IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, September 2023

Comparative Analysis of Estimation of Effort in Machine-Learning Techniques

Monika Tiwari[1] and Dr. Kalpana Sharma[2]

Research Scholar, Department of CS, Bhagwant University, Ajmer, Rajasthan, India[1]

Assistant Professor, Department. of CS, Bhagwant University, Ajmer, Rajasthan, India[2]

Abstract: *Effort estimation in software engineering provides an important role for software development and management of project cost, quality and time. Over the past decades, software inference has been receiving significant attention from researchers and substantial research has been conducted using various techniques and algorithms of machine learning. This paper suggests various machine learning techniques such as Naive Bayes, Random Forest Logistic Regression, Stochastic gradient boosting, decision trees and story points for estimation to assess the prediction more efficiently. Nowadays these approaches to software estimation are used by software development industries to overcome the shortcomings of parametric and traditional estimation techniques, increasing project. A comparative study of the techniques mentioned in this paper has been presented and examined to critically evaluate the performance of these techniques.*

Keywords: Software, EE, ML, SD, Techniques etc.

I. INTRODUCTION

Machine learning has gained tremendous momentum in recent years, revolutionizing various industries with its predictive and analytical capabilities. However, successful deployment of machine learning projects requires careful planning, resource allocation, and efficient management. An important aspect of managing machine learning projects is estimating the effort required to develop, deploy, and maintain models. Effort estimation plays a critical role in project planning, cost management, and resource allocation, ensuring that machine learning projects are executed efficiently and within predefined constraints.

Estimating effort in machine learning involves predicting the time, personnel, and computational resources required for tasks such as data preprocessing, feature engineering, model training, evaluation, deployment, and maintenance. Accurate estimation is essential for project managers, stakeholders, and data scientists to make informed decisions and ensure project success.

The purpose of this comparative analysis is to explore and evaluate different techniques and methods used to estimate effort in machine learning projects. In particular, it will examine the strengths and weaknesses of different approaches and their suitability for different types of projects and organizations.

The main objectives of this study include:

Review of effort estimation techniques: A comprehensive examination of existing techniques and methods used to estimate effort in machine learning, including expert judgment, historical data analysis, and algorithmic models.

Performance metrics and evaluation: The analysis will delve into various performance metrics used to evaluate the accuracy and reliability of effort estimates. Common metrics include mean absolute error (MAE), root mean square error (RMSE), and others.

Data-driven approaches: Exploring data-driven approaches to effort estimation, including the use of historical project data, benchmarking, and machine learning algorithms to predict effort.

Expert opinion versus data-driven approaches: A comparison of traditional expert judgment-based approaches with data-driven methods to assess their accuracy and effectiveness in different scenarios.

Case Studies: Real-world case studies and examples will be included to illustrate the practical application of various inference techniques in machine learning projects in various domains such as healthcare, finance and e-commerce.

Challenges and Best Practices: Identification of common challenges and best practices in effort estimation for machine learning projects with a focus on reducing risks and improving estimation accuracy.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 2, September 2023

Future Trends: Discussion of emerging trends and technologies in machine learning effort assessment, such as integration of AI-based tools and automation.

The purpose of this comparative analysis is to provide valuable insight into the complex task of estimating effort in machine learning projects. By understanding the strengths and limitations of different approaches, organizations can enhance their project planning and management processes, ultimately leading to more successful and cost-effective machine learning implementations.

II. LITERATURE REVIEW

White. KR etal. (2022), Machine Learning is the latest trending term which plays an important role in various fields of medicine, research and industrial application. It is difficult to weigh the real values or value of software. The best way to estimate software development cost, effort, size, and time is based on previous experience in software development. To measure the standard cost of software, as a unit of software value, machine-learning algorithms are used to increase the level of end user satisfaction through accurate and quick calculations of software cost and effort estimation. In this research work, an innovative cost assessment for software project management was developed using improved artificial neural network models. Two publicly available datasets with different machine learning algorithms are compared and the results show that the proposed model has high accuracy and low error rate in predicting the first phase of cost and effort evaluation.

Gauthaman et al (2021) stated that recently, frustration of programming project is increasing due to lack of planning and limitations of financial planning [2]. Deren et al (2020) applied expense evaluation to board development using an ANN model [3]. Fangwei Ning et al (2020) proposed a three-dimensional CNN for feasibility cost evaluation [4]. Eric Mattel et al (2019) recommended quotes allow project directors to assess the deliverability of activities and feasibly control costs [5]. Mahmood et al (2019) build a product cost evaluation model using AI approach [6]. Michael et al (2018) applied neural convolution computation to cost evolution [7]. Przemys et al (2017) proposed different AI calculations for exertion and time evaluation [8]. TMS Elhag et al (1998) proposed ANNs for the development of programming projects [9]. Richa Yadav et al (2016) opine that the achievement of any enterprise is further characterized by a well developed amount and cost valuation strategy that deals with the ideal utilization of assets [10]. Murat Gunaydin et al (2004) investigate the usefulness of neural organization systems to overcome cost evaluation issues in the early stages of building configuration processes [11].

Machine learning techniques used

The following machine learning techniques are applied to various datasets that are considered to calculate the effort of a software product. The decision to choose machine learning techniques for the purpose of implementation in the proposed research is made on the basis of previous research studies conducted in the literature survey [12-15]. Many researchers have previously applied some of the following machine learning techniques for their research purpose. But none of these techniques were previously applied for inference using CP, UCP, Web, and SP datasets. Each proposed contribution also describes a detailed representation of the results obtained using these techniques for their respective datasets. Each contribution also shows a detailed comparison of these techniques with first results obtained from the literature to reach their performance.

Inspiration

The motivation for this paper is to provide the estimation community with a new approach to the estimation problem, which can complement current practices.

- Ineffective results from algorithmic models
- Lack of appropriate techniques to assess software developed using an object-oriented approach
- Lack of applicable procedures to estimate the effort required to develop web-based applications.
- Non-availability of proper estimation techniques for software developed using Agile methodology.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

IJARSCT

Impact Factor: 7.301

Volume 3, Issue 2, September 2023

Problem Description

Previous research has shown that approximately one third of projects go over their budget and are delivered late. Two-thirds of the projects exceed their original estimates. It is an exceptionally troublesome task for a manager or system analyst to estimate with accuracy the effort required to develop software, when many external parameters such as vague project definition, technical uncertainty, implementation complexity, team experience, etc. play a significant role. [11] Role. Therefore, project managers are usually not able to determine how much time and manpower will be required for a successful project. However, during the initial phase of SDLC, a valid estimate of the software is necessary to help the organization develop a qualitative product within the planned period.

Combative Analysis

It has been observed in the literature that analysts and practitioners have proposed many techniques for the purpose of software assessment. However, CP, UCP and SPA are among the definitive estimation models used due to their simplicity, rapidity and to some extent accuracy. Taking into account the experimental research work conducted, the research contributions, conclusive comments as well as the scope of future work are included in this thesis. The overall conclusion that can be drawn from the research work displayed in this thesis is that the various findings obtained are certainly beneficial to analysts, experts and product specialists, in light of the fact that CPA and UCP are fundamentally the object of -oriented software was used and adapted. By employing ML techniques to provide more accurate estimation results. To handle web-based applications, the ISBSG Release 12 dataset is employed and then optimized using various ML techniques to predict the outcomes more accurately. Similarly, SPA is an improved estimating model that can be applied to estimate the effort required to develop software using agile methodology. The obtained results have been optimized using various ML techniques to improve the accuracy of the estimated effort value. Of all the techniques used in various chapters, the SVR polynomial performs better in most cases. Each SVR kernel is based on some kernel function. Any operation for that kernel is performed with the help of their respective kernel functions. The RBF kernel uses the exponential function, while the sigmoid kernel uses the sigmoid function. Linear kernel is more preferable for linearly separable data. Therefore, by analyzing the results obtained, it is observed that different results (error and prediction accuracy values) are obtained using different kernels and the results obtained using SVR RBF kernel based effort estimation model are CPA. outperform the results obtained from other models for SPA as well as UCP for web applications. Calculations were performed for the above methods, and results were obtained using MATLAB.

All the models proposed for agile software estimation are developed assuming that an initial project velocity value is given. This value is derived from previous projects developed by the same team under similar working conditions. But when a team is new, the company may not have any track record of it. In that case, no clear assignment can be made to the initial project velocity. The dataset collected from [7] for the purpose of estimating agile software does not provide any information on the types of projects considered for this study. For the results obtained to be valid for the general software engineering paradigm, it is desired to be based on working data that includes all categories of software developed using various agile methodologies.

Comparative analysis shows parameter values obtained by employing all 8 different machine learning algorithms: SVM (on all 4 kernels: linear, polynomial, RBF and sigmoid), RF, SGB, DT, KNN, LR, NB and MLP from 12 More different datasets namely Albrecht, China, COCOMO81, Desharnais, Finnish, Kemmerer, Kitchenham, Maxwell, Miyazaki, NASA18, NASA93, Telecom. 2 details the number of observations used to apply the ML model, with some entries missing in the dataset, which are being ignored for correct results. Some other statistical details like mean, median effort, maximum and minimum efforts are shown in the table.

Table 1.1 provides a comparative study on the Albrecht dataset on applying the ML estimation model. The result shows, MLP gives higher pred accuracy, higher R-square value, minimum MAE then RF and NB. So from above analysis we can say, MLP is better ML algorithm when implemented with Albrecht table 1.2.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, September 2023

Table 1.1: Comparative analysis of performance of ML algorithms on ALBRECHT dataset.

Models	Pred (25)	Pred (50)	MAE	MMRE	MMER	MdMRE	R Square	MSE	RMSE
SVM	37.50%	50%	0.0461	0.7629	-1.5324	0.4703	0.8852	0.0040	0.0631
RF	62.50%	62.50%	0.0425	0.7119	0.2155	0.1570	0.9202	0.0028	0.0526
DT	25%	50%	0.1032	0.6764	0.2780	0.1989	0.1877	0.0282	0.1679
SBG	37.50%	37.50%	0.0667	1.7262	0.3734	0.4173	0.8168	0.0064	0.0798
NB	50%	62.50%	0.0464	1.0948	-2.9651	0.4302	0.9072	0.0032	0.0567
MLP	75%	75%	0.0405	0.8617	0.2728	0.1870	0.9284	0.0025	0.0498
LR	37.50%	37.50%	0.0632	1.4789	-0.1823	0.5429	0.8598	0.0049	0.0698
KNN	37.50%	50%	0.0958	0.7200	0.3871	0.2751	0.2892	0.0247	0.1571

Table 1.2 shows a comparison of applying ML models to the China dataset. As can be seen from the results, KNN has the highest prediction accuracy followed by MLP. When it comes to comparing R-squared values, RF comes out ahead, and it has the lowest absolute error. So from the above analysis, it can be concluded that RF performs better with two metrics.

Table 1.2: Comparative analysis of the performance of ML algorithms on the China dataset.

Models	Pred (25)	Pred (50)	MAE	MMRE	MMER	MdMRE	R Square	MSE	RMSE
SVM	20%	41.33%	0.0441	1.4281	0.1680	0.6108	0.6914	0.0056	0.0748
RF	10.67%	25.33%	0.0204	0.1367	0.1033	0.0531	0.7453	0.0046	0.0679
DT	22.66%	46.66%	0.0366	1.0713	0.5456	0.5011	0.6409	0.0065	0.0807
SGB	23.33%	46%	0.0436	1.9753	0.4937	0.5501	0.5297	0.0085	0.0923
NB	21.33%	47.33%	0.0415	0.9399	0.3359	0.4545	0.6239	0.0068	0.0826
MLP	27.33%	49.33%	0.0357	0.9481	0.3734	0.5025	0.7015	0.0054	0.0735
LR	26%	46.66%	0.0433	0.9087	0.2739	0.4753	0.4531	0.0099	0.0995
KNN	38.66%	70.66%	0.0332	0.3484	0.3264	0.2142	0.6541	0.0063	0.0792

Table 1.3 provides a comparative study on telecom datasets on applying ML estimation models. As the above result shows, NB and LR gives the highest prediction accuracy, highest R-squared value, lowest absolute error, followed by RF. So we can say that both NB and LR models perform best with telecom dataset.

Table 1.3: Comparative analysis of performance of ML algorithms on telecom dataset.

Models	Pred (25)	Pred (50)	MAE	MMRE	MMER	MdMRE	R Square	MSE	RMSE
SVM	16.66 %	33.33 %	0.0570	0.5530	0.2638	0.2353	0.5933	0.0047	0.0689
RF	33.33 %	33.33 %	0.0676	0.7271	0.2233	0.1334	0.4512	0.0064	0.0800
DT	33.33 %	33.33 %	0.0735	0.5899	0.2043	0.0811	0.2811	0.0084	0.0916
SGB	16.66 %	16.66 %	0.1218	1.3561	0.3350	0.3267	-0.5649	0.0182	0.1351
NB	100 %	100 %	0.0016	0.0174	0.0186	0.0055	0.9997	0.0000	0.0017
MLP	16.66 %	16.66 %	0.1904	1.9030	0.4637	0.8072	-3.1509	0.0484	0.2200

Models	Pred (25)	Pred (50)	MAE	MMRE	MMER	MdMRE	R Square	MSE	RMSE
LR	100 %	100 %	0.0016	0.0174	0.0186	0.0055	0.9997	0.0000	0.0017
KNN	16.66 %	33.33 %	0.0974	0.7965	0.3255	0.2324	-0.0452	0.0122	0.1104

There may be many valid arguments to support the appropriateness of models. Relative error measures such as MRE, MMRE and prediction accuracy are measures that are independent of the output value. While estimating the effort of small and large systems with competing relative error is more logical, the concept of relative error appears to be obvious to software academics and practitioners [8]. The use of PRED, which indicates that a higher PRED model has a higher proportion of accuracy, also meets this requirement. When using MMRE to choose between competing models, models with lower MMRE values are better. A low MMRE is taken to indicate minimal uncertainty or inaccuracy and is also used to provide a quantitative assessment of the uncertainty of a prediction [8]. Therefore further statistical analysis is based on the comparison of MMRE value obtained by different ML models and existing work.

Table 1.3 gives a statistical analysis based on the performance of existing estimation models and the proposed model when considering MMRE. In the papers taken into consideration various inference techniques are used such as genetic programming, particle swarm optimization, ensemble learning, deep learning and many others. According to the results, a lower MMRE value is preferred, which indicates minimal uncertainty. In most cases, machine learning models give lower MMRE values except in the case of the Albrecht dataset, which gives lower MMRE values when estimated using analogs. The results obtained show that the machine learning model measures generally reduce the relative error compared to existing models in almost all cases except Albrecht.

On considering prediction accuracy to compare algorithms on different datasets. Albrecht dataset predicts effort accurately with MLP algorithms, COCOMO81, China, Desharnais, Kemmerer and Miyazaki predicts effort accurately with KNNs. The Finnish and Maxwell prediction accuracy is higher when the random forest is applied. The datasets Kitchenham and NASA93 give higher prediction accuracy with DT and stochastic gradient boosting algorithms, respectively. Both NASA18 and Telecom, being two small datasets, provide 100% prediction accuracy with NB and LR.

III. CONCLUSION

Comparative analysis of effort estimation in machine learning techniques involves assessing and comparing different methods and tools used to estimate the time, resources, and complexity involved in developing a machine learning model.

Effort estimation in machine learning is essential for project planning and resource allocation.

A combination of expert judgment, historical data analysis, and automated tools can provide more accurate and reliable estimates.

Continuous monitoring and adjustment of estimates throughout the project lifecycle is critical to success.To conclude, we believe that our analysis here has highlighted the consistency achieved by Random Forest in almost every case. Specifically, the metrics we used to compare, out of 36 cases for each metric, considering three cases for each dataset, MMRE, MMER, MDMRE; Random forest turnout is more stable after decision tree. On considering the prediction accuracy; Random Forest is more stable and provides accurate results after k nearest neighbors. In the case of R-Square; Naive Bayes, MLP appears to be more accurate, followed by Linear Regression and Random Forest. Overall we can specify that different algorithms have different approaches and nature, which vary with the results obtained.

IV. FUTURE DIRECTIONS

Future research could focus on improving automated estimating tools, incorporating more advanced machine learning techniques for estimation, and developing standardized practices for the industry.

In summary, estimating effort in machine learning projects is a complex and important task that requires a combination of methods and tools. Each approach has its own advantages and limitations, and the choice of method should be tailored to the needs of the specific project and available data. Automation and data-driven estimation methods are likely to play an important role in improving accuracy and consistency in the future.

REFERENCES

[1]. Shweta.KR etal. (2022),” IMPROVED ARTIFICIAL NEURAL NETWORK MODEL FOR SOFTWARE PROJECT COST ESTIMATION IN EARLIER STAGE”, International Journal of Mechanical Engineering, ISSN: 0974-5823 Vol. 7 No. 1 January, 2022.

[2]. Gouthaman P, Suresh Sankaranarayanan, Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers”, computers and electrical engineering (2021)

[3]. DareenRyied Al-Tawal, MazenArafah. GhalebJalilSweis, “A model utilizing the artificial neural network in cost estimation of construction projects in Jordan”, ECAM (2020)

[4]. Fangwei Ning, Yan Shia, MaolinCai, Weiqing Xu, Xianzhi Zhang, “Manufacturing cost estimation based on the machining process and deep learning method”, journal of manufacturing systems (2020)

[5]. Erik Matel, FaridaddinVahdatikhaki, SiavashHosseinyalamdary, Thijs Evers,HansVoordijk, “An artificial neural network approach for cost estimation of engineering services”, International journal of construction management (2019)

[6]. Mahmood Mohd Al Asheeri, Mustafa Hammad, “Machine Learning Models for Software Cost Estimation” International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) IEEE (2019)

[7]. Michal Juszczyk ,AgnieszkaLesniak, Krzysztof Zima, “ANN Based Approach for Estimation of Construction Costs of Sports Fields”, wiley (2018)

[8]. Przemys_awPospieszny ,BeataCzarnacka-Chrobot , Andrzej Kobylin_ski, “An effective approach for software project effort and duration estimation with machine learning algorithms,The Journal of Systems & Software (2017)

[9]. T.M.S. Elhag, A.H. Boussabaine, “An Artificial Neural System for Cost Estimation of Construction Projects”, In: Hughes, W (Ed.), 14th Annual ARCOM Conference, 9-11 September Vol. 1, 219-26. (1998)

[10]. Richa Yadav, Monica Vyas, Vivekanada Vyas, Sanket Agrawal, “Cost Estimation Model (Cem) for Residential Building using Artificial Neural Network”, (IJERT) Vol.5(1) (2016)

[11]. H. Murat Gunaydin, S. ZeynepDogan, “A neural network approach for early cost estimation of structural systems of buildings”, International journal of project management (2004)

[12]. Lawrence H. Putnam. A general empirical solution to the macro software sizing and estimating problem. IEEE transactions on Software Engineering, 4(4):345, 1978.

[13]. Krishnakumar Pillai and VS Sukumaran Nair. A model for software development effort and cost estimation. Software Engineering, IEEE Transactions on, 23(8):485–497, 1997.

[14]. Allan J Albrecht. Measuring application development productivity. In Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, volume 10, pages 83–92, 1979.

[15]. IFPUG IFPUG. Function point counting practices manual, release 4.2. International Function Point Users Group, USA–IFPUG, Mequon, Wisconsin, USA, 2004.

[16]. Barry W Boehm et al. Software engineering economics, volume 197. Prentice-hall Englewood Cliffs (NJ), 1981.

[17]. Robert T Hughes. Expert judgement as an estimating method. Information and Software Technology, 38(2):67–75, 1996.

[18]. Norman Dalkey and Olaf Helmer. An experimental application of the delphi method to the use of experts. Management science, 9(3):458–467, 1963.

[19]. Magne Jørgensen. Forecasting of software development work effort: Evidence on expert judgement and formal models. International Journal of Forecasting, 23(3):449–462, 2007.

[20]. Martin Shepperd, Chris Schofield, and Barbara Kitchenham. Effort estimation using analogy. In Proceedings of the 18th international conference on Software engineering, pages 170–178. IEEE Computer Society, 1996.

[21]. Fiona Walkerden and Ross Jeffery. An empirical study of analogy-based software effort estimation. Empirical software engineering, 4(2):135–158, 1999.

[22]. Ali Idri, Fatima azzahra Amazal, and Alain Abran. Analogy-based software development effort estimation: A systematic mapping and review. Information and Software Technology, 58:206–230, 2015.

[23]. Adriano LI Oliveira. Estimation of software project effort with support vector regression. Neurocomputing, 69(13):1749–1753, 2006.