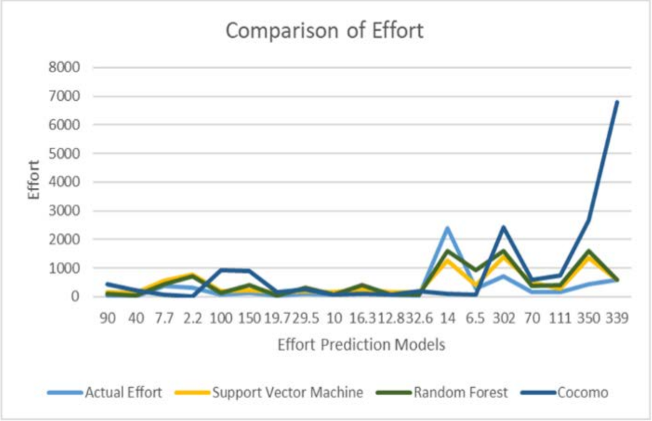

Fig. 5. Comparison of Actual Effort and Estimated Effort.

V. DISCUSSION

The results of the experiments found clear support that Support Vector Machine and Random Forest algorithms impressively give consistent results with the COCOMO datasets regardless on the number of effort attribute used.

However, the planned comparison in this paper reveals the good performance and increase in the accuracy of Support Vector Machine for estimation of software project effort. Support Vector Machine also able to delivers significantly better results with five important attributes compared to all attributes used to estimate project effort and cost, as some of the attributes are irrelevant to estimate.

VI. Conclusion and Future Work

To conclude, many existing machine learning algorithms can train predictive models, however, the right and suitable machine learning model are needed to give an accurate estimation. In this research, the five selected attributes with high positive correlation toward actual effort attribute are obtained from the correlation matrix, DATA, STOR, LOC, TIME and TOOL. The five important attributes give better result compared from using all the attributes in COCOMO dataset. Hence, not all attributes in the dataset are relevant to be used to measure the project estimation. Further improvement, during training and testing data, the data is advised to split three parts, 70 percent of training data, 20 percent of testing data, and 10 percent of validation data.

Further improvement of this research on machine learning models is to perform ensemble stacking also known as blending, to ensemble the four machine learning models in order to optimize the predictive model. In development of machine learning model, percentage of accuracy should be included to show the difference percentage between the predicted value and the actual effort value.

Acknowledgment

This research is funded by IIUM Research Acculturation Grant Scheme with the Ref: IRAGS18-012-0013.

References

[1] M. Jorgensen, “What We Do and Dont Know about Software Development Effort Estimation,” IEEE Softw., vol. 31, no. 2, pp. 37–40, 2014.

[2] J. Kaur, T., & Singh, “A Review on Cost Estimation Models for Effort Estimation,” Int. J. Sci. Eng. Res., vol. 6, no. 5, pp. 179–183, 2015.

[3] M. Aljohani and R. Qureshi, “Comparative Study of Software Estimation Techniques,” Int. J. Softw. Eng. Appl., vol. 8, no. 6, pp. 39–53, 2017.

[4] U. Shekhar, S., & Kumar, “Review of Various Software Cost Estimation Techniques.,” Int. J. Comput. Appl., vol. 141, no. 11, pp. 31–34, 2016.

[5] S. Ardiansyah, A., Mardhia, M. M., & Handayaningsih, “Analogy-based model for software project effort estimation,” Int. J. Adv. Intell. Informatics, vol. 4, no. 3, pp. 251–260, 2018.

[6] G.-H. Cho, H.-G., Kim, K.-G., Kim, J.-Y., & Kim, “A Comparison of Construction Cost Estimation Using Multiple Regression Analysis and Neural Network in Elementary School Project.,” J. Korea Inst. Build. Constr., vol. 13, no. 1, pp. 66–74, 2013.

[7] A. Banimustafa, “Predicting Software Effort Estimation Using Machine Learning Techniques,” 8th Int. Conf. Comput. Sci. Inf. Technol. CSIT 2018, (October), pp. 249–256, 2018.

[8] R. Kalaivani, N., & Beena, “Overview of Software Defect Prediction Using Machine Learning Algorithms,” Int. J. Pure Appl. Math., vol. 118, pp. 3863–3873, 2018.

[9] J. Thomas, “Blown Budgets and Destroyed Schedules. Sometimes, It’s Weak Project Estimation That’s to Blame,” Sci. Uncertain., pp. 56–61, 2019.

[10] N. Ghatasheh, H. Faris, I. Aljarah, and R. M. H. Al-Sayyed, “Optimizing Software Effort Estimation Models Using Firefly Algorithm,” J. Softw. Eng. Appl., vol. 8, no. 3, pp. 133–142, 2015.

[11] M. Z. Alsaeedi, A., & Khan, “Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study,” J. Softw. Eng. Appl., vol. 12, no. 5, pp. 85–100, 2019.

[12] B. W. Boehm et al., Software Cost Estimation with COCOMO II. Upper Saddle River, NJ: Prentice Hall, 2000.

[13] A. B. Azzeh, M., & Nassif, “A hybrid model for estimating software project effort from Use Case Points,” Appl. Soft Comput., vol. 49, pp. 981–989, 2016.

[14] M. Fatima, M., & Pasha, “Survey of Machine Learning Algorithms for Disease Diagnostic,” J. Intell. Learn. Syst. Appl., vol. 9, no. 1, pp. 1–16, 2017.

[15] N. Rambhajani, M., Deepanker, W., & Pathak, “A Survey on Implementation of Machine Learning Techniques for Dermatology Diseases Classification.,” Int. J. Adv. Eng. Technol., vol. 8, pp. 194–195, 2015.

[16] N. Wang, “Bankruptcy Prediction Using Machine Learning,” J. Math. Financ., vol. 7, no. 4, p. 908–918.

[17] P. K. Tripathi, R., & Rai, “Machine Learning Methods of Effort Estimation and It’s Performance Evaluation Criteria.,” Int. J. Comput. Sci. Mob. Comput., vol. 6, no. 1, pp. 61–67, 2017.

[18] C. Wadhwa, A., Jain, S., & Gupta, “An Effective Precision Enhancement Approach to Estimate Software Development Cost: Nature Inspired Way.,” J. Telecommun. Electron. Comput. Eng., vol. 9, no. 3, 2017.

[19] S. Malathi, S., & Sridhar, “Analysis of size matrics and effort performance criterion in software cost estimation.,” Indian J. Comput. Sci. Eng., vol. 3, no. 1, 2012.

[20] S. Kaushik, A., Chauhan, A., Mittal, D., & Gupta, “COCOMO Estimates Using Neural Networks.,” Int. J. Intell. Syst. Appl., vol. 4, no. 9, pp. 22–28, 2012.

[21] V. K. Bhatia, S., & Attri, “Machine Learning Techniques in Software Effort Estimation Using COCOMO Dataset,” vol. 2, no. 6, pp. 101–106, 2015.

[22] B. Illowsky, & S. Dean, Testing the Significance of The Correlation Coefficient. Introductory Statistics, Rice University, Houston, USA, 2013.

[23] A. R. Ismail, “Immune-Inspired Self-Healing Swarm Robotic Systems,” York., 2011.

[24] “Texas Gateway,” 2019. [Online] https://www.texasgateway.org/resource/95-additional-information-and-full-hypothesis-test-examples

[25] R. Malhotra, “A systematic review of machine learning techniques for software fault prediction.,” 2014. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1568494614005857.