A Hybrid Machine Learning Method for Estimating Software Project Cost

Article *in* International Journal of Innovative Science and Research Technology · March 2023

CITATIONS
0

READS
106

4 authors:

Beatrice Akumba
Benue State University
13 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Emmanuel Ogala
Joseph Sarwuan Tarka University
58 PUBLICATIONS **47** CITATIONS
SEE PROFILE

Iorshase Agaji
Federal University of Agriculture Makurdi, Nigeria
23 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Nachamada Vachaku Blamah
University of Jos
39 PUBLICATIONS **199** CITATIONS
SEE PROFILE

ISSN No:-2456-2165

A Hybrid Machine Learning Method for Estimating Software Project Cost

Beatrice O. Akumba.,[1] Emmanuel Ogala[2]Iorshase Agaji.,[2] Nachamada Blamah[3]
[1]Department of Mathematics/Computer Science, Benue State University Makurdi, Nigeria
[2]Department of Mathematics/Statistics/Computer Science, Joseph Sarwuan Tarka University Makurdi, Nigeria
[3]Department of Computer Science, University of Jos, Nigeria

Abstract:- This paper introduced the concept of a hybrid machine learning method for estimating software project cost. The literature review of some of the models commonly used in the software project cost estimation was carried out. A hybrid method of algorithms comprising Random Forest (RF), Kalman Filter (KF) and Support Vector Machine (SVM) algorithms respectively were proposed to predict the software project cost and its completion time for software projects. The proposed architecture of the model was presented as well as the proposed the model.

Keywords:- *Software Cost Estimation, Machine Learning, Cost Estimation Models, Kalman Filter Algorithm, Support Vector Machine, Random Forest.*

I. INTRODUCTION

Software project cost estimation (SCE) entails forecasting the entire cost required to undertake a software project system (Shukla, *et al*., [1]). The process is significant because it is key to estimating the software project's costs at the initial phase (Fadhil *et al.,* [2]). Project cost estimations are important for computing resources and the budget required for a software project. The estimation of resources and schedule can be achieved through accurate estimation and forecasting. It is among the pivotal task in software project development that entails forecasting the software cost, software size and effort required to obtain a software project system. (Chirra and Reza, [3]).

Software cost estimation requires the determination of one or more of the following estimates; effort (usually in person-months), project duration (in calendar time) and cost (in money). Most of the software project cost estimation models try to generate an effort estimate, which can then be converted into the project duration and cost. Although, effort and cost are closely related, they are not necessarily related by an easy changeover or conversion function. The software effort is usually calculated as the person-months spent by the programmers, analysts and project managers on the actualization of the software project. The estimation of the software effort can be represented in Naira or dollars by computing the average salary divided by the time spent by the staff involved and further multiplied by the required estimated software effort.

A lot of software cost estimation frameworks have been established to enhance the estimation in the cost of software projects. The main categories of these models are algorithmic, non-algorithmic and learning oriented models. The algorithmic models make use of equation and mathematics to carry out the estimation process. The software projects' Source Lines Of Code (SLOC), function points, the software cost drivers such as programming languages, risk assessments, design methodology and so on are mostly used as inputs to the model. Some examples of algorithmic models include COCOMO, Functional Point Analysis (FPA), and Putnam model among others. The non- algorithmic models on the other hand which are known as non- parametric models are estimation processes based on deduction and analogy. They require the knowledge of previously completed software projects that are similar to the current software project for its estimations. Some examples include; expert judgment, Analogy based, Price to Win, Top-Down and Bottom-up models respectively. The learning-oriented models otherwise called the Machine learning methods include the Artificial Neural Networks (ANN), Fuzzy logic (FL), Analogy based, Bayesian Networks, Regression Tree, Genetic Algorithm (GA), Support Vector Machines (SVM), Case-Based Reasoning (CBR) and Kalman Filter Forecasting Methods (KFFM) [3].

The models have their merits and demerits and therefore are not the same with respect to their performance, such as inadequacy and inefficiency of the mathematical models to explain the nonlinear properties existing between the input and output parameters of the software project cost estimation datasets. The shortcomings of algorithmic models led to the consideration of non-algorithmic models and intelligent techniques which are soft computing based (Resmi *et al.,* [4]). All the aforementioned software cost estimation techniques/models are faced with the problem of the absence of accuracy [2].

Most of the existing software project cost estimation models are faced with challenges and there are two substantial reasons behind the software project failure. They are inappropriate project estimation in terms of size, costs and personnel required and uncertainty about the software and system requirements due to advancement in technology. With early and accurate estimate of project costs, client expectations can be set and ensure that funds are made available to achieve the project. There is need to have an accurate estimate of the worth of a software project before embarking on it as it is critical to the timely delivery and subsequently return on investment. Therefore, this paper seeks to propose a hybrid machine learning method for estimating software project cost and to propose the use of non-algorithmic learning oriented technique of Kalman Filter Forecasting Algorithm (KFA) to forecast the completion time of software projects.

ISSN No:-2456-2165

This paper is organized as follows: Section I is Introduction, Section. II outlines the objectives of the paper. Section III presents the literature reviewed. Section IV explains the Methodology while Section V is Conclusions and Future Work.

II. OBJECTIVES

The specific objectives to be focused on this paper are:

- Carry out a literature review on machine learning techniques of software project cost estimation
- Develop an architecture of the framework of the hybrid software project cost estimation model.
- Proposed a hybrid cost estimation model based on machine learning that will improve on the cost accuracy.

III. LITERATURE REVIEW

This section reviews relevant literatures with respect to the concept of software project cost estimation; software project cost estimation methods/techniques and SCE based on machine learning methods.

A. THE CONCEPT OF SOFTWARE PROJECT COST ESTIMATION

According to Fashina [5], software project cost estimation is a process that estimates/predicts the estimated cost of the software project before the commencement of the development. It explains the approximate requirements of effort, development time and resources that are required to complete the software project. It is among the essential processes to start development for software by taking into consideration all internal and external cost factors. The cost estimation is a tool that estimates the planning, budgeting and resource utilisation for the software projects. Before cost estimation for a software project is done, the actual requirements for the project, the complexity of the requirements, and the software cost driver factors that affect the development process (like, product factor, project factor, personal factor and hardware factor) must be considered. They act as the inputs to the cost estimation process. The output of the estimation process is Effort, Development Duration, and Resources. Figure 1 illustrates the software project cost estimation process diagrammatically showing its inputs and outputs.

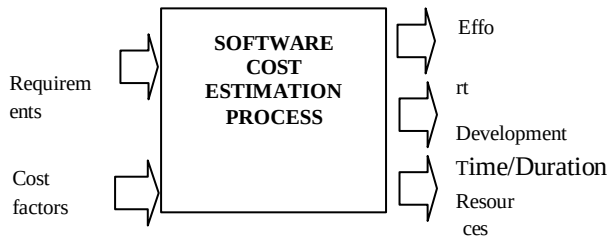

Fig. 1: The Software Cost Estimation Process (Source: [5])

Shekhar *et al.,* [6] explained the various models and techniques used in determining the cost of the software. The study tried to justify if there could be a best software cost estimation method for software projects. An inference was made that there is as such not any single method that can be cost. However, it suggested that efforts should be made to use a combination of the estimation techniques (hybrid) in order to arrive at a better software project cost and quality estimate. Thus, to achieve reliable estimates, it is required to have proper knowledge and understanding of each technique and the relationship that exist between the software attributes respectively. A combination of the methods would greatly enhance accurate cost estimation of software projects and would be seen to have addressed one of the complexes and most challenging activity for the software companies Iqbal *et al.,* [7] investigated the pros and cons of different software cost estimation modeling techniques and concluded that one cost estimation method should not be stuck on. That for cost and timeframe estimations, different types of estimation methods should be used (A Hybrid model), comparison of the results and analysis of the variations should be explained, a documentation of the prediction made when making the cost estimates. Monitoring and evaluation of the entire project accuracy in software cost estimate process should be made. Also suggested was to maintain all previous/past data of software projects. The study concluded that it was very difficult to recommend a fail-safe cost estimation method as most of the cost estimations techniques are scenario dependent. In order to select a method that suits any given project, a proper analysis of the project is must be carried out. This is because a wrong estimation technique can significantly delay a software project's delivery, and a right technique can make a software project breeze through its deadlines. In addition, a detailed analysis of some global factors must be made; otherwise, it becomes very difficult to give accurate deadlines. Also, an emphasis on smaller details is necessary, as they can cause delays when they add up together.

B. SOFTWARE COST ESTIMATION METHODS

According to Iqbal *et al.,* [7], the software project cost estimation is broadly classified into three methods into which these models are categorized and they are; Algorithmic methods, Non-algorithmic methods and the Machine Learning methods.

Chirra and Reza [2] stated that the algorithmic method was developed to deliver a few equations mathematically that can be used to carry out software cost estimation.

The mathematical equations were based on historical data and research and made use of Source Lines of Code

(SLOC), number of function calls to execute, cost drivers like programming language, design methodology, skill- levels and risk assessment and so on as inputs to the model. The algorithmic methods comprised the COCOMO, Function Point Analysis (FPA), Putnam model/SLIM respectively. The non-algorithmic models on the other hand

regarded as the best method for estimating software project

also known as non- parametric models make their estimation process based on deduction and analogy. They require the knowledge of previously completed software projects that are similar to the current software project for their estimation process. Examples include; expert judgment, Analogy based, Price to Win, Top-Down and Bottom-up models respectively.

According to Aljohani and Qureshi, [8], the Non-Algorithmic models are sub-classified into Learning based models, Expert based Models and Dynamics Based models. Examples of learning based methods are Artificial Neural Networks, Genetic Algorithms, Fuzzy Logic, Bayesian Networks, Support Vector Regression, Regression Tree, Analogy and so on. The Expert based Model comprises Top-Down estimation, Bottom-Up estimation, Expert Judgment and Price-to-Win estimation methods. The Dynamics based models are techniques that are used to approximate the changes in effort in the course of software development of a system. The model is built such that it predicts changes in the cost; the staffing team needs, schedules and the time of project as far as the main initial values of the project are made available to estimate.

C. SCE BASED ON MACHINE LEARNING METHODS

Pospieszny *et al.*, [9] predicted the continuous cost and duration target of variable of software projects using machine learning algorithms. Three effective machine learning predictive algorithms of Support Vector Machine (SVM), Multi-layer Perceptron Artificial Neural Network (MLP-ANN) and Generalised Linear Models (GLM) were used. This was achieved by using an International Software Benchmarking Standards Group (ISBG) dataset after being preprocessed through smart data preparation. The input dataset was used for modeling by the application of three machine learning algorithms (SVM, MLP, GLM) ensemble averaging and cross validation. The obtained model results for cost and duration estimation served as decision support system for organizations that are into software development and implementation systems.

Resmi *et al*., [4] deployed an innovative process of Fuzzy analogy algorithm connected through the firefly algorithm to estimate the effort of software projects. The input dataset used was from the Promise data repository comprising four datasets of NASA93, NASA60, COCOMO81 and Deshnaris datasets respectively. A combination of the fuzzy analogy by the side of the firefly and the Expectation-Maximisation (EM) process for the estimation of software project enhanced the accuracy level of the prediction. The fusion of fuzzy and firefly algorithm on clustered datasets improved the accuracy of the estimation to a considerable level.

Arora and Mishra [10] proposed an estimation model for software cost evaluation using Artificial Neural Network (ANN) Multilayer Feed forward Neural Network using perception learning algorithm and COCOMO dataset. The experiment was performed on 14 software projects with the actual effort of the model compared with the experimental effort values. The result indicated that accurate values were attained by the model.

BaniMustafa [11] investigated the validity of machine learning techniques to predict software effort estimation by proposing an alternative method of COCOMO. This was achieved by performing data mining on historical data to obtain better software estimation. The dataset was preprocessed in order to calculate the effort using the COCOMO model equation. The estimated effort was then compared with both the actual effort and the predicted effort by applying Naïve Bayes, Logistic Regression and Random Forest machine learning algorithms. The dataset used was surveyed by using the density function and projection plots to check both the nature and potential of the dataset after which a number of models were built by applying the selected machine learning techniques. The built Naïve Bayes, Logistic Regression and Random Forests models generated were further tested using five folds cross- validation and the models performance was evaluated with Area Under the Curve (AUC), CA, Precision and Recall, Confusion Matrix and ROC. The results of the investigation confirmed the validity of data mining as an alternative for COCOMO. Although, it was observed that no hybrid of the ML techniques were done and recommended that more ML techniques should be adopted and used as well as extending the research to cover more machine learning techniques and a hybrid of them if possible.

Khazaiepoor *et al*., [12] combined different Machine Learning (ML) methods during several phases to provide a software development effort estimation model that irrespective of its high efficiency in estimating software was totally independent on the type of dataset used. Six datasets were used to test the models independence in effort estimation. The developed model comprised three phases; phase one combined the genetic algorithm and neural network to perform the feature selection operation. The second phase employed the genetic algorithm and imperial competitive algorithm (ICA) to perform a clustering operation. The third phase applied neural network to perform modeling and testing in clusters. Evaluation metrics of MMRE, MDMRE and PRED(0.25) were applied to know the accuracy of the model and the learning rate was compared with those of other models and the model was seen to have outperformed all the other methods for all the datasets used.

Abdulmajeed *et al*., [13] adopted machine learning (ML) techniques in solving the problem of predicting the cost of the software project as it tries to give an estimate as close as possible to the real cost of the project. The study was based on the completed projects in the NASA (National Aeronautics and Space Administration) data set, which contained the attributes of 60 software projects. It also tried to find and define a suitable method to give the closest possible estimate to the true cost. The K-Nearest Neighbors KNN algorithm was used and applied to the NASA dataset, and its results were compared with both Cascade Neural Network and Elman Neural Network. The results performance was assessed and compared with the aid of performance standards such as; MMRE, RMSE, and BRE. The results of the K-Nearest Neighbors (KNN) algorithm indicated good accuracy levels in the course of forecasting the required cost of developing software projects. The CNN algorithm and the ENN network gave less accuracy levels in software project cost estimation.

Zakaria *et al*., [14] used several algorithms of machine learning to estimate the effort of software project development. An application known as SOFREST estimator was invented to explain how the software estimation process

ISSN No:-2456-2165

works and to ascertain the inputs required to yield the result/output. The system required that the user inserted a software project's five inputs which are; the Lines of Code (LOC) number, Size of Database (DATA), needed Software Reliability (RELY), the Execution Time Constraint (TIME) and Main Storage Constraint (STOR). The system's output was the estimation of the effort that was required by the software project in person-months. Four Machine Learning algorithms were used and they 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).

Karimi and Gandomani [15] modeled the effort estimation needed for software development by using a hybrid of Fuzzy Neural Network (FNN) and Differential Evolution (DE) algorithm. The model obtained a better and accurate estimation of the software projects' effort that was a better off as compared to previous methods. The model was called ANFIS-DE and had two phases. The first phase was the training phase and was applied to obtain the optimal parameter for ANFIS using the DE algorithm and the second phase known as the test phase used the obtained optimal parameters to test and assess the model. The DolBat algorithm had a better performance in all the assessed criteria than the other algorithms in experimental functions. MMRE and PRED were used as performance parameters to assess the model. The accuracy was seen to be better as compared to other optimization algorithms adopted from genetic, evolutionary and meta-heuristics respectively.

Dashti *et al.*, [16] proposed a weight optimization technique based on the learnable evolution model (LEM) in analogy-based estimation (ABE) for software projects. The work employed three control parameters of similarity function, K-nearest neighbor, and solution function suggested by the LEM algorithm to find the optimal weights which were parsed into the model. The amount was estimated as the proposed cost for the project. The proposed cost of the project was compared to the real cost of the project and its magnitude of relative error (MRE) metric was calculated. Euclidean and Manhattan distances were employed to obtain the similarity function and the most common solution functions of median and mean were used to obtain the proposed estimation value. Three datasets were used. These were Desharnais, Maxwell and International Software Benchmarking Standards Group (ISBSG). The datasets were used to evaluate the model and were tested in different states. The datasets were divided into training and testing datasets and used 10-fold cross-validation method to train the model. Three evaluation criteria, of MMRE, PRED (0.25), and MdMRE were used to evaluate and compare the model against other evolutionary algorithms like genetic algorithm, differential evolution, and particle swarm optimization and more desirable results were obtained. The results revealed that employing the model indicated considerable improvement in estimating software cost estimation.

Wani *et al.*, [17] proposed an analogy centered model that was based on differential evolution (DE) exploration process for software cost estimation to help the industry practitioners. The model was evaluated on 676 software projects from five different data sets (Kemerer, ISBSG10, Miyazaki, Desharnais and China). MRE and MAE were used to evaluate the model. The model results achieved were notably better as compared to other benchmarked analogy-based estimations. The result of the experiment indicated that the approach particularly DE was able to find the best possible estimators rapidly and out-performed the other widely used standard methods like base-line Analogy Based Estimation (ABE0) and automatically transformed linear model (ATLM).

Bushra and Kadam [18] developed an improved technique for software cost estimations in agile software development environment based on Support Vector Machine and Naïve Bayes algorithm. The system used COCOMO (Constructive Cost Model) to achieve better software project cost estimation. The Profit or loss estimation forecast for new software project was done with the help of some historical data of a software company. Machine learning techniques of Naïve Bayes and Support Vector Machine were used to perform the prediction using historical data. The SEERA dataset was used to evaluate the behavior of the model. The result indicated that the forecasted profit and loss prediction had an accuracy of 86.59% and 24.80% respectively. The accuracy of the software effort calculation was higher by 95.06% as compared to the SVM, 93.45%. It recommended a further study to increase the accuracy of the future prediction of the forecasted profit and loss of the system more than 90% by adopting the hybrid algorithm in the machine learning with the 97-98% overall effort calculation accuracy.

Zakaria *et al.*, [14] employed Support Vector Machine, Linear Regression, Regression Tree and Random Forest machine learning algorithms on COCOMO dataset to estimate software project cost. This was done by developing an application known as SOFREST Estimator. . Random Forest (RF), Linear Regression (LR), Regression Tree (RT) and Support Vector Machine (SVM) algorithms were applied on the COCOMO NASA 1, COCOMO NASA 2, COCOMO81 datasets respectively. The results of the experiments showed that Support Vector Machine and Random Forest algorithms impressively gave consistent results with the COCOMO datasets regardless of the number of effort attribute used. The experiment revealed that there was good performance and increase in the accuracy of Support Vector Machine for estimation of software project effort.

Nawaz *et al.*, [19] proposed a model to predict the effort of software cost estimation using genetic programming, neural networks and genetic algorithms with a distributed programming project dataset. Six types of classifiers were used to test the 13 algorithms. The classifiers used were Random Forest, REPTree, M5P, ZeroR, Decision Table, Input Mapped Classifier, Additive Regression, IBK, KStar, Gussian Processes, Linear Regression, Multilayer Perceptron, and SMOreg. The result

ISSN No:-2456-2165

indicated that Random Forest achieved the best results in the first experiment using the first procedure dataset while the three models of REPTree, Additive Regression and Kstar scored the best results using the second procedure dataset. ZeroR method had the worst results using the first dataset while some other methods including Multilayer Perceptron, IBk, and Linear Regression performed poorly on the second dataset.

Varshini *et al.*, [20] presented software effort estimation process using ensemble techniques and machine and deep-learning algorithms to solve the single model approaches and ensemble approaches that were considered for software cost estimations. This was achieved by the use of averaging, weighted averaging, bagging, boosting, and stacking ensemble techniques to create multiple models termed base-level classifiers which were combined to produce better predictions as compared to single-level models. The various stacking models considered and evaluated were stacking using a generalized linear model, stacking using decision tree, stacking using a support vector machine, and stacking using random forest. The results proved that the stacking with random forest gave the best results as compared with the single model approaches using the machine or deep learning algorithms and other ensemble techniques.

.

IV. METHODOLOGY

A Hybrid Machine Learning Method for Estimating Software Project Cost based on the commonly used machine learning algorithms is proposed in Figure 2. The proposed model is divided into the Calendar Time Prediction Phase using Kalman Filter Algorithm (A) and the Financial Cost Estimation Models using Random Forest and Support Vector Machine learning techniques (B). The implementation of this model is beyond the scope of this paper.

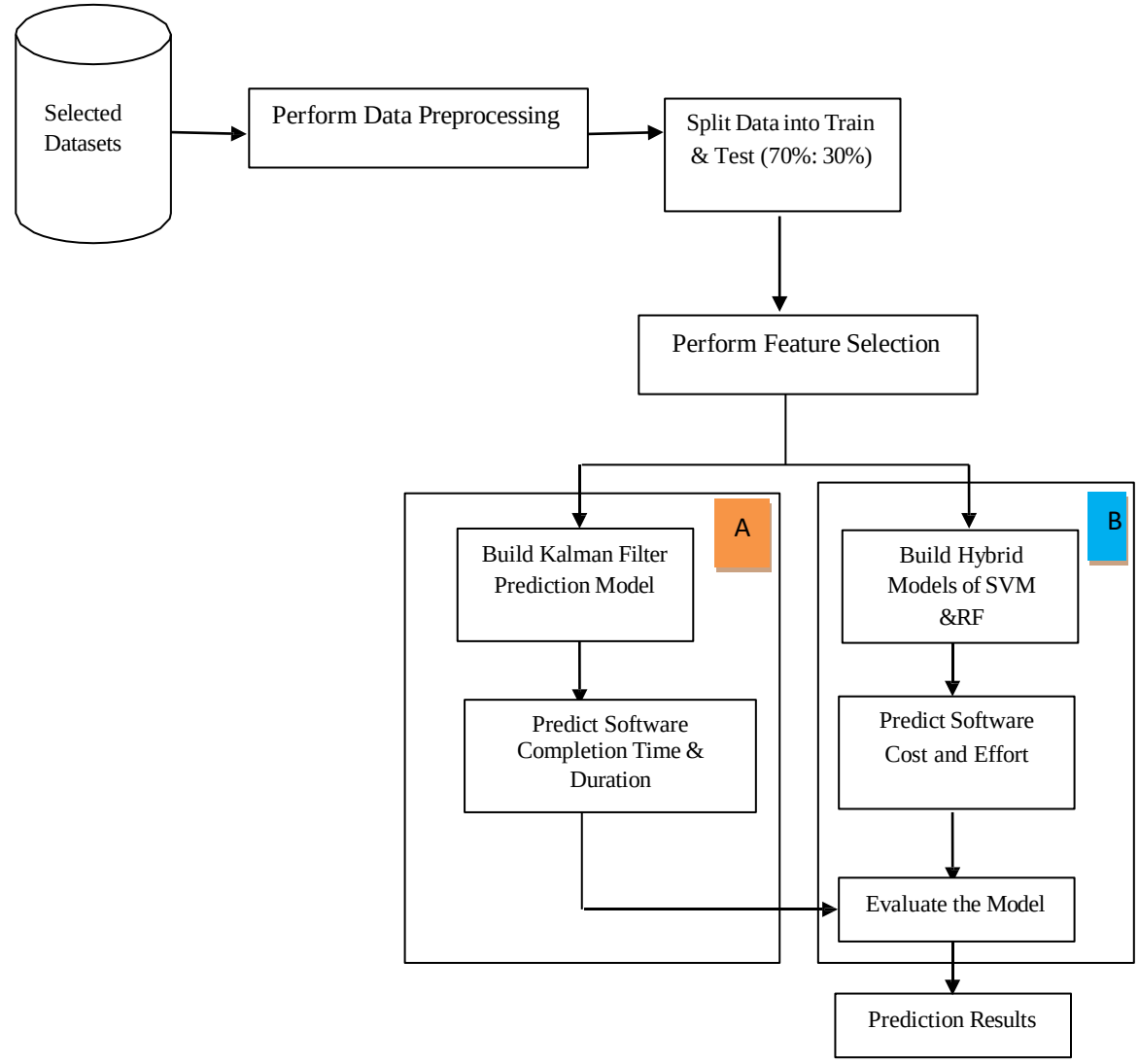

Fig. 2: Proposed framework for Hybrid Software Cost Estimation

V. CONCLUSION AND FUTURE WORK

This paper has made efforts to present an elaborate concept of software cost estimation. It reviewed software cost estimation techniques/methods and the different machine learning techniques used for predicting the software development cost in order to increase an understanding of this area of research. The paper went on to propose a hybrid software cost prediction based on the commonly used machine learning algorithms. The proposed hybrid model is divided into the Calendar Time Prediction Phase using Kalman Filter Algorithm (A) and the Financial Cost Estimation Models using Random Forest and Support Vector Machine learning techniques (B).

For future work, the implementation of this proposed model will be conducted along with the comparison of proposed model with existing techniques with benchmarked and new datasets. More advanced schemes such as ten-fold cross-validation and feature selection will be considered for the estimation of the prediction accuracy. In addition, an evaluation of the proposed hybrid model and their comparison with existing techniques will be conducted.

REFERENCES

[1.] H. K. Shukla, S. V. Singh and R. B. Singh. Cost Estimation of Software by ANFIS based Artificial Intelligence Approach. International Journal of Research and Development in Applied Science and Engineering 2021. (IJRDASE) Available online at: www.ijrdase.com

[2.] A. A., Fadhil, R. G. Alsarraj, and A. M. Altaie. Software Cost Estimation Based on Dolphin Algorithm. IEEE Acess Volume 8, 2020. Digital Object Identifier 10.1109/ACCESS.2020.2988867.

[3.] S. M. R. Chirra, and H. Reza. A Survey on Software Cost Estimation Techniques. Journal of Software Engineering and Application 2019, 12, 226-248. https://doi.org/10.4236/jsea.2019.126014.

[4.] V. Resmi, S. R. Vijayalakshmi and S. Chandrabose. An effective software project effort estimation system using optimal firefly algorithm. Cluster Computing https://doi.org/10.1007/s10586-017-1388-0. © Springer Science+Business Media, LLC, part of Springer Nature 2017.

[5.] A. Fashina. Challenges in the process of Estimating Software Cost . Jyvaskyla: University of Jyvaskyla 2021, 50 p. Information System Science, master's thesis.

[6.] S. Shekhar, and U. Kumar. Review of Various Software Cost Estimation Techniques. International Journal of Computer Applications 2016 (0975 – 8887) Volume 141 – No.11.

[7.] S. Z. Iqbal, M. Idrees, A. B. Sana, and N. Khan. Comparative Analysis of Common Software Cost Estimation Modelling Techniques. Mathematical Modelling and Applications. Vol. 2, No. 3, 2017, pp. 33-39. doi: 10.11648/j.mma.20170203.12

[8.] M. Aljohani and R. Qureshi. Comparative Study of Software Estimation Techniques. International Journal of Software Engineering & Applications (IJSEA) 2017, Vol.8, No.6, DOI: 10.5121/ijsea.2017.8603.

[9.] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobyliński. An Effective Approach for Software Project Effort and Duration Estimation with Machine Learning Algorithms, The Journal of Systems & Software 2017. doi: 10.1016/j.jss.2017.11.066.

[10.] S. Arora, and N. Mishra. Software Cost Estimation Using Artificial Neural Network. Soft Computing: Theories and Applications, Advances in Intelligent Systems and Computing 2018, 584, Springer Nature Singapore Pte Ltd. M. Pant et al. (eds.), https://doi.org/10.1007/978-981-10-5699-4_6

[11.] A. BaniMustafa. Predicting Software Effort Estimation Using Machine Learning Techniques. 8th International Conference on Computer Science and Information Technology (CSIT) 2018. 978-1-5386-4152-1/18/$31.00 ©2018 IEEE . ISBN: 978-1-5386-4152-1 . American University of Madaba Kings Highway, Madaba, Jordan, +96253294444, a.banimustafa@aum.edu.j

[12.] M. Khazaiepoor, A. K. Bardsiri and F. Keynia. A Dataset-Independent Model for Estimating Software Development Effort Using Soft Computing Techniques. Sciendo Journal of Applied Computer Systems. ISSN 2255-8691 (online) ISSN 2255-8683 (print) December 2019, vol. 24, no. 2, pp. 82–93 https://doi.org/10.2478/acss-2019-0011 https://content.sciendo.com

[13.] A. A. Abdulmajeed, M. A. Al-jawaherry, and T. M. Tawfeeq. Predict the Required Cost to Develop Software Engineering Projects by Using Machine Learning. Ashraf Journal of Physics 2021: Conference Series 1897 012029.

[14.] N. A. Zakaria, A. R. Ismail, A.Y. Ali, M. Khalid and N. Z. Abidin. Software Project Estimation with Machine Learning. International Journal of Advanced Computer Science and Applications, (IJACSA) 2021, Vol. 12, No. 6.

[15.] A. Karimi, and T. J. Gandomani. Software Development Effort Estimation Modeling using a Combination of Fuzzy-Neural Network and Differential Evolution Algorithm. International Journal of Electrical and Computer Engineering (IJECE) Vol. 11, No. 1, February 2021, pp. 707~715 ISSN: 2088-8708, DOI: 10.11591/ijece.v11i1.pp707-715. http://ijece.iaescore.com

[16.] M, Dashti, T. J. Gandomani, A. D. Hasanpoor, H. Zulzalil, A. B. M. Sultan. LEMABE: a novel framework to improve analogy-based software cost estimation using learnable evolution model 2022. PeerJ Comput. Sci. 7:e800 DOI 10.7717/ peerj-cs.800

[17.] Z. H. Wani, J. I. Bhat, and K. J. Giri.. A Generic Analogy-Centered Software Cost Estimation Based on Differential Evolution Exploration Process. Computational Intelligence, Machine Learning and Data Analytics 2021. The Computer Journal, Vol. 64 No. 3.

[18.] Q. Bushra,. and A. Kadam. An Improved Technique for Software Cost Estimations in Agile Software Development using Soft Computing Techniques. IT

(Information Technology) in Industry 2021, Vol. 9, No.2. ISSN (Online): 2203-1731

[19.] M. A. Nawaz, M. Kumar, M. D. Zaki, K. G. Sunil, R. Prithvi, K. Neeraj, and B. D. Raja. A Methodology for Software Cost Estimation Using Machine Learning Techniques. International conference on Recent Trends in Artificial Intelligence, IOT, Smart Cities & Applications (ICAISC-2020) (May 27, 2020)

[20.] P. A. G. Varshini, A. K. Kumari, and V. Varadarajan. Estimating Software Development Efforts Using a Random Forest-Based Stacked Ensemble Approach. Electronics 2021, 10, 1195. https://doi.org/10.3390/ electronics10101195.