Evaluation of Software Effort Estimation Methods for Machine learning techniques

[1]Montaser Fadulalla Adam , [2]Haroun Abdalla Eissa

[2]Faculty of computer scince , [2]Associate Professor

[1] Alneelain University ,khartoum , [2] University of Alsharg,kassala

E.mail:montaser.fadulalla@gmail.com

Abstract

Software development is a complicated process and includes a unique effort that Include many activities, resources, skills, and people to build a quality product. Thus, effort estimation is very important activity in scheduling of software project in order to deliver project on time and better effectively evaluate predictions. There are many models used in software effort estimation, including algorithmic, non-algorithmic, and machine learning models. This paper present a review of deferent machine learning methods that are using for effort estimation like regression models(liner regression ,decision trees, random forest), neural networks , and then evaluate this models based on performance criteria such as MAE (Mean Absolute Error) and R^2 Score. These models were tested on desharnias data using Python, The results were compared for the different models and gives notes on each model. It was found that the Random forest model is the best with deshanias data among the four models and achieved the highest score on the R SCORE scale, which amounted to 0.75

Keywords: Effort, Estimation ,Machine Learning , Neural Networks, Regression Models

تقييم أساليب تقدير جهد البرمجيات لتقنيات التعلم الآلي

[1] منتصر فضل الله ادم، [2] هارون عبد الله عيسى

[1] كلية علوم الحاسوب ، [2] استاذ مشارك كلية علوم الحاسوب

[1] جامعة النيلين ، [2] جامعة الشرق

البريد الالكتروني للباحث المرسل:montaser.fadulalla@gmail.com

مستخلص الدراسة

تطوير البرمجيات عملية معقدة وتتضمن جهدًا فريدًا يشمل العديد من الأنشطة والموارد والمهارات والأشخاص لبناء منتج عالي الجودة. وبالتالي، يعد تقدير الجهد نشاطًا مهمًا جدًا في جدولة مشروع البرمجيات من أجل تسليم المشروع في الوقت المحدد وتقييم التنبؤات بشكل أفضل وفعال. هناك العديد من النماذج المستخدمة في تقدير جهد البرمجيات، بما في ذلك النماذج الخوارزمية وغير الخوارزمية ونماذج التعلم الآلي. تقدم هذه الورقة مراجعة لطرق التعلم الآلي المختلفة التي تستخدم لتقدير الجهد مثل نماذج الانحدار (الانحدار الخطي، وأشجار القرار، والغابة العشوائية"Random Forest")، والشبكات العصبية، ثم تقييم هذه النماذج بناءً على معايير الأداء مثل متوسط الخطأ المطلق (MAE) و R^2 . تم اختبار هذه النماذج على بيانات deshanias باستخدام برنامج بايثون، وتمت مقارنة النتائج للنماذج المختلفة وإعطاء ملاحظات على كل نموذج. وقد وجد أن نموذج الغابة العشوائية هو الأفضل مع بيانات deshanias من بين النماذج الأربعة وحقق أعلى درجة على مقياس R SCORE والتي بلغت 0.75.

الكلمات المفتاحية: الجهد ، التقدير، تعلم الالة ، الشبكات العصبية ، نماذج الأنحدار

1. Introduction

Effort estimation of different parts of software development affects anything from project proposal investigation, project management, analysis, and design, to product quality and efficiency, customer satisfaction and success in the market. Effort estimation is applied as an input for project planning, iterations planning, budgeting, capital analysis, pricing process and tendering. Accuracy of the software development effort estimation is_one of the challenges for every software project since it has a severe impact on expense, timing, functionality, and the development software quality[1] .

Planning, monitoring, and controlling software development projects need the effort and expenses estimated properly [2]. If we can exactly estimate effort for the project, quality and efficiency are controllable, since there is no need for many changes sporadically causing the quality and efficiency sacrificed.

In fact, software estimation in the preliminary phase of the development life cycle process surely decreases the risks [2]. Rapid and accurate estimation of software projects development effort in the information technology industry is determinant and fundamental [3].

Software effort estimation (SEE) is the prediction about the amount of effort required to make a software system and its duration [4]. There is lot of models for the effort and cost estimation, since there is no unique model that completely satisfies the need for objective, fast and accurate predictions in all circumstances. These models were Separation with three major awarded: algorithmic and non-algorithmic and machine learning.

Algorithmic Methods based on the special algorithm. They usually need data at first and make results by using mathematical relations. The Differences among the existing algorithmic methods are related to choosing the cost factors and function.

Project factors: multisite development; use of software tools; required development schedule. [5]

Non Algorithmic Methods is based on analytical comparisons and inferences. For using some information about the previous projects which are similar to the underestimate project is required and usually, the estimation process in these methods is done according to the analysis of the previous datasets. [6]

Machine learning is an alternative to algorithmic model building. Artificial neural networks (ANN), case-based reasoning (CBR), decision trees (DT), fuzzy models, regression models, and genetic algorithms are all examples of machine learning estimation approaches[7] .

A machine learning method plays an important role in effort estimation because it can increase the efficiency of estimation by applying the training rule to estimate a correct effort required [8].

2. Related Work

Machine learning techniques are being widely used in many fields, and in effort estimation used as theoretically alternative to traditional methods, Machine learning techniques are proving very useful to accurately predict software effort values. They are many studies that have addressed techniques for estimating effort using machine learning.

Jyoti Shivhare, Santanu Ku. Rath (2014): This paper presents an approach for estimation based upon machine learning techniques for non-quantitative data and is carried out in two phases. The first phase concentrates on selection of optimal feature set in high dimensional data, related to projects undertaken in past. A quantitative analysis using Rough Set Theory is performed for feature reduction. The second phase estimates the effort based on the optimal feature set obtained from first phase. The estimation is carried out directly by applying Naive Bayes Classifier and Artificial Neural

Network techniques respectively. The feature reduction process in first phase considers public domain data

(USP05). The performance of the proposed methods is evaluated and compared based on the parameters such as Mean Magnitude of Relative Error (MMRE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Correlation Coincident. It is observed that Naive Bayes classifier achieved better results for estimation when compared with that by using Neural Network technique. [9].

Monika ,Om Prakash Sanguan,(2017): This paper presents a review of various machine -learning techniques using in estimation of software project effort namely Artificial Neural Network, Fuzzy logic, Analogy estimation etc. Machine learning techniques consistently predicting accurate results because of its learning natures form previously completed projects. This paper summarizes that each technique has its own features and behave differently according to environment so no technique can be preferred over each other[10].

Vehbi YURDAKURBAN , Nadia ERDOĞAN (2018): In this work, Machine Learning based software effort estimation methods are compared and their error rates are documented. Decision Tree, Naive Bayes and Multiple Regression models were inspected and they were trained and tested using data obtained from a local software house[11].

Ashwni kumar, Dr D.L.Gupta,(2019): This paper focus on performance of M5 Rule, Decision Table, Conjection Rule, Zero Rule classifier is experimented for software effort estimation. The performance measures criteria are based on RMSE and MAE values. The result shows that the M 5 Rule technique gives best performance in software effort estimation model. [12].

Mizanur Rahman , Partha Protim Roy, Mohammad Ali, Teresa Goncalves,(2023): This study recommends various machine learning algorithms for estimating, including k-nearest neighbor regression, support vector regression, and decision trees. These methods are now used by the software development industry for software estimating with the goal of overcoming the limitations of parametric and conventional estimation techniques and advancing projects. Our dataset, which was created by software Company called Edusoft Consulted LTD, was used to assess the effectiveness of the established method. The three commonly used performance evaluation measures, mean absolute error (MAE), mean squared error (MSE), and R square error, represent the base for these. Comparative experimental results demonstrate that decision trees perform better at predicting effort than other techniques [13]..

P. V. Terlapu, K. K. Raju, G. Kiran Kumar, G. Jagadeeswara Rao, K. Kavitha and S. Samreen(2024): This study systematically reviewed the literature on effort-estimating models from 2015-2024, identifying 69 relevant studies from various publications to compile information on various software work estimation models. This review aims to analyze the models proposed in the literature and their classification, the metrics used for accuracy measurement, the leading model that has been chiefly applied for effort estimation, and the benchmark datasets available. The study utilized 542 relevant articles on software development, cost, effort, prediction, estimation, and modelling techniques in the search strategy. After 194 selections, the authors chose 69 articles to understand ML applications in SEE comprehensively. The researchers used a scoring system to assess each study’s responses (from 0 to 5 points) to their research questions. This helped them identify credible studies with higher scores for a comprehensive review aligned with its objectives. The data extraction process identified 91% (63) of 69 studies as either highly or somewhat relevant, demonstrating a successful search strategy for analysis. The literature review on SEE indicates a growing preference for ML-based models in 59% of selected studies. 17% of the studies chosen favor hybrid models to overcome software development challenges. They qualitatively analyzed all the literature on software effort estimation using expert judgment, formal estimation techniques, ML-based techniques, and hybrid techniques. They discovered that researchers have frequently used ML-based models to estimate software effort and are currently in the lead. This study also explores the application of feature importance and selection in machine learning models for Software Effort Estimation (SEE) using popular algorithms like support Vector Machine (SVM), AdaBoost (AB), Gradient Boost (GB), and Random Forest (RF) with six benchmark datasets like CHINA, COCOMO-NASA2, COCOMO, COCOMO81, DESHARNAIS, and KITCHENHAM [24].

Muhammad Abid, Sama Bukhari, Muhammad Saqlain(2025): This paper discloses the techniques that utilize machine learning models for ameliorating software effort estimation by using biomedical datasets, including Breast Cancer Wisconsin, COVID-19, Sleepy Drivers EEG Brainwave, Heart Disease Prediction and Food Nutrition. All of these datasets are being trained by four popular machine learning models; Linear Regression, Gradient Boosting, Random Forest, and Decision Tree. Furthermore, correlation based features are selected in the feature matrix to investigate the influence of statistically linked features and to promote reliability. For evaluation and measurement of the effectiveness of these models, two performance metrics namely: R2 and Root Mean Squared Error are employed. The outcomes of the study delineate that Linear Regression and Gradient Boosting models give substantially better results than other models when choosing features on the basis of correlation. R2 scores are strikingly impressive for Food Nutrition, Breast Cancer, COVID-19, while RMSE scores are lowest for COVID-19[25].

Vaishali Thakur, Kamlesh Dutta(2025): This research evaluates the performance of machine learning models for the estimation of software effort using diverse datasets with distinct feature sets and varying training-test splits. To address the risks of under or overestimation in project management, the study employs robust data preprocessing, feature engineering, and selection techniques. Through extensive experimentation, the study analyzes model accuracy using multiple metrics such as Mean Absolute Error, R-squared, precision, recall, F1-score, and Receiver Operating Characteristic Area Under the Curve. Preprocessing techniques, including feature scaling and outlier removal, were applied to reduce overfitting and improve generalization. Performance was assessed using various indicators, with results showing that the stacked ensemble model consistently outperformed other models like Linear Regression, Support Vector Regression, and Gradient Boosting in generating accurate predictions across datasets. The analysis revealed that no single data split yielded optimal performance for all datasets; instead, different datasets performed best under customized splits. Larger datasets performed well with splits like 70-30, while smaller ones exhibited overfitting even after preprocessing. Models like Random Forest, XGBoost, and Gradient Boosting demonstrated robust performance, especially on larger datasets, while smaller datasets highlighted challenges with models like K-Nearest Neighbors and Naive Bayes due to overfitting and poor generalization. The research highlights that larger datasets enable better generalization, while smaller datasets require careful model and preprocessing choices to mitigate over fitting and achieve reliable results[26].

3. Methodology

There has been extensive study into software effort estimation based on machine learning .The goal of this machine learning method is to provide accurate estimation for software effort. in this paper we will discuss four machine learning methods for software effort estimation, namely regression models(linear , random forest),Decision Tree and neural networks. We will use the Python language to apply it on (desharnais) dataset, and then the models will be evaluated based on the MAE and R Score measures. Figure 1 show the methodology used to apply and evaluate the models.

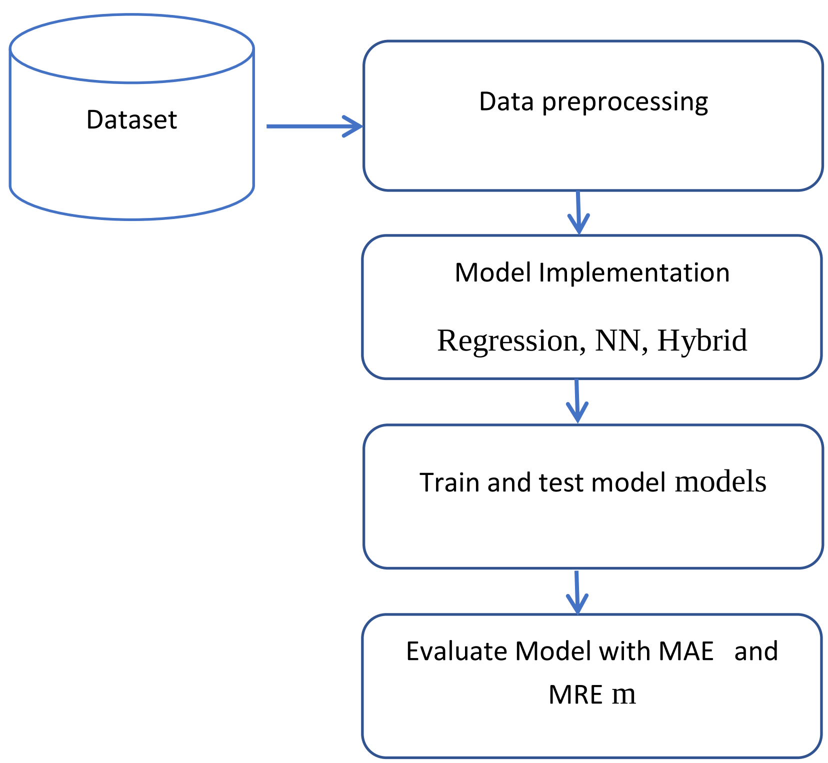

Fig. 1. Methodology of evaluate model

(3.1) Dataset

Will use the Desharnais dataset which is composed of a total of 81 projects developed by a Canadian software house in 1989. Each project has twelve attributes which are described in table(1)[14].

Attribute	Description	Variable type
Project	Project ID which starts by 1 and ends by 81	Numeric
TeamExp	Team experience measured in years	Numeric
ManagerExp	Manager experience measured in years	Numeric
YearEnd	Year the project ended	Numeric
Length	Duration of the project in months	Numeric
Effort	Actual effort measured in person-hours	Numeric
Transactions	Number of the logical transactions in the system	Numeric
Entities	Number of the entities in the system	Numeric
PointsNonAdjust	Size of the project measured in unadjusted function points. This is calculated as Transactions plus Entities	Numeric
Envergure	Function point complexity adjustment factor. This is based on the General Systems Characteristics (GSC). The GSC has 14 attributes; each is rated on a six-point ordinal scale. $Envergure = \sum_{i=1}^{14} GSC_i$	Numeric
PointsAdjust	Size of the project measured in adjusted function points. This is calculated as: PointsAdjust PointsNonAdjust Envergure = × + × (0.65 0.01)	Numeric
Language	Type of language used in the project expressed as 1, 2 or 3. The value "1" corresponds to "Basic Cobol", where the value "2" corresponds to "Advanced Cobol" and the value "3" to 4GL language	Categorical

(3.2) Methods of Machine Learning using in software Effort Estimation

Machine Learning can increase the accuracy of estimation by training rules of estimation and repeating the run cycles. These are some of the methods used in software effort estimation:

(3.2.1) Linear Models

In the linear model-based cost estimation, the most preferred and used method of estimation is regression analysis[15] .Commonly linear models have the simple structure and trace a clear equation as below:

$$\text{Effort} = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$$

Where,a1,a2..,an are selected according to the project information.

(3.2.2.) Random Forest

The Random Forest (RF) classifier, a variant of Bagging, includes a collection of tree-structured classifiers. RF employs random feature selection and bootstrap models using DT [17]. In the RF algorithm, K attributes are randomly chosen at each node to construct a classification tree. For classification, RF predicts the majority class among the predictions made by the individual trees [19, 20]. If the forest consists of T trees, the number of votes received by class m can be calculated as follows:

$$U_m = \sum_{t=1}^{T} I(\check{C}_t == m)$$

(3.2.3) Decision Tree

A decision tree (DT) is a valuable decision support tool that uses a tree-like model to represent decisions and their outcomes [17]. The DT classifier categorizes examples by sorting them based on their feature values. Each node in the tree represents a feature of the example to be classified, and the branches from the node represent possible values for that feature. The DT classifier constructs a tree using the C4.5 algorithm, which partitions the data into smaller subsets and assesses the difference in entropy (normalized information gain). It then makes decisions based on the attribute with the highest information gain [18]. The entropy formula is shown below:

$$s_j = f\left(\sum_{i=1}^{N} s_i \cdot \omega_{ji} + \theta_j\right)$$

(3.2.4) Neural networks

ANNs have been used for over fifteen years to predict software[15] . Neural networks are nonlinear computational models, inspired in the human brain structure and operation that aim to reproduce human features, such as learning, association, generalization, and abstention. Neural networks are made up of various processing elements (PEs) (artificial neurons), highly interconnected, that perform simple operations, transmitting their results to the neighboring processors. The neural networks ability to perform nonlinear mappings between its inputs and outputs have made then prosperous in pattern recognition and complex systems modeling.

The neural network PE is a simplified mathematical representation of the biological neuron, which executes the sum of its inputs si (dendrites) modified by the associated weights wji (synaptic weights). Each PE then applies an activation function to that result in order to generate its output sj (axon). The applied activation function is usually a nonlinear function f (e.g. sigmoid or hyperbolic tangent functions), a feature that enables the ANN to represent more complex problems. Therefore, the output of a PE is calculated as in Eq[21]:

$$E(F) = \sum_{i=1}^{c} - p_i \ log_2 p_i$$

where N is the total number of inputs and θj is a bias term that has the effect of increasing or reducing the net input of the activation function[21].

(3.3) Experiment and Evaluation

The previous models will be tested in software effort estimation using Python, and each model will be evaluated based on the MAE and R Score metrics.

The tests will be conducted on the previously mentioned (desharnias) dataset.

Measures used in the evaluation

MAE (Mean Absolute Error) calculates the average absolute difference between predicted values and actual values [22]. It gives a clear understanding of how far off predictions are, with lower values indicating better performance.

R-squared (R^2) represents the proportion of variance in the target variable that is explained by the model. An R^2 value of 1 indicates a perfect fit, while 0 indicates no variance explained[23].

(3.3.1) Linear Models

Estimating effort using Linear Regression on the Desharnais dataset using Python. The results of the experiment were as follows:

MAE(Mean Absolute Error)= 500

^{2}R Score =0.65

Most influential features (from regression coefficients):

"Length" and "Transactions" are usually the ones most closely related to effort.

(3.3.2) Random Forest

Software effort estimation using Random Forest on the Desharnais dataset using Python. The results of the experiment were as follows:

MAE(Mean Absolute Error)=400

^{2}Score R =0.75

Most important features Typically, "Transactions", "Entities", and "Length" are the most influential.

It is a significant improvement over linear regression, due to the model's ability to handle non-linear relationships and interactions between features.

(3.3.3) Decision Tree

Software effort estimation using Decision Tree on the Desharnais dataset using Python. The results of the experiment were as follows:

MAE(Mean Absolute Error)= 480.35

²Score R =0.62

- The tree shows partitioning rules based on features (such as `Transactions” or “Entities”).

(3.3.3) Neural Network

Software effort estimation using Neural Network on the Desharnais dataset using Python. The results of the experiment were as follows:

MAE(Mean Absolute Error)=450

²Score R=0.7

(3.4) Comparison Methods

This comparison in the table (2) could be useful for changing for an appropriate method in a particular project and environment. The table shows some machine learning models mentioned for software effort estimation and compares them based on the MAE and R Score metrics by applying to Desharnais dataset and providing notes on deferent models. For making the comparison the popular existing method has been selected،

Table (2) explains the Comparison between methods in Software Engineering.

Sr. no	Model	Type	MRE	R^2 SCORE	NOTES
1	Linear model	Algorithmic	500	0.65	Simple but limited
2	Random Forest	Machine learning	400	0.75	Optimized for this data
3	Decision Tree	Machine learning	460.35	0.62	Needs more data
4	Neural net works	Machine learning	450	0.7	Is better at nonlinear modeling

4. Experimental Results and Discussion

Previous studies have shown that machine learning methods are superior and more accurate than traditional methods at estimatingSoftwareained methods are effort. Constr not adaptable to unconventional projects and rely heavily on human estimates. Machine learning methods, on the other hand, handle nonlinear relationships, are characterized by and can be used across ،automatic adaptation, improving as new data becomes available .diverse projects

In this study, the machine learning model experiment achieved good results in estimating programming effort when tested on the Deshanias dataset, and theR^2score achieved ich are considered higher accuracy than scores ranging between 0.65 and 0.75, wh traditional methods as follows:

linear regression achieved ($R^2 = 0.65$) while models such as Random Forest achieved ($R^2 = 0.75$) and neural network achieved ($R^2 = 0.7$).

- Reason for the relative weakness of linear regression: The nonlinear nature of the data and the dependence of effort on complex interactions between features.

Random Forest deals with nonlinear relationships it does not require linear assumptions between features and effort.

Decision Tree provides slightly better results than linear regression in modeling nonlinear relationships, but is less accurate than Random Forest.

Neural networks provides better results than Linear Regression and Decision Tree , but is less accurate than Random Forest.

5. Recommendations

Based on the previous results, we recommend using machine learning techniques to large projects, it's best to start with simple models -to-estimate software effort. For medium When rich .like Linear Regression and progress to advanced models like Random Forest historical data is available, it's preferable to use neural networks or hybrid models. When results need to be interpreted, interpretable models (Decision Trees) are used.

6. Conclusion

Machine learning improves the accuracy of software effort estimation compared to traditional methods, but its success depends on the quality of the data and an understanding of the project context.

Random Forest is an excellent choice for estimating the software effort on the Desharnais dataset, as it significantly improves accuracy compared to other models.

Neural networks are not the best choice for the small Desharnais dataset, but they show reasonable results.

References:

1- Yavari Y, Bashiri H. Effort Estimation of Software Development: Depth View on IRAN Case. American Journal of Software Engineering. 2013;1(1):5-11.
2. Abdel-Hamid TK, Sengupta K, Swett C. The impact of goals on software project management: An experimental investigation. MIS quarterly. 1999:531-55.
3. Singh Y, Bhatia PK, Kaur A, Sangwan O, editors. A review of studies on effort estimation techniques of software development. Proceedings of Conference Mathematical Techniques: Emerging Paradigms for Electronics and IT Industries; 2008.
4. Yavari1 Y, HB, * Effort Estimation of Software Development: Depth View on IRAN Case American Journal of Software Engineering. 2013;1.
5. 1Vahid Khatibi, Jawawi DNA. Software Cost Estimation Methods: A Review Journal of Emerging Trends in Computing and Information Sciences 2010; 2 (2079-8407).
6. BINGCHIANG JENG1, DOWMING YEH2, DERON WANG3, SHU-LAN CHU2 and others. A Specific Effort Estimation Method Using Function Point. JOURNAL OF INFORMATION SCIENCE AND ENGINEERING. 2011;27:1363-76
7. Parashar UD, Gibson CJ, Bresee JS, Glass RI. Rotavirus and severe childhood diarrhea. Emerging infectious diseases. 2006;12(2):304.
8. Mamoona Humayun and Cui Gang, "Estimating Effort in Global Software Development Projects Using Machine Learning Techniques" , International Journal of Information and Education Technology, Vol. 2, No. 3, June 2012
9. Jyoti Shivhare, Santanu Ku. Rath "Software Effort Estimation using Machine Learning Techniques",2014. ISEC '14 February 19 - 21 2014, Chennai, India ACM 978-1-4503-776-
10. Monika ,Om Prakash Sanguan," Software Effort Estimation Using Machine Learning Techniques" 2017 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence.
11. Vehbi YURDAKURBAN , Nadia ERDOĞAN" Comparison of Machine Learning Methods for Software Project Effort Estimation" , 2018 26th signal processing and communications application conference (SIU),izmir,Turkey,2018,pp.1-4,
12. Ashwni kumar, Dr D.L.Gupta," A Comparison of Machine Learning Techniques for Software Effort Estimation",2019, Journal of Emerging Technologies and Innovative Research (JETIR), June 2019, Volume 6, Issue 6
13. Mizanur Rahman , Partha Protim Roy, Mohammad Ali, Teresa Goncalves,"Software Effort Estimation using Machine LearningTechnique", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 14, No. 4, 2023
14. J. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," Master's Thesis, University of Montreal, 1989.
15. Fatih Yücalar*, Deniz Kilinc "Regression Analysis Based Software Effort Estimation Method "International Journal of Software Engineering and Knowledge Engineering Vol. 26, No. 5 (2016) 807–826 #.c World Scienti¯c Publishing Company
16. Dinesh R. Pai, Kevin S. McFall & Girish H. Subramanian (2013) Software Effort Estimation Using a Neural Network Ensemble, Journal of Computer Information Systems, 53:4, 49-58, DOI: 10.1080/08874417.2013.11645650
17. Alsawalqah H, Hijazi N, Eshtay M, Faris H, Radaideh AA, Aljarah I, Alshamaileh Y (2020) Software Defect Prediction Using Heterogeneous Ensemble Classification Based on Segmented

Patterns, Applied Sciences, Vol. 10, No. 5, p. 1745, h t t p s : / / d o i . o r g / 1 0 . 3 3 9 0 / a p p 1 0 0 5 1 7 4 5

18. Sharma S, Kumar S (2018) Analysis of Ensemble Models for Aging Related Bug Prediction in Software Systems, Proceedings of the 13th International Conference on Software Technologies, Porto, Portugal, pp. 256–263, h t t p s : / / d o i . o r g / 1 0 . 5 2 2 0 / 0 0 0 6 8 4 7 7 0 2 9 0 0 2 9 7
19. Ferenc R, Tóth Z, Ladányi G, Siket I, Gyimóthy T (2018) A Public Unified Bug Dataset for Java, In Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering, Association for Computing Machinery, New York, USA, pp.12–21, h t t p s : / / d o i . o r g / 1 0 . 1 1 4 5 / 3 2 7 3 9 3 4 . 3 2 7 3 9 3 6
20. Kaur A, Kaur K (2014) Performance analysis of ensemble learning for predicting defects in open-source software, International Conference on Advances in Computing, Communications and Informatics, Delhi, India, pp.219–225, h t t p s : / / d o i . o r g / 1 0 . 1 1 0 9 / I C A C C I . 2 0 1 4 . 6 9 6 8 4 3 8
21. B.K. Behera,P.K. Hari (2010), Woven Textile Structure, 15 Modeling for textile product design, Copyright © 2010 Woodhead Publishing Limited. All rights reserved.
22. Weijie Wang and Yanmin Lu 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **324** 012049**DOI** 10.1088/1757-899X/324/1/012049
23. Gao J. R-Squared (R2) – How much variation is explained? Research Methods in Medicine & Health Sciences. 2023;5(4):104-109. doi:10.1177/26320843231186398.
24. P. V. Terlapu, K. K. Raju, G. Kiran Kumar, G. Jagadeeswara Rao, K. Kavitha and S. Samreen, "Improved Software Effort Estimation Through Machine Learning: Challenges, Applications, and Feature Importance Analysis," in IEEE Access, vol. 12, pp. 138663-138701, 2024, doi: 10.1109/ACCESS.2024.3457771.
25. Abid, M., Bukhari, S. and Saqlain, M. (2025) “Enhancing Software Effort Estimation in Healthcare Based Feature -orrelationInformatics: A Comparative Analysis of Machine Learning Models with C .66–Selection”, Sustainable Machine Intelligence Journal, 10, pp. 50 .doi:10.61356/SMIJ.2025.10451
26. V. Thakur and K. Dutta, "Machine Learning based Effort Estimation Models for Software Development Projects related Datasets with diverse features," 2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2025, pp. 807-813, doi: 10.1109/CICTN64563.2025.10932601.