

Early Size and Effort Estimation of Mobile Application Development

Ziema Mushtaq^{1,*}, Sami Alshmrany², Fahad Alturise³, and Tamim Alkhalifah³

¹Research Scholar, Maulana Azad National Urdu University, Hyderabad, India.

²Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia.

³Computer Department, College of Science and Arts in ArRass Qassim University, ArRass, Qassim, Saudi Arabia.

Abstract

With the increased complexity in mobile applications, many challenges and issues emerged for the software project management team to develop mobile application effectively and accurately. Effort estimation is one of the most critical issues the Software management project team faces in general, and the mobile application development team in specific. Effort estimation helps to approximate the cost required for successful software application development. The mobile application is distinct in various aspects from the traditional software and web-based applications. There is a need for a specific methodology to be followed for accurate estimation of size and efforts. This research aims to review the effectiveness of COSMIC and Machine Learning techniques in performing mobile effort estimation and propose a hybrid approach for efficient mobile effort estimation. This research work's mains represent the methodology followed to achieve the input parameters and mobile applications' efforts using a tailor-made approach. The significance of this research work is to propose a framework that will help both researchers and mobile application estimators approximate the efficient efforts precisely.

Keywords: Effort Estimation, Mobile application development, Mobile effort estimation methodology, Hybrid Machine learning techniques, CPEEM Approaches.

Received on 22 January 2021, accepted on 27 May 2021, published on 31 May 2021

Copyright © 2021 Ziema Mushtaq *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution, and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.31-5-2021.170010

*Corresponding author. E-mail: Sheikhz123@gmail.com

1. Introduction

The usability of mobile phones-based communication has increased rapid life. The increasing diversity and heterogeneity in mobile phone communication have redefined mobile and its various deliverability's fundamental principles. The range of applicability provided through mobile phones has increased the tradition of mobile-based software known as mobile applications. Most of the services and operations delivered through software applications or web-based applications are presented using mobile-based applications.

Thus, software companies try to maintain their reputation in the market by better planning and estimating the software. Software Estimation is defined as how much cost, time, efforts are required to build a specific product by time and within budget. Inaccuracy in estimation may lead to

overestimation or underestimation (over budgeting or under budgeting), which can be disastrous for any software company. As earlier, it was reported that the product might go 150% to 200% high if a proper estimation process is not followed [1]. According to World mobile application market estimates, more than 2 million mobile applications are currently available for mobile users, mainly in communication, games, social networks, multimedia, travels, and utility [2]. The ITU (International Telecommunication Union) estimates that there are more than 6 billion mobile clients worldwide. According to Gartner, 1.75 billion people own mobile phones with advanced capabilities [3]. According to a forecast by International Data Corporation's (IDC), a mobile application installed on client devices will increase to 210 billion annually from 2016 to 2020, adding more than 55 billion dollars to the total revenue [4]. Hence, there is a need for smart applications development with smartphones as the users demand smart functionality with time.

For software estimation, one should know the size of that software, which means size depends on the system's functionality provided to the user. There are internationally accepted standards for measuring the surface area of a wall, the volume of a bottle, weight of a person; likewise, we need to express standards for measuring the size of a software or application. There are two ways to measure the size of software or an application, i.e., Code Size: It measures Source Lines of Code (SLOC) in software and Functional Size: It is measured in functional points using a technique called Functional Size Measurement (FSM). FSM is used to measure the size of software by quantifying the functionality delivered to the user. FSM is used to overcome the limitations imposed by the Code Size technique having the following advantages:

- Based on Functional User Requirements.
- Independent of technology, Language, tools, and technique used.

In this paper, we propose a methodology to carry out this research work to get precise efforts. This paper discusses the proposed framework where mobile application development parameters that influence mobile applications' size are collected. This paper explains the pre-processing approach carried to derive the mobile application parameters called Input Factors. These factors are used as input parameters in the tailor-made approach to get the size and, hence, mobile application development efforts. The approach utilises the input factors with the proposed approach to estimate mobile application development's approximate efforts. Our proposed framework includes applying Machine Learning techniques and CPEEM to get the proposed model's performance in terms of the efforts calculated [5]. This research paper is organised in various segments, starting with the introduction followed up by a Literature review. Section 3 represents the methodology followed in this research work. Section 4 represents a pre-processing approach for collecting input parameters. Section 5 presents the proposed framework for building a robust approach for predicting accurate efforts for developing mobile applications.

2. Related Literature

Huang [6] used regression trees, case-based trees, and artificial neural networks for feature selection. Multiple data processing techniques were suggested. This study predicts the successful use of techniques mentioned above on used datasets. In this study, three cross-validations were used to calculate the accuracy in the prediction of efforts. It was suggested to use 10-fold cross-validation for better software effort estimation.

Li et al. [7] carried out a research study for feature selection-based model MICBR where feature selection algorithm and wrapper-based techniques were used. The results showed the space for improvement for MICBR. This study further suggested the usage of ISBUG datasets for evaluations of software effort estimation approaches.

Oteriveria [8], in this study author, proposed a genetic algorithm for software effort estimation. This study used three Machine Learning techniques, e.g., Model Trees, Multi-Layer Perception (MLP) neural networks and Support Vector Regression (SVR). This study showed promising results by using combined input of feature selection and optimised parameters of machine learning.

Attarzadeh [9] proposed a hybrid approach using a novel neural network and COCOMO model to estimate cost. This study used 63 projects for evaluation and suggested using a new neural network to check its accurate estimates.

Kumar [10] uses Wavelet Neural Network to predict accurate software efforts. In this study, three-fold cross-validation was used on datasets, but it was suggested to use ten-fold cross-validation to get precise results. The results were validated with the help of MMRE. This study suggested WNN somehow produced valuable results.

Shin and Goel [11] revealed a Neural Network method where Linear Regression analysis was used to estimate software. The proposed model was used on the NASA database, where several software projects were evaluated to check its efficiency

Burgees [12] used genetic programming and compared the results with Linear Square Regression, K-Nearest Neighbour (KNN) and MLP neural network and showed the results of genetic programming quite satisfactory to the rest of the three methods. The authors suggested using another method as genetic programming takes much time in experimentation.

Huang et al. [13] proposed a Neuro-fuzzy constructive cost model for effort estimation of software projects. The author compared the proposed method with the COCOMO approach and showed better results for the proposed approach.

Oliveria [14] used SVR (Support Vector Regression) for software effort estimation. In this study, Linear and RBF kernels were used for simulation with SVR. The author used software projects from the NASA database for experimentation. The study showed better results for SVR in comparison to the SG algorithm and Linear Regression.

Table 1. Tabular form of literature review with corresponding findings

Author	Technique used	Findings
Huang [6]	Regression Trees, Artificial Neural Networks, and case-based trees	In this study, the authors suggested Multiple data processing techniques and a 10-fold cross-validation technique to get the precise output.
Li et al. [7]	Feature selection techniques and wrapper-based techniques	This study suggested the usage of ISBUG datasets for the evaluation of methods.
Oteriveria [8]	Model Trees, Multi-Layer Perception neural networks, Support Vector	Promising results were found by using hybrid approaches

	Regression (SVR)	
Attazadeh [9]	Neural Network and COCOMO approach	This study suggests using a new novel neural network to get more accurate results.
Kumar [10]	Wavelet Neural Network	In this study, 10-fold cross-validation was suggested to get precise results.
Shin and Goel [11]	Neural Network	
Burgees [12]	Genetic programming, KNN, MLP and LSR	This study suggested using another method as genetic programming takes much time in experimentation.
Huang et al. [13]	Neuro-Fuzzy Constructive Cost Model	This study showed positive outcomes for the proposed approach.
Oliveria [14]	SVR, Linear and RBF kernel	This study showed better results for SVR used for software effort estimation.

After the literature review (Table 1), we observed that there is still room for precise cost estimation method development to get accurate results. Hence in this research, we will overcome the drawbacks of the previous research that are often ignored and still hope for improvement.

3. Research Methodology

A methodology is followed to derive a tailor-made approach to precise and accurate estimates in this research work. Below given figure 1 shows the pictorial representation of the methodology followed. This research is an empirical research type where the IFPUG dataset is used to obtain distinct features of mobile application projects like the name of the project, Actual size, Actual efforts, Techniques used etc. In this research work, 16 mobile applications were examined and used to check the proposed model's appropriateness. The step of activities starts with the requirement analysis, where mobile application requirements are elicited. The user selects these requirements, commonly termed as Function User Requirements (FUR). These FUR's are the functionality being delivered to the user through mobile applications. Once requirement elicitation is done, the activity follows towards the collection of data.

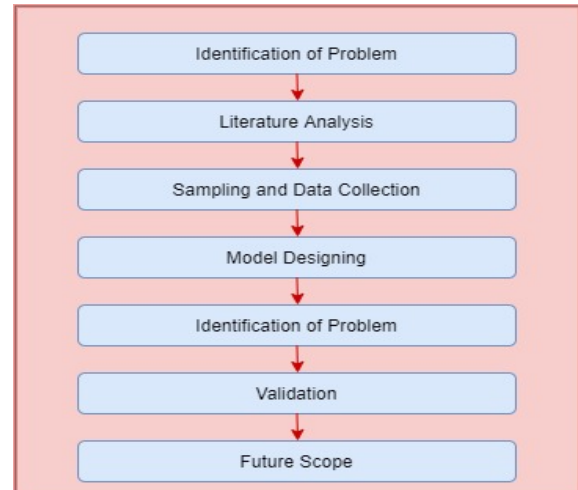


Figure 1. Stepwise actions of research methodology

Sample Data collection and organisation: Both the Primary and Secondary data is collected during this research work. For secondary data, 16 mobile applications of the IFPUG dataset are used for statistical evaluation. At the same time, primary data is collected in the form of a questionnaire from industry people having the same expertise. The primary data is filtered to collect the selected mobile application parameters for mobile application development. These input factors act as input parameters for the proposed approach, where, with the help of a tailor-made approach, the final output is calculated. The output is the overall efforts for various mobile application developments. Below given figure 2 show the categorisation of sample data collection. These obtained efforts for various mobile applications are validated through some statistical techniques, e.g., MRE, MMRE, Pred(x), and standard deviation. The validated result is again evaluated with industry people's help through a procedure that again involves stepwise activities.

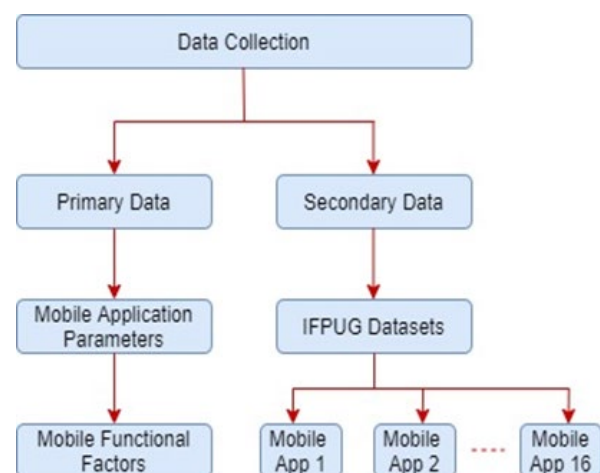


Figure 2. Categorisation of Data collection

4. Data Pre-Processing approach and Input Factor selection for Mobile application development

This step involves the collection of input factors of mobile applications. These input factors are the Mobile application parameters that are quite distinct from traditional software and web-based applications. These parameters influence the size of an application. Hence, they directly impact the accuracy in functionality delivered to the user through mobile applications, which depends on the customer’s requirements. This pre-processing approach is a set of activities influenced by previous research [15]. The author has used a tool “Questionnaire” to collect the Mobile application development parameters for mobile application development. Various parameters were analysed and listed as mobile application development parameters from previous studies [15]-[19]. These listed parameters were sent to different respondents having expertise in this field. A total number of 64 parameters were collected from previous research and framed in a questionnaire (Shown in Appendix-A). Each parameter has given five options with it. The scale used is a 5-Point rating scale where 5 distinct options were provided [20]-[25]. Respondents were asked to rate the particular parameter according to the given rating scale [26]-[28].

4.1.1 Input factor collection

Based on the respondents' feedback, 34 parameters were collected that directly impacted the size and efforts of Mobile application development. It was observed that there were duplicates of parameters that share the comparable interpretation. Therefore, these parameters were filtered to get the second list containing a non-ambiguous set of parameters where only 21 parameters were finalised. The approach (shown below, given in figure 3) was followed to achieve the parameters set.

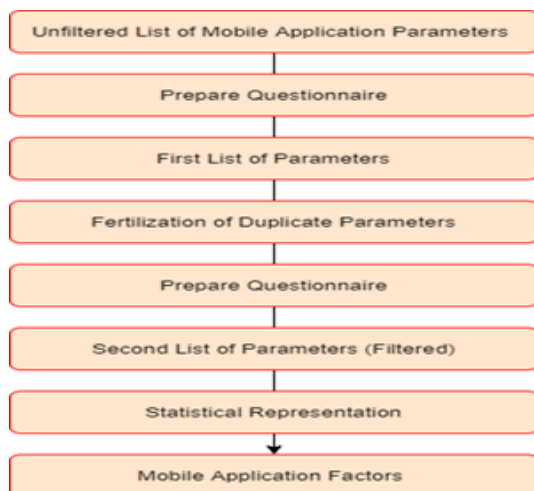


Figure 3. Stepwise activities for the selection of Mobile application factors

4.1.2 Mobile Functional Factors (MFF)

A list of 21 parameters was framed, which were analysed to check these specific parameters' appropriateness in line with the modern-day mobile application development. Therefore, a Questionnaire (Shown in Appendix A) listing 21 parameters were framed. A 5-point rating scale is used to collect different opinions from software industry experts to check which parameter is most suited nowadays and contributes to mobile application development. Table 2 shows the list of final parameters that constitute part of Mobile Function Factors (MFF) for modern Mobile application development. The respondents' responses are collected and are graphically shown in the figure below (see figures 4 to 10). As shown in table 3, six parameters are selected parameters out of 10 listed parameters based on the responses received. Responses received were based on the ranking scale values from 1-5, where 1 occupies the lowest ranking value, and 5 occupies the highest-ranking value on the scale, as shown in the table below (See Table 2). These parameters are here named Mobile Function Factors as they calculated the functionality of a mobile application delivered to the users.

Table 2. Different values for the 5-Point ranking scale

Value	Description
1	Less Impact
2	Unimportant
3	Standard
4	Significant
5	Powerful

Graphical representation of responses:

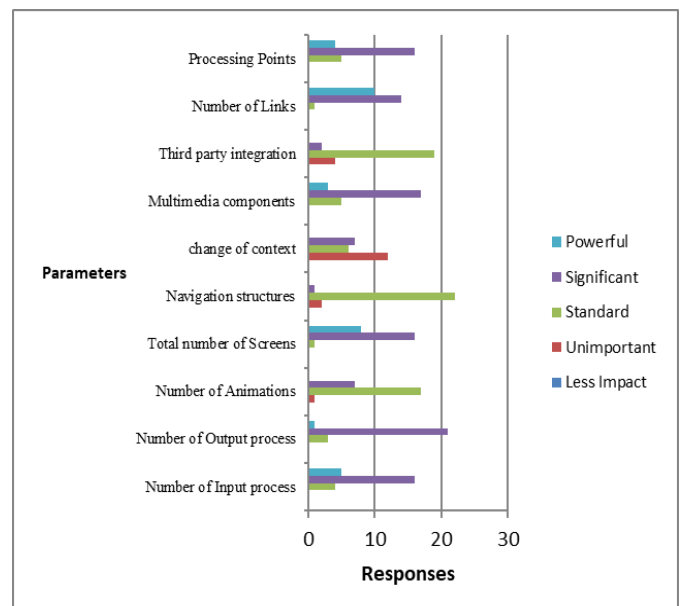


Figure 4. Graphical representation for the responses of different parameters from distinct respondents

Table 3. List of parameters that constitutes Input parameters.

Serial No	Parameters
1	Number of the Input process
2	Number of the Output process
3	Number of Animations
4	Total number of Screens
5	Navigation structures
6	change of context
7	Multimedia components
8	Third-party integration
9	Number of Links
10	Processing Points

To refine these ten parameters called Mobile Function Factors (MFF), as shown in the table above (see Table 3), a revision of these parameters was carried out. A new questionnaire was prepared, carrying 10 questions. Each question was scaled from 1 through 5, as shown in a table (see table 2). Out of these 10 parameters, 6 were received as such parameters having a huge impact on mobile application development functionality and were in line with the modern development of mobile applications. The responses are shown in the graphical representation in the below-mentioned figures (see Figure 5, 6, 7, 8, 9, 10), as shown in table 4.

Table 4. List of parameters that constitutes MFF as per the responses from respondents

Serial No.	Parameters
1	Number of the Input process
2	Number of the Output process
3	Total number of Screens
4	Multimedia components
5	Number of Links
6	Processing Points

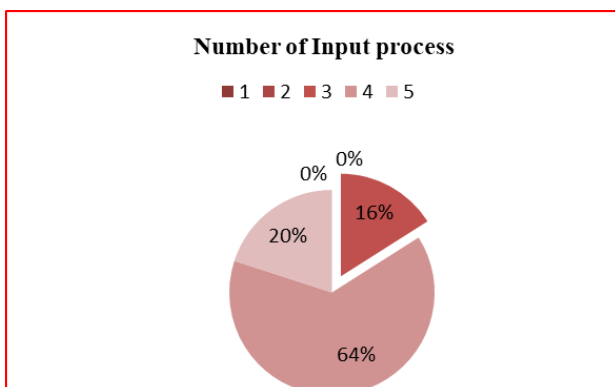


Figure 5. Responses for the parameter “Number of Input Process.”

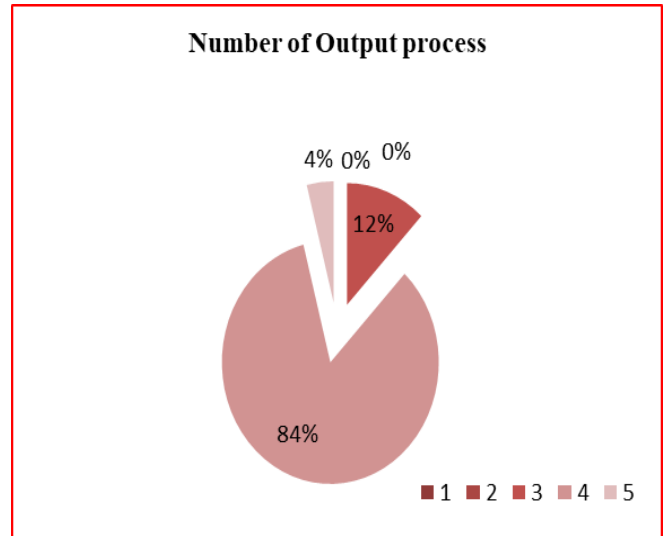


Figure 6. Responses for the parameter “Number of Output Process”

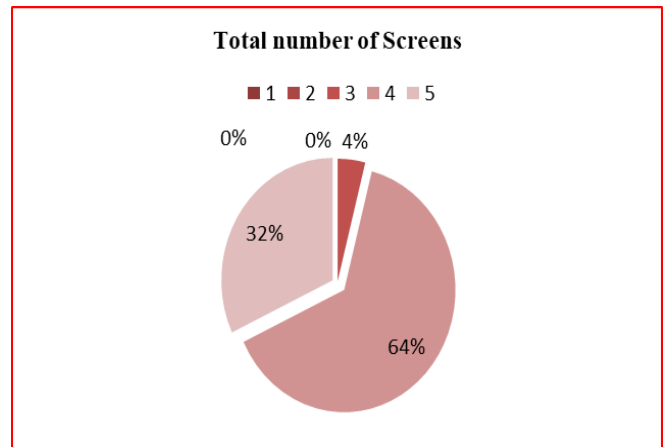


Figure 7. Responses for the parameter “Total number of screens”

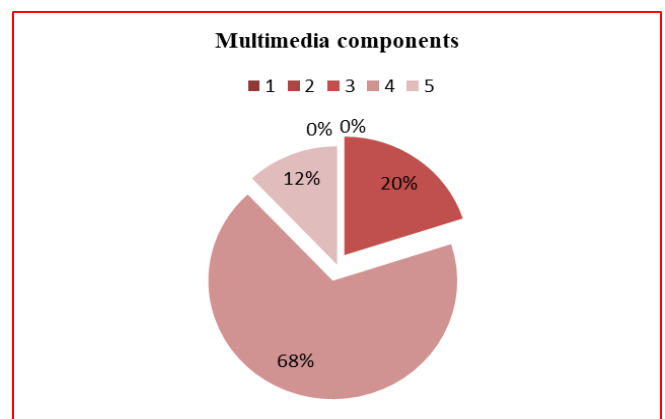


Figure 8. Responses for the parameter “Multimedia.”

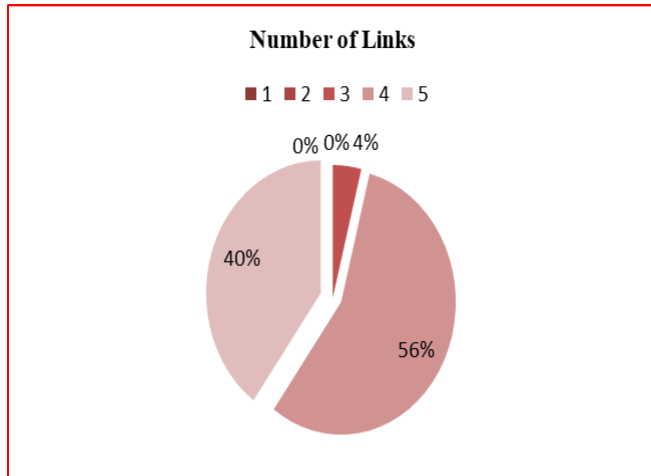


Figure 9. Responses for the parameter "Number of Links"

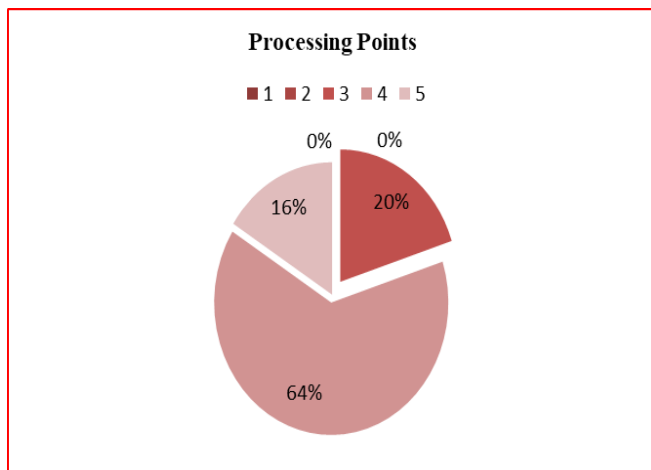


Figure 10. Responses for the parameter "Processing Points"

5. Proposed Framework

The proposed framework is based on machine learning technique and COSMIC Plus Effort Estimation Model (CPEEM) [5]. Where size is a primary means wherefrom efforts can be calculated for mobile application development. The size of a mobile application depends on the total number of functionalities an application can deliver to users. The functionalities of mobile applications depend on the user requirements calculated from the Mobile Function Factors. Mobile Function Factors calculates the size of a mobile application that can be used to get its overall efforts. The best suited Machine Learning technique in synchronisation with the CPEEM approach will be prepared so that the proposed tailor-made approach can provide efficient efforts for real mobile application projects. The diagram below (See Figure 11) gives the basic idea of a tailor-made approach where Mobile Function Factors (MFF) can be applied to calculate the overall efforts to carry out mobile applications.

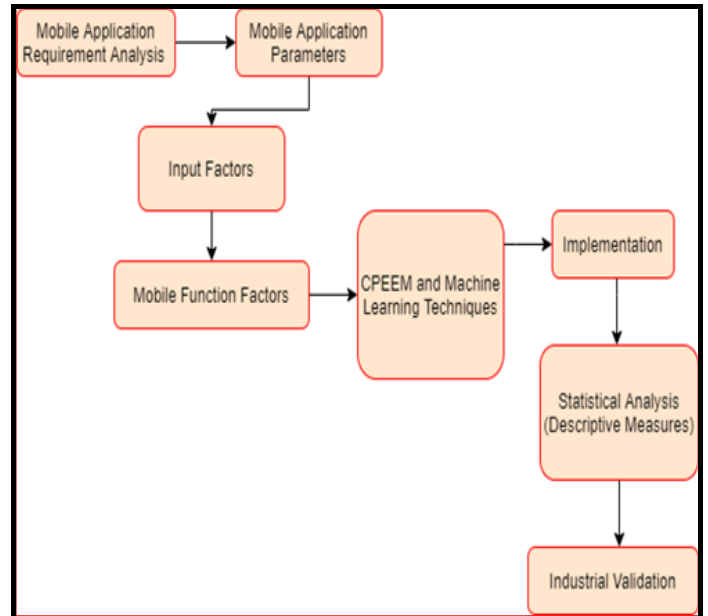


Figure 11. Stepwise measures for the proposed framework

The proposed framework is a hybrid approach that includes concepts of the COSMIC Plus Effort Estimation Model (CPEEM) and insights of Machine Learning Technique that can be used as a tailor to compute the general size efforts of mobile application development. The whole process involves certain steps that are shown in the figure above (see figure 11). The framework begins with analysing the mobile application development requirements, followed by the derivation of Input factors that leads to the collection of Mobile Function Factors (MFF). MFF is used as input parameters to calculate mobile application development's size and efforts with a hybrid tailor-made technique. This whole process involves the concepts of the COSMIC Plus Effort Estimation Model and Machine learning algorithm. The model implementation involves assured technology mathematical calculations that can help us execute the theoretical framework into possible practical results. Several sixteen mobile application projects are used in this research to derive statistical interpretation so that obtained results can be compared to the used mobile applications' actual values as a dataset. The final step is to send the proposed model to the mobile application estimators to check its industrial efficiency.

6. Conclusion and Future work

The work carried out in this paper contributes to the field of software engineering to calculate the overall efforts of mobile application development. An efficient approach can be used to provide effective results for the software project management team. This research work uses a hybrid approach that imbibes the concepts of CPEEM and Machine learning technique. In this research, we have utilised sixteen mobile application projects of the IFPUG repository as a dataset to use in our research for model evaluation and

validation. A descriptive statistic will be applied to the available dataset to check the performance of the tailor-made approach. The framework uses Mobile Functional Factors as input parameters for the proposed approach where CPEEM and Machine learning technique is used to calculate the size and efforts of mobile application development. These efforts are then compared with the real mobile applications' actual efforts to see whether the proposed approach is efficient or not. Therefore, this study attempts to propose a model that can conquer the challenges posed by other existing software estimation models and reduce the gap between actual efforts and mobile application development's estimated efforts. Future work is to design a model based on the framework given in this paper.

References

- [1] A. L. I. Oliveira, "Estimating software projects effort with support vector regression". *Neurocomputing*, vol. 69, no. 13-15, pp. 1749-1753, August 2006.
- [2] Andr'e Nitze (2014). "An Analogy-Based Effort Estimation Approach for Mobile Application Development Projects". In 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement.
- [3] Attarzadeh, I., Ow, S.: Proposing a new software cost estimation model based on artificial neural networks. In: Conference on Computer Engineering and Technology, pp. 487-491 (2010).
- [4] C. J. Burgess, M. Lefley. "Can genetic programming improve software effort estimation? A comparative evaluation". *Information and Software Technology*, vol. 43, pp. 863-873, 2001.
- [5] C. Paper and R. Berlin, "Measuring Mobile Application Size Using COSMIC "March 2018.
- [6] <https://www.streetinsider.com> (As accessed on 22/10/2017).
- [7] <http://www.sciencedirect.com/article/pii/S0920548919301916>
- [8] Huang, J., Li, Y., Xie, M.: An empirical analysis of data pre-processing for machine learning-based software cost estimation. *Inf. Softw. Technol.* 67, 108-127 (2015).
- [9] I. Gartner. (2013) Gartner says worldwide mobile phone sales declined by 1.7 percent in 2012. egham, UK: Gartner, 2013. [Online]. Available: <http://www.gartner.com/newsroom/id/2335616>.
- [10] Li, Y., Xie, M., Goh, T.: A study of project selection and feature weighting for analogy-based software cost estimation. *J. Syst. Softw.* 82(2), 241-252 (2009).
- [11] M. Shin, A.L. Goel, "Empirical data modeling in software engineering using radial basis functions". *IEEE Transactions on Software Engineering*, vol. 26, no. 6, pp. 567-576, June 2000.
- [12] Narendra Sharma and RatneshLitoriya, "Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement" *International Journal of Emerging Technology and Advanced Engineering*, 2012, vol.-2.
- [13] Oliveira, A., Braga, P., Lima, R., Cornello, M.: GA-based method for feature selection and parameters optimisation for machine learning regression applied to software effort estimation. *Inf. Softw. Technol.* 52(11), 1155-1166 (2010)
- [14] Vahid Khatibi, "Software Cost Estimation Methods: A Review", in *Journal of Emerging Trends in Computing and Information Sciences*, Volume 2 No. 1, ISSN 2079-8407.
- [15] Vinay Kumar, K., Ravi, V., Carr, M., Raj Kiran, N.: Software development cost estimation using wavelet neural networks. *J. Syst. Softw.* 81(11), 1853-1867 (2008)
- [16] X. Huang, D. Ho, J. Ren and L. F. Capretz, "Improving the COCOMO model using a neuro-fuzzy approach". *Applied Soft Computing*, vol. 7, pp. 29-40, January 2007.
- [17] Laxmi Lydia, E., Kannan, S., SumanRajest, S. and Satyanarayana, S. 'Correlative study and analysis for hidden patterns in text analytics unstructured data using supervised and unsupervised learning techniques', *Int. J. Cloud Computing*, Vol. 9, Nos. 2/3, pp.150-162, 2020.
- [18] Ziema Mushtaq*, Abdul Wahid, "Mobile complex factors: an approach for predicting mobile size parameters,RecentPat.Comput.Sci.(2019)12(2019)1,https://doi.org/10.2174/2213275912666190218152109
- [19] Ziema Mushtaq, Abdul Wahid, Revised approach for predicting the functional size of mobile application, *Appl. Comput. Inf.* (2019), <https://doi.org/10.1016/j.aci.2019.03.002>, 2019, ISSN, 2210-8327.
- [20] Ziema Mushtaq, Abdul Wahid, Inclusion of Functional and Non-Functional Parameters for the Prediction of Overall Efforts of Mobile Applications., *Computer Standards & Interfaces*, Volume 71, 2020,103404, ISSN0920-5489, <http://doi.org/10.1016/j.csi.2019.103404>
- [21] S. Suman Rajest, D.K. Sharma, R. Regin and Bhopendra Singh, "Extracting Related Images from E-commerce Utilizing Supervised Learning", *Innovations in Information and Communication Technology Series*, pp. 033-045, 28 February, 2021.
- [22] D Datta, S Mishra, SS Rajest, (2020) "Quantification of tolerance limits of engineering system using uncertainty modeling for sustainable energy" *International Journal of Intelligent Networks*, Vol.1, 2020, pp.1-8, <https://doi.org/10.1016/j.ijin.2020.05.006>
- [23] Leo Willyanto Santoso, Bhopendra Singh, S. Suman Rajest, R. Regin, Karrar Hameed Kadhim (2021), "A Genetic Programming Approach to Binary Classification Problem" *EAI Endorsed Transactions on Energy*, Vol.8, no. 31, pp. 1-8. DOI: 10.4108/eai.13-7-2018.165523
- [24] G. A. Ogunmola, B. Singh, D. K. Sharma, R. Regin, S. S. Rajest and N. Singh, "Involvement of Distance Measure in Assessing and Resolving Efficiency Environmental Obstacles," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021, pp. 13-18, doi: 10.1109/ICCIKE51210.2021.9410765.
- [25] Souvik Ganguli, Abhimanyu Kumar, Gagandeep Kaur, Prasanta Sarkar and S. Suman Rajest, "A global optimization technique for modeling and control of permanent magnet synchronous motor drive", *Innovations in Information and Communication Technology Series*, pp. 074-081, 28 February, 2021.
- [26] Jappreet Kaur, Tejpal Singh Kochhar, Souvik Ganguli and S. S Rajest, "Evolution of Management System Certification: An overview", *Innovations in Information and Communication Technology Series*, pp. 082-092, 28 February, 2021.
- [27] R. Regin, S. Suman Rajest and Bhopendra Singh, "Spatial Data Mining Methods Databases and Statistics Point of Views", *Innovations in Information and Communication Technology Series*, pp. 103-109, 28 February, 2021.
- [28] K. Sharma, B. Singh, E. Herman, R. Regine, S. S. Rajest and V. P. Mishra, "Maximum Information Measure Policies in Reinforcement Learning with Deep Energy-Based Model," 2021 International Conference on Computational Intelligence and Knowledge Economy, 2021, pp. 19-24, doi: 10.1109/ICCIKE51210.2021.9410756.

Appendix

Questionnaire to review the relevance of various parameters with Mobile application effort estimation

The relevance of parameters with Mobile application Effort Estimation		
Description for Rating scale:		
1: Less Important		
2: Unimportant		
3: Standard.		
4: Significant.		
5: Powerful.		
Based on your experience, please suggest whether the existing parameters listed below have relevance with modern-day Mobile application development to perform Mobile application effort estimation. You are requested to put rating against parameter(P1-P21) given below kindly:		
Label	Mobile Application Development Parameters	Rating (1-5)
P1	Limited energy	
P2	Small screen	
P3	Bandwidth	
P4	Change of context	
P5	Reduced memory	
P6	Connectivity	
P7	Interactivity	
P8	Usability	
P9	Security	
P10	Reliability	
P11	Efficiency	
P12	Interoperability	
P13	Response time privacy	
P14	Data integrity	
P15	screens	
P16	The constant interruption of activities	
P17	Functional area	
P18	Input interface factor and Navigation	
P19	Graphic interface factor	
P20	Context factor	
P21	User interface	
Keeping your expertise in mind, which new parameters will you suggest for inclusion to improve the effectiveness of mobile effort estimation		
Write comments: (please mark corrections if any)		
Expert Name:		Designation:
Organisation:		E-mail:
Please return the filled form to: Ziema Mushtaq E-mail: sheikhz123@gmail.com		