International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 6, Issue 10, pp 56-62, October-2022
https://doi.org/10.47001/IRJIET/2022.610009

Software Maintenance Potential Prediction Based on Machine Learning

[1]Mazin Mohammed Ismael, [2]Ibrahim Ahmed Saleh

[1]M.Sc. Student, Dept. of Software, College of Computer & Math., University of Mosul, Iraq
[2]Professor, Dept. of Software, College of Computer & Math., University of Mosul, Iraq

***Abstract* - Software maintenance (SM) is one of important environment in software engineering field more complex. Therefore maintains critical to extend an environment for sharing and sustaining knowledge. In this paper introduce presents a survey reviews published materials of predication software engineering maintenance Based on several algorithms form field of artificial intelligence. This survey is useful as researchers that work for objective of software maintenance Based on algorithms of Artificial intelligence and will understand and gain the current landscape of the research and the insights gathered.**

Keywords: Machine learning; software maintainability; software predicate; object-oriented metrics; fuzzy-logic; Artificial intelligence.

I. INTRODUCTION

Software maintainability means ease a software system or component can be modified to correct faults, improve performance, other attributes and adapt to a changing environment. Software change is required to meet customer modifications requirements, which may arise due to many reasons such as changes in technology, introduction of new hardware or enhancement of the features provided, etc. [1]. It is both impractical and extremely unprofitable to create software that will never need to be updated. This process of changing software has been delivered is called software maintenance. It is the most costly software development phase, which sometimes even consumes almost 60-70% of the total development cost depending on the complexity of the system under consideration [2].

To differentiate between maintenance and maintainability we consider the following definitions:

Software maintenance is defined as “the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changing environment”, while software maintainability is defined as “the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changing environment” [3].

From the definitions, it is clear that maintenance is a process performed as part of the Software Development Life Cycle (SDLC) whereas maintainability is a quality attribute associated with the software product. These are two inherently different but interlocked concepts. Maintenance vs. maintainability represent process vs. quality attribute and their predictions are called process cost prediction vs. quality attribute measurement, respectively. A software maintainability prediction model enables organizations to predict the maintainability of their software systems, thus providing means to be better manage their maintenance resources in addition to adopting a defensive design. This can reduce the maintenance effort and therefore, reduce the overall cost and time spent on a software project [4].

Given how an improved understanding of software maintainability can help organizations create better systems, it is important to understand this process to make significant contributions in this area. Therefore, this paper presents a Systematic Review (SR) of many researches on Software Maintainability Prediction (SMP) based on machine learning will be mentioned it turns out that one of main approaches to controlling maintenance costs is to monitor software metrics during the development phase. Software maintainability depends on two types of attributes, including internal and external attributes. Internal attributes such as size, cohesion, and coupling can easily be derived directly from the software. On the other hand, external attributes such as modifiability, understandability, maintainability, etc., are challenging to capture. External attributes often depend on internal attributes and can be derived indirectly through them [5].

DeMarco has righteously quoted that something that cannot be measured, cannot be even controlled also [6]. To measure the maintainability we first find out “the change effort”. It is defined as “how much amount of average effort is required to add, change or delete existing classes”. And to find change effort to measure maintainability, it is necessary to measure various features of object-oriented paradigms such as inheritance, cohesion, coupling, memory allocation, etc. different metrics are carefully selected.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 10, pp 56-62, October-2022

https://doi.org/10.47001/IRJIET/2022.610009

Because the relationships between static software metrics and their maintainability are very complex and nonlinear, hence conventional statistical techniques-based models, which are purely based on quantity, would not help much to the problem. Instead, using machine learning algorithms to establish the relationship between metrics and maintainability would be a much better approach as these are based on quantity and quality.

II. LITERATURE REVIEW

In literature, various Machine Learning (ML) metrics and models have been proposed for Software Maintainability Prediction (SMP). These methods vary from simple statistical models such as regression analysis to complex machine learning algorithms such as neural networks etc. the elaborate few important studies here.

In 2006 Chikako V. K. and Gray A. R.[7] presented in their paper a maintainability prediction model for an object-oriented software system using a Bayesian network. They used two commercial software datasets User Interface Management System (UIMS) and Quality Evaluation System (QUES) developed in ADA. The model prediction accuracy was assessed and contrasted with other popular regression-based models. According to results, researchers reached that Bayesian network model can predict maintainability more accurately than regression-based models for one system, and almost as accurately as the best regression-based model.

In 2007 Yuming Z. and Leung H.[8] used Multiple Adaptive Regression Splines (MARS), a novel exploratory modeling technique, to create a Software Maintainability Prediction (SMP) model from metrics gathered from two different object-oriented systems (UIMS and QUES datasets). Multivariate Linear Regression Model, Artificial Neural Network Model, Regression Tree Model and Support Vector Model were used to assess and compared the MARS model predictive accuracy .overall, it was concluded that MARS model was better than the other models built using typical modeling techniques and it could be a useful modeling technique for software maintainability prediction.

In 2009 Li-jin Wang et.al [9] introduce model to predict the maintainability of object-oriented software using a Projection Pursuit Regression (PPR). To evaluate the benefits of using PPR over nonlinear modeling techniques, they also built Artificial Neural Network Model and Multivariate Adaptive Regression Splines Model. They used UIMS and QUES datasets to build models. This study, maintainability was calculated as the number of modifications made the code throughout a maintenance period. they used all metrics except NOC from the QUES dataset to build a model by Artificial Neural Networks (ANN), MARS, and PPR respectively. The outcomes implied that PPR was more predictively accurate than the other two modeling approaches. The study also offered helpful guidance for building software quality models. Therefore Also In 2009, Mahmoud O. et. al [10] proposed TreeNet for predicting object-oriented software maintainability. It had used also two datasets (UIMS and QUES datasets) and compared its prediction performance against Multivariate Adaptive Regression Splines, Multivariate Linear Regression, Support Vector Regression, Artificial Neural Network and Regression Tree Models. The results of the prediction accuracy measures achieved by the TreeNet model for the UIMS dataset achieved the best Mean Magnitude of Relative Error (MMRE) value. The results indicate that improved or at least competitive, prediction accuracy had been achieved when applying the TreeNet model.

In 2010 Kaurand A. et.al [11], they evaluated and compared the application of different artificial technical to constructed models for the prediction of Software Maintenance Effort, they used Artificial Neural Networks (ANN), Fuzzy Inference Systems (FIS), and Adaptive Neuro-Fuzzy Inference Systems (ANFIS),also proposed hybrid approach with combine "ANN &FIS". They used UIMS and QUES datasets. The maintenance was measured using a number of lines for each category, and it could be deleting or adding a line. They formulated the architecture as one hidden layer in which five neurons were set up using a hyperbolic tangent as an activation function in the hidden layer. Later they used Mean Absolute Relative Error (MARE), and Mean Relative Error (MRE) to estimate the models variability towards overestimation and underestimation then P-Value to find out the no correlation hypothesis. The results showed the MARE of the Feed forward Backdrop Artificial Neural Network model was best from whereas Generalized Regression Neural Network (GRNN). Hence it was concluded soft computing techniques could be successfully used for the prediction of software maintenance efforts. In the same year, Jin and Liu [12] used Support Vector Machine (SVM) and clustering techniques to predict the software maintenance effort. And they used object-oriented metrics, maintenance effort was chosen as the dependent variable, and five metrics were chosen as the independent variable. That shows were a high degree of confidence for successful validations and the correlation between predicted maintenance effort and actual maintenance was 0.769. Therefore, they found that SVM and clustering techniques were useful in constructing software maintainability predictors.

In 2012 Malhotra R. and Chug A.[13] proposed three ML algorithms including Group Method of Data Handling (GMDH), Probabilistic Neural Network (PNN) and Genetic Algorithms (GAs) for SMP. This study was based on two

ISSN (online): 2581-3048
Volume 6, Issue 10, pp 56-62, October-2022
https://doi.org/10.47001/IRJIET/2022.610009

commercial software datasets UIMS and QUES developed in ADA. The prediction performance of the machine learning algorithms-based models like GMDH, GA, and PNN were assessed and compared with prevailing models in terms of MMRE. It was found that the GMDH model outperformed the prevailing models as the least MMRE value was recorded and it was 0.210. As far as Pred(0.25) and Pred(0.30) values were concerned, all three proposed models were significantly better than the others. Thus, it was concluded the GMDH network model was indeed a very useful modeling technique and it could be used as a sound alternative for the prediction of software maintainability.

In 2013 Ahmed M. A. and Al-JamimiH. A. [2] introduce a paper for predicting maintainability using Fuzzy-Logic-Based models. They applied a model to predict software maintainability through some software measures; they used two common and publicly available datasets UIMS and QUES datasets. They applied process to a case study where Mamdani Fuzzy Inference Engine was used to predict software maintainability and they compared Mamdani-based models with T-S-based, SVM, PNN, RBF, BN, and MARS models. Out of all the models. The NRMSE values for the model accuracy after applying the Mamdani -based model to QUES dataset were 0.11 for the training process and 0.16 for the testing process, while they were 0.13 for the training process and 0.21 for the testing process after applying the Mamdani-based model to the UIMS dataset. The results showed the Mamdani-based model was superior to all.

In 2015 Yadav V. K. et.al [14] concentrated on building a method based on data mining techniques K-means and hierarchical clustering, which were implemented in the MATLAB package on two commercial systems, UIMS (User Interface Management System) and QUES (Quality Evaluation System). It was established the algorithm would be capable of selecting clusters with easy, medium and high requirements of maintainable classes of similarity based on object-oriented metrics. As a result, this method was the most beneficial for the software maker and maintainers to take necessary steps at the design level to design maintainable object-oriented software.

In 2016 Kumar L. and Rath S. K. [15] building object-oriented software maintainability prediction models based on three Artificial Intelligence techniques such as a hybrid approach of Functional Link Artificial Neural Network (FLANN) with Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Clonal Selection Algorithm (CSA), i.e., FLANN-Genetic (FGA and AFGA), FLANN-PSO (FPSO and MFPSO), FLANN-CSA (FCSA) were applied to design a model for predicting maintainability. The software metrics were used as requisite input data to train models to estimate the number of changes made in code during the maintenance phase. The paper also focuses on the effectiveness of feature reduction techniques such as Rough Set Analysis (RSA) and Principal Component Analysis (PCA) when they were applied to predicting maintainability. The results showed feature reduction techniques were very effective in obtaining better results while using FLANN-Genetic. While in same year, Jain A. and other researchers [16] introduced Evolutionary techniques, specifically Genetic Algorithms for software maintainability prediction and compared their performance with other machine learning techniques like Decision Tables, Radial Basis Function Neural Networks, Bayes Nets and Sequential Minimal Optimization (SMO). The paper has two versions from four open source software and calculated change for each one of them. The independent metrics were calculated using ckjm tool. The prediction models were built by using the machine learning classifiers provided by the weka tool. Then the output of the proposed classifiers was then assembled and examined in terms of Mean Absolute Error (MAE), which had a value of 1, and Root Mean Square Error (RMSE), which had a value of 1, respectively. It was clear from the findings that the genetic algorithm outperforms the other machine learning classifiers. According to their findings, the evolutionary algorithm performed better than all the other classifiers.

In 2017 Kumar L. and Rath S. K. [17] used a hybrid neural network with Fuzzy Logic approach to develop a maintainability prediction model for two commercial object-oriented software. It had applied this method using two different datasets (UIMS and QUES datasets). A suitable collection of measures were chosen from ten metrics utilized to enhance performance of maintainability prediction model using Rough Set Analysis (RSA) and Principal Component Analysis (PCA). The performance of the Neuro-Fuzzy model was assessed and compared with those of the work carried out by several researchers and observed after using PCA and RSA the training time got reduced. The algorithms gave better performance results for (MMRE) reached to 0.2826.Further, it was observed that a selected subset of metrics using feature selection techniques i.e., PCA, and RSA was able to predict maintainability with a higher accuracy.

In 2018 Baskar N. and Chandrasekar C.[18] proposed optimized Neuro-PSO model for Software Maintainability Prediction (SMP). The paper for maintenance efforts data was obtained from object-oriented software datasets namely UIMS and QUES for computing the maintenance efforts. This method applied dimensionality reduction using relief feature selection method for identifying the optimal feature subsets in order to increase the accuracy and reduce time complexity of prediction model. The paper achieved an improved result in

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 10, pp 56-62, October-2022

https://doi.org/10.47001/IRJIET/2022.610009

developing a well-equipped software maintenance prediction model for object-oriented software.

In 2019 Wang X. et.al [19] proposed a study on SMP. The study proposed a new framework for software maintainability prediction, which was based on Fuzzy network, a new exploratory modeling technique. The proposed framework used both metric data collected from a system of programs and self-evaluations from experts. They noted that when they compared models based on the standard Fuzzy system, models based on Fuzzy Network improved transparency by more than 71.3% and accuracy by more than 11.0%. It was emphasized that the framework based on the fuzzy grid was more suitable for building an SMP model.

In 2020 Gupta S. and Chug A. [20] presented an enhanced Random Forest Algorithm (RFA) method for software maintainability prediction. It was presented for software maintainability prediction. The suggested method combines the Random Forest (RF) algorithm with three popular feature selection methods, including Chi-Squared, RF, and Linear Correlation Filter, as well as a resampling method, intending to increase the core RF algorithm's prediction accuracy. QUES and UIMS, two commercially accessible datasets, are subjected to Enhanced – RFA, and performance is assessed using R2. Results indicated that the proposed approach performs significantly better than RFA for both the datasets with an improvement in R2 values equal to 69.50%, 65.57% & 69.40% for QUES and 31.90%, 44.94% & 51.81% for UIMS using chi-squared, RF and linear correlation filter techniques respectively.

Table 1: Summary of Previous Studies

S.NO.	Author	year	Methods Used
1.	Chikako V. K. and Gray A. R.	2006	Bayesian network
2.	Yuming Z. and Leung H.	2007	Multivariate Adaptive Regression Splines(MARS)
3.	WANG L., HU X., NING Z. and Ke W.	2009	Projection Pursuit Regression(PPR),Artificial Neural Network(ANN), Multivariate Adaptive Regression Splines(MARS)
4.	Mahmoud O. E. and Karim O. E.	2009	TreeNets
5.	Kaurand A., Kaur K. and Malhotra R.	2010	Artificial Neural Network, Fuzzy Inference System (FIS), Adaptive Neuro Fuzzy Inference System (ANFIS)
6.	Jin C. and Liu J. A.	2010	Support Vector Machine, clustering
7.	Malhotra R. and Chug A.	2012	Group Method of Data Handling (GMDH), Probabilistic Neural Network (PNN) & Genetic Algorithms (GAs)
8.	Ahmed M. A. and Al-Jamimi H. A.	2013	Fuzzy-logic
9.	Yadav V. K. and Kumar S. and Mittal M.	2015	K-means and Hierarchical clustering
10.	Kumar L. and Rath S. K.	2016	Artificial neural network (FLANN) with genetic algorithm (GA), particle swarm optimization (PSO) and clonal selection algorithm (CSA)
11.	Jain A., Tarwani S. and Chug A.	2016	Evolutionary technique particularly genetic algorithm
12.	Kumar L. and Rath S. K.	2017	Used NeuroFuzzy approach: hybrid neural network and fuzzy logic
13.	Baskar N. and Chandrasekar C.,	2018	Neuro-PSO model
14.	Wang X., Gegov A.,Farzad A., Chen Y. and Hu Q.	2019	Fuzzy network
15.	Gupta S. and Chug A.	2020	Enhanced-RFA (Random Forest Algorithm)

III. SOFTWARE MAINTENANCE

The software has gone through several changes throughout its lifetime. Some reasons are causing these changes. They include:

- Changes in software requirements
- An erroneous software design
- Change in environment
- A change in hardware demands a change in the software too

Software maintenance is an important phase in the software life cycle. Its basic objective is to keep the software fully functional and up-to-date. Due to the changing requirements on performance over time, software products are modified to provide new functionalities or to change the existing ones. Software maintenance can be categorized into four types: adaptive, corrective, preventive, and perfective maintenance [21]:

Adaptive maintenance is performed in order to adapt the software to a new environment, such as new hardware, a new operating system, or a new database system. Corrective

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 10, pp 56-62, October-2022

https://doi.org/10.47001/IRJIET/2022.610009

maintenance is performed to correct any faults in the analysis, design, implementation, documentation, or any other type of fault. A software system is said to be repairable or corrective if it allows the removal of residual errors present in the product while its delivery and the errors introduced in the software during its maintenance. Preventive maintenance is performed to keep the software product working and/or to extend its life. Perfective maintenance is performed to improve production efficiency by increasing its speed, adding/modifying new/existing functionalities, and enhancing the performance level.

IV. OBJECT-ORIENTED METRICS

The available studies related to software maintainability prediction focused on object-oriented metrics. The relationship between OO metrics and software maintenance effort is complex and nonlinear [5]. Hence these object-oriented metrics, which is a good platform to assess maintenance effort.

The UIMS and QUES datasets, which Li and Henry [22] released, are two well-known object-oriented maintainability datasets used in these studies. These datasets were selected mainly because they have recently been used to assess efficacy of several machine learning models in predicting the maintainability of object-oriented software[8].The UIMS dataset contains class-level metrics data collected from 39 classes of a user interface management system, whereas the QUES dataset contains the same metrics collected from 71 classes of a quality evaluation system. Both systems were implemented in Ada. Eleven class-level measures total, ten independent and one dependent, are presented in both datasets. Five Chidambar and Kemerer metrics [23]—WMC, DIT, NOC, RFC, and LCOM—are the independent variables; four Li and Henry metrics [22]: MPC, DAC, NOM, and SIZE2; and one traditional line of code metric (SIZE1). The number of lines of code that were modified for each class during a three-year maintenance period serves as the dependent variable for the maintenance effort surrogate measure (CHANGE).A line change could be an addition or a deletion. A change in the content of a line is counted as a deletion and addition. Table 2 defines each metric in the datasets.

Table 2: Metrics Definition

Metric	Definition
WMC (Weighted Methods per Class)	This metric measures the static complexity of all methods of a class.
DIT (Depth of Inheritance Tree)	DIT is the maximum depth of the inheritance hierarchy of each class.
NOC (Number of Children)	It counts the number of direct children for each class.
RFC (Response For a Class)	RFC is used to measure the cardinality of the responses of all the classes used for the study.
LCOM (Lack of Cohesion of Methods)	It is used to measure the lack of cohesion of a class.
MPC (Message Passing Coupling)	This metrics indicate the number of messages passed between the objects of the class.
DAC (Data Abstraction Coupling)	DAC metric measures the number of abstract data types defined in a class
NOM (Number of Methods)	The NOM metric counts the number of local methods of a class.
SIZE1 (Lines of code)	The number of lines of code excluding comments
SIZE2 (Number of properties)	The total count of the number of data attributes and the number of local methods in a class
CHANGE (Number of lines changed)	The number of lines added and deleted in a class, and changes in the content are counted as 2

V. CONCLUSION

This survey discusses software maintainability that introduces a set of different papers and ideas for many researchers that used UIMS and QUES datasets help predication criteria. In software engineering, maintainability is required for the correction of defects, meeting new requirements, coping with a changing environment, and modifying a software product with ease. Maintainability can also affect time, it restores the system to its operational status following a failure or removal from operation for an upgrade. Software maintainability identifies and fixes a fault is required.

These studies are used numerous machine learning techniques to predict software maintainability, which was dependent on the historical dataset including software metrics that have a strong association with software maintainability.

REFERENCES

[1] "IEEE Standard for Software Maintenance," *IEEE Std* 1219-1993, 1993.

ISSN (online): 2581-3048
Volume 6, Issue 10, pp 56-62, October-2022
https://doi.org/10.47001/IRJIET/2022.610009

[2] Ahmed M. A. and Al-JamimiH. A., "Machine learning approaches for predicting software maintainability: a fuzzy-based transparent model," *IET Softw.*, vol. 7, no. 6, pp. 317–326, 2013.

[3] IEEE Std. 610.12-1990. Standard Glossary of Software Engineering Terminology, *IEEE Computer Society Press, Los Alamitos, CA*, 1993.

[4] Oman P. and Hagemeister J., "Construction and Testing of Polynomials Predicting Software Maintainability*", Journal of Systems and Software* 24.3 (1994): 251-266.

[5] Dash Y., Dubey S.K. and Rana A., "Maintainability Measurement in Object Oriented Paradigm", *International Journal of Advanced Research in Computer Science (IJARCS)*, Vol.3, no.2, April 2012, pp. 207-213.

[6] DeMarco T., Controlling software projects: Management, measurement, and estimates. *Prentice Hall PTR Upper Saddle River, NJ, USA*, 1986.

[7] Chikako V. K. and Gray A. R., "An application of Bayesian network for predicting object-oriented software maintainability." *Information and Software Technology* 48.1 (2006): 59-67.

[8] Yuming Z. and Leung H., "Predicting object-oriented software maintainability using multivariate adaptive regression splines." *Journal of systems and software* 80.8 (2007): 1349-1361.

[9] WANG L., HU X., NING Z. andKe W., "Predicting object-oriented software maintainability using projection pursuit regression." *2009 First International Conference on Information Science and Engineering. IEEE*, 2009.

[10] Mahmoud O. E. and Karim O. E.,"Application of tree net in predicting object-oriented software maintainability: A comparative study." *2009 13th European Conference on Software Maintenance and Reengineering. IEEE*, 2009.

[11] KaurandA., Kaur K. and Malhotra R., "Soft computing approaches for prediction of software maintenance effort." *International Journal of Computer Applications* 1.16 (2010): 69-75.

[12] Jin C. and Liu J. A., "Applications of support vector machine and unsupervised learning for predicting maintainability using object-oriented metrics." *2010 Second International Conference on Multimedia and Information Technology*. Vol. 1. IEEE, 2010.

[13] Malhotra R. and Chug A., "Software Maintainability Prediction using Machine Learning Algorithms," *Softw. Eng. an Int. J.*, vol. 2, no. 2, pp. 19–36, 2012.

[14] Yadav V.K. and Kumar S. and Mittal M., "Prediction of Software Maintenance Effort of Object Oriented Metrics Based Commercial Systems." *African Journal of Computing & ICT* 8.1 (2015): 163-172.

[15] Kumar L.and RathS. K., "Hybrid functional link artificial neural network approach for predicting maintainability of object-oriented software," *J. Syst. Softw.*, vol. 121, pp. 170–190, 2016.

[16] Jain A., Tarwani S. and Chug A., "An empirical investigation of evolutionary algorithm for software maintainability prediction." *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS). IEEE*, 2016.

[17] Kumar L. and RathS. K., "Software maintainability prediction using hybrid neural network and fuzzy logic approach with parallel computing concept," *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. 2, pp.1487–1502, 2017.

[18] BaskarN. and Chandrasekar C., "Optimized Neuro-PSO-based Software Maintainability Prediction Using Relief Features Selection Method." *International Journal of Pure and Applied Mathematics* 119.18 (2018): 1-15.

[19] Wang X., Gegov A., Farzad A., Chen Y. and Hu Q., "Fuzzy network based framework for software maintainability prediction." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 27.05 (2019): 841-862.

[20] Gupta S. and Chug A., "Software maintainability prediction using an enhanced random forest algorithm." *Journal of Discrete Mathematical Sciences and Cryptography* 23.2 (2020): 441-449.

[21] Olatunji S. O., Rasheed Z., Sattar K. A., Al- Mana A. M., Alshayeb M. and El-Sebakhy E. A., "Extreme Learning Machine as Maintainability Prediction Model for Object-Oriented Software Systems", *Journal of Computing*, vol. 2, no. 8, pp. 49-56, 2010.

[22] Li W. and Henry S., "Object-Oriented Metrics that Predict Maintainability,*" Journal of Systems and Software*, vol. 23, pp. 111-122, 1993.

[23] Chidamber S. R. and Kemerer C. F., "A Metrics Suite for Object Oriented Design,*" IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, 1994.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 6, Issue 10, pp 56-62, October-2022

https://doi.org/10.47001/IRJIET/2022.610009

Citation of this Article:

Mazin Mohammed Ismael, Ibrahim Ahmed Saleh, "Software Maintenance Potential Prediction Based on Machine Learning" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 6, Issue 10, pp 56-62, October 2022. Article DOI https://doi.org/10.47001/IRJIET/2022.610009

****

