Predict the required cost to develop Software Engineering projects by Using Machine Learning

Ashraf Abdulmunim Abdulmajeed¹, Marwa Adee Al-jawaherry² and Tawfeeq Mokdad Tawfeeq³

¹Department of Software, College of Computer Sciences and Mathematics, University of Mosul, Iraq / Mosul
²Department of Computer Science, College of Computer Sciences and Mathematics, University of Tikrit, Iraq / Tikrit

E-mail: ashraf_althanoon@uomosul.edu.iq

Abstract. Software project cost prediction is a very important task during building and developing software projects. This process helps software project engineers to accurately manage and plan their resources in terms of cost estimation. However, Need for accurate cost development prediction model for a software project is not a simple procedure. Predicting the cost required while developing software engineering projects is the most difficult challenge that attracts the attention of researchers and practitioners. This paper adopts a new model in estimating the cost of building or developing software engineering projects using a machine learning approach. The results prove that machine learning methods can be used to predict program cost with high accuracy rate compared with traditional software estimation techniques. The proposed model in this research was trained on the NASA (National Aeronautics and Space Administration) data set, which contains the characteristics of 60 projects in addition to the real cost of the projects. An analysis of the results of the implementation for the proposed methods showed that the cost Predicting process using K-Nearest Neighbours algorithm (KNN) , Cascade Neural Networks (CNN) and Elman Neural Networks (ENN) It has the ability to predict the costs required to build or develop software engineering projects, K-Nearest Neighbours algorithm has shown high accuracy for Predict the required cost to develop Software Engineering projects Compared to Cascade Neural Networks and Elman Neural Networks ENN.

1. Introduction

Accurate forecasting is a critical task for developers and customers alike. The aim of this process is to have a better view to the future of the stages of project proceed. The other essential aim is to obtain specifications and details clearly of the project to be developed to help stakeholders in management of the project as far as software, human resources, data Furthermore actually in the financial possibility study of the project. Accurate prediction results certainly help the project administrator to make a better predict of the project time required for different project stages, cost and resources. However, inaccuracies might come about from the process of the project cost predicting which will negatively influence the project. The project that is wrongly evaluated or inaccurate in estimating its cost will encounter problems with required resources, budget, or even quality, and in some cases the project
might fail or be foiled. Hence, the predicting of the cost is one of the most important side of the software projects that must be taken care of and it is still a complicated matter in the field of software engineering [1]. Thus, many research and studies have been performed with the purpose of enhance and improve the process of prediction and getting results more precise and reliable.

On the other hand, machine learning (ML) technologies have recently become important in providing an accurate estimate of required costs during the early stages of a project's life cycle. The behavior and performance of the software project process has been used as a benchmark for cost predicting models. Knowing the expected cost of the project, the software management team will be able to control the software development process using an efficient method [2].. In many scientific researches, machine learning methods and their implementation are used in different domains, relying on the nature of the study and its goals, given that the process of Predicting the required cost to build the program is developing rapidly, which may include technological advancements, the skills of the programmers team working on developing projects, its expertise, programming languages available in addition to their obtainable tools. It gives progression to machine learning techniques over than some other techniques that might stick to arithmetical and statistical work [3]. Hence, ML can be an appropriate approach to building the proposed model in cost prediction due to the ability to learn from prior accomplished projects data and adaptation the wide difference that combined with the software project development.

The research aims to adopt 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. This research relies on completed projects in the NASA (National Aeronautics and Space Administration) data set, which contain the properties of 60 software projects, and also 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 were evaluated and comparing them using the standards (MMRE, RMSE, BRE), where the results using the K-Nearest Neighbors (KNN) algorithm showed high accuracy in the process of cost predicting needed to develop software engineering projects compared to the algorithm of CNN and the ENN network, where the last two networks obtained less accuracy in estimating Cost.

2. Related work

In 2008, researcher Sultan Aljahdali presented a paper in which a new method is presented, which is Differential Evolution (DE) as a technique for estimating and synthesizing the parameters of the COCOMO model. Then the performance of the proposed model for estimating effort and cost is tested on the NASA dataset, where it was observed that the developed model COCOMO-DE had very good results in the guessing process compared to other estimation models such as Walston-Felix, Bailey- and Doty models, Halstead, Basili, Fuzzy logic (FL) also gives results similar to the COCOMO-PSO model [4].

In 2017, researchers Omar and Betul presented a study about software effort estimation and they used the machine learning techniques. They designed model by using two machine learning techniques (Support Vector Machine (SVM) and K-Nearest Neighbor (k-NN)) and used two generic data sets (Desharnais and Maxwell). The designed model showed that merging these two techniques gives 91.35% accuracy for the Desharnais dataset and 85.48% accuracy for Maxwell dataset[5].

An in-depth study was conducted in 2018 with an analysis 25 versions holding hundreds of categories for effort indicators testing. The research showed that between 18 ML techniques used IBk, KStar, Additive Regression, and Multi-Layer Perceptron were able to accurately estimate effort testing.Moreover, Khaled et al. [8] presented a proposed prediction model to estimate the duration of the software by using ML algorithms. They used two training models (LM) and Bayesian Backward Spread Regular (BR)) to assess and test FFNN and RBNN algorithms. And they makes a comparison between the two models and the results showed that BR were a little better. Furthermore, BR is preferred because its implementation is effective in cost. Also,Yeh and Deng [9] have provided a
A new research was performed in 2019 on breast cancer [6] trying to design models to visualize and detect signs of breast cancer by using ML algorithms. Prediction algorithms were developed to specify aspects of breast cancer survival rate by using random forest SVM, extreme reinforcement, logistic regression, decision tree and KNN. Very high and close results were recorded for all the algorithms and the best one was the random forest, and we understand from this that we can use these algorithms in prediction of breast cancer. During the software cost predicting operation, Usually, several main challenges developed like innovation and technology factors. As well as other challenges like unforeseen problems which can happen through the implementation like reduction in resources or an overcharge in cost of the operating system. Therefore, Kumari and Pushkar [1] offered a hybrid algorithm for predict the cost in a better way and this algorithm based on incorporation of KNN and COA-Cuckoo. The proposed algorithm works on 6 varios datasets and assessed by using 8 criteria. The final results showed that there is a noticeable improvement in the accuracy of the cost prediction.

3. Software Cost Predicting

One of the most important activities in software engineering is predicting software development cost. It is usually used as a basis for bidding for a contract for the company or developer team and for resources to be allocated and plays an important role in project and decision making. Software prediction methods cover a wide range of methods that are used to predict effort, time and cost of the program [10].

Estimation is a statistical estimate with a reasonable value of accuracy within the center of the range.

The software estimation can be classified into three stages:
1. The first stage: includes estimating the size.
2. The second stage: It consists estimating the effort and the time.
3. The third stage: is to estimate the cost and estimate the number of employees required.

Figure 1 shows the overlap between the three stages in the typical software estimating process in the software development life cycle [11-19].

![Figure 1. shows the sequence of estimation processes in the software development life cycle](image)

4. Methodology

In this part, the methodology of the work is discussed by illustrating the ML algorithms used and common methods of estimating costs and comparing results. The data sets used and their structure are
then explained. Finally, the discussion focuses on the evaluation criteria. Figure 2 illustrates the general framework for work:

![Figure 2. Methodology of the study](image)

4.1. Dataset used
The data used in this work are implemented and completed projects and derived from 60 NASA projects. A ready historical database was used for previous implemented and completed projects, and the estimated cost of each project was calculated in it accurately, and this data is provided by NASA space agency [12].

4.2. Pre-Processing Procedures
When using ML techniques or ANNs as a tool in estimating, the estimator must take several basic steps for any neural network. In this research, the estimated cost of each project in it was calculated accurately and this data is provided by NASA space agency [12].

Each of these projects contains 15 values representing the values of the fifteen cost driver factors. Table 1 shows examples of these values in their fuzzy case [13-19].

| Cost factor | RELY | DATA | CPLX | TIME | STOR | VIRT | TURN | ACAP | AEXP | PCAP | VEXP | LEXP | MORP | TOOL | SCED | KSLOC | Actual effort |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|----------------|

Table 1. NASA dataset used in the research with their fuzzy values
Where each of these fuzzy values has a corresponding numerical value according to its own cost factor. Table 2 shows us, which represents the NASA data used in its numerical values [13].

**Table 2.** NASA dataset used in the research with its numerical values

| Cost factor | RELY | DATA | CPLX | TIME | STOR | VIRT | TURN | ACAP | AEXP | PCAP | VEXP | LEXP | MODP | TOOL | SCED | KSLOC | Actual effort |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|---------------|
| Project 1   | 1.15 | 0.94 | 1.15 | 1.00 | 1.00 | 0.87 | 0.87 | 1.00 | 1.00 | 1.00 | 1.00 | 0.95 | 0.91 | 1.00 | 1.08 | 2.2   | 8.4           |
| Project 30  | 1.15 | 1.00 | 1.15 | 1.00 | 1.00 | 1.00 | 0.91 | 0.86 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 38   | 210              |
| Project 60  | 1.15 | 1.00 | 1.15 | 1.11 | 1.06 | 0.87 | 1.07 | 1.00 | 0.91 | 1.00 | 1.00 | 1.00 | 1.10 | 0.83 | 1.00 | 370  | 3240          |

4.3. *Techniques applied*

Over view of the techniques applied.

4.3.1. *KNN Algorithm.*

The closest KNN algorithm works when there is no previous knowledge about describing data. The goal of use KNN to predict unknown future events or scenarios for teams. The technique works by calculating the distance from one instance to another and finding the neighbor k closest to that state.
The KNN algorithm was first described in K selection which considers a very important matter. If the value of K is specified too small, the algorithm becomes noise sensitive. If the value of K is specified too large, it leads to an error in estimating and other classes can also be included among the closest neighbors [14]. In this technique the closest K neighbors are measured. In order to describe a typical data point class, K shows the number of neighbors The closest to be checked. KNN technology falls into two categories (structure-based and structure less). The three fundamental elements of this approach are:

1. an existing set of labeled objects.
2. a distance metric to Predict distance between objects.
3. the number of nearest neighbors (k).

The structure-based KNN algorithm deals with the data basic structure wherein the topology has a mechanism less related to the training data samples approved in this paper. In contrast, for unstructured KNN technology, the data that is fully handled is classified into sample data and training data point.

Here, the calculated distance between the sample points for which the estimated cost is to be calculated and all the training points and the point with the smallest distance is known as the nearest neighbor [15-20]. The results of the assessment were approximately 90% identical to the real cost of Target, and with this it was verified that the network is ready to give us a cost estimate that accurately approximates the real cost and can be relied upon in predicting during the development of any software project as shown in the following Table 3:

| Projects No | Proj.1 | Proj.10 | Proj.20 | Proj.30 | Proj.40 | Proj.50 | Proj.60 |
|-------------|--------|---------|---------|---------|---------|---------|---------|
| Actual cost | 62     | 300     | 48      | 10.8    | 120     | 370     | 60      |
| Estimated cost | 58.8   | 345.6   | 42.9    | 13.3    | 109.8   | 344     | 51.5    |

4.3.2. Cascade Neural Network

The CNN is created and trained using the new cf function in MATLAB. In this phase, CNN is trained to assess the programming cost and give an estimate of the cost closer to the real cost by entering and processing the training set data in addition to the inputs for this phase. The outputs of this phase will be the time spent in the training process as well as the ideal weights that are stored in the traind_CNN.mat file for later use in the testing process and the tr variable that contains the details of the network training. The network is tested by entering the validation set data as well as the traind_CNN.mat file, and the results of the estimation were approximately 85% identical to the real cost of Target. Thus, it was verified that the network had been trained well and is now ready to give us an estimated cost for the software projects to be built. Results of the trained network test to estimate the real cost as shown in the following Table 4:

| Projects No | Proj.1 | Proj.10 | Proj.20 | Proj.30 | Proj.40 | Proj.50 | Proj.60 |
|-------------|--------|---------|---------|---------|---------|---------|---------|
| Actual cost | 62     | 300     | 48      | 10.8    | 120     | 370     | 60      |
| Estimated cost | 58.8   | 345.6   | 42.9    | 13.3    | 109.8   | 344     | 51.5    |
### 4.3.3. Elman Neural Network

The ENN is created and trained using the newelm function in MATLAB. Where, in this phase, the ENN network is trained to estimate the cost and give an estimate as close as possible to the real cost by entering and processing the training set data, in addition to the inputs for this phase. The outputs of this phase will be the time spent in the training process as well as the ideal weights that are stored in the traind_ENN.mat file for later use in the testing process and the tr variable that contains the details of the network training. The network is tested by entering the validation set data as well as the traind_ENN.mat file, where the results of the cost estimation were approximately 81% identical to the real cost. The results of the trained network test to estimate the cost were as shown in the following Table 5:

| Projects No | Proj.1 | Proj.10 | Proj.20 | Proj.30 | Proj.40 | Proj.50 | Proj.60 |
|-------------|--------|---------|---------|---------|---------|---------|---------|
| Actual cost | 62     | 300     | 48      | 10.8    | 120     | 370     | 60      |
| Estimated cost | 76.3 | 368.8  | 36      | 17.2    | 129.8   | 243.4   | 43.3    |

#### Table 5. results of the ENN estimation

### 5. Evaluating the performance of the models

To make a comparison between the used models, the following measures were used:

#### 5.1. Mean Magnitude Relative Error (MMRE)

It is the percentage of absolute values of mean relative error (MRE) divided by N and its equation is [22-23]:

$$\text{MMRE} = \frac{1}{N} \sum_{i=1}^{N} \text{MRE}_i \times 100$$  \hspace{1cm} (1)

Where

$$\text{MRE} = \left| \frac{\hat{C} - C}{C} \right|$$  \hspace{1cm} (2)

#### 5.2. Root Mean Square Error (RMSE)

This scale computes the square root of the mean square error computed between the true value (Target) and the resulting value divided by N and its equation is [17]

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (C_i - \hat{C}_i)^2}$$  \hspace{1cm} (3)

#### 5.3. BRE Balanced Relative Error
It is the absolute value of the error divided by the smaller value between the estimated voltage Ĉ and the real voltage E [16-18].

\[
BRE = \frac{|c - Ĉ|}{\min (c, Ĉ)}
\]  

(4)

Where in all measures, Ĉ is the estimated cost, C is the actual cost, and N is the total number of projects. And in all the scales used in this research, the lower the scale value, the better the result.

6. Results and Analysis

In this paper, the project cost estimation process for software projects was conducted based on ML technology to estimate the program cost. Therefore, the K-Nearest Neighbors algorithm was used. The goal of the proposed model is to predict the cost by using the characteristics of the data set and comparing it with the real cost in order to increase the accuracy in cost predicting. The project cost estimation process for software projects was also performed using both the Cascade Neural Network algorithm and the Elman Neural Network algorithm, where the results showed that the KNN-based model has a higher accuracy in estimating and predicting the costs required to develop software projects compared to both the Cascade Neural Network algorithm and Elman Neural Network algorithm gave the worst results.

Table 6 shows a comparison between the results of the methods used in the search and Prediction Accuracy using the standards (MMRE, RMSE, BRE):

| Models Used         | Factor        | MMRE  | RMSE  | BRE  |
|---------------------|---------------|-------|-------|------|
| K-Nearest Neighbors |               | 0.101 | 0.547 | 0.205|
| Cascade Neural Network |           | 4.938 | 31.875| 0.301|
| Elman Neural Network |              | 43.16 | 184.10| 0.859|

Figure 3. MMRE values of predict cost using KNN algorithms and CNN, ENN Neural Network for NASA dataset.
In future, we can use more datasets in the study in order to have a broader and more varied view of the inputs which will be reflected to get a better predict and more precise results. Furthermore, we can tested several other machine learning algorithms and involved in future researches to covering all machine learning algorithms.

7. Conclusion
Cost estimation remains a complex problem that attracts researchers to study and to try different approaches to solve it. In this paper three intelligent technologies were used to predict the costs required for developing software projects K-Nearest Neighbors algorithm, Cascade Neural Network algorithm and Elman Neural Network. This models was tested on NASA project data presented in [12] and the results were compared with both KNN,CNN and ENN. Where the results showed that the KNN a higher accuracy in estimating and predicting the costs required and based model had the lowest values of MMRE and RMSE, BRE i.e. 0.101, 0.547 and 0.205 respectively. When the accuracy of the proposed system was calculated and then showed 90.238% accuracy. This means KNN technique that the system shows a very low error and a high value for the desired accuracy of the cost predicting process. Therefore KNN is recommended to be used as a model based system for predicting the estimated cost of the projects to be built or developed. For future work several different methods can be applied to solve the problem of predicting the cost closest to the real cost, and this work can be considered a starting point for the launch of various future businesses, such as the use of other intelligent methods in addition to the methods used in this paper such as GA, Fuzzy Logic, and Swarm intelligent and other methods, or it can be relied on a different database of certified and reliable software companies . Also, non-computational methods can be used to find the software prediction and combine them with the computational methods.

Acknowledgment
The authors would like to thank the University of Mosul/ College of Computer Sciences and Mathematics for their provided facilities, which helped to improve the quality of this work.
References

[1] S. Kumari and S. Pushkar, "Cuckoo search based hybrid models for improving the accuracy," Microsystem Technologies, vol. 24, no. 12, pp. 4767-4774, 2018.

[2] Zhiwei Xu, Taghi M. Khoshgoftaar, 2003, "Identification of fuzzy models of software cost estimation", Elsevier B.V,141-163.

[3] P. Pospieszny, B. Czarnacka-Chrobot and A. Kobylnski, "An effective approach for software project effort and duration estimation with machine learning algorithms," The Journal of Systems & Software, vol. 184, 2017.

[4] Sultan Aljahdali, 2007, "Development of a Software Effort Estimation Model Using Differential Evolution", Journal of Electronics & Computer Science.

[5] Omar Hidmi and Betul Erdogdu Sakar, Software Development Effort Estimation Using Ensemble Machine Learning, Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) vol. 4, Issue 1 (2017) ISSN 2349-1469 EISSN 2349-1477.

[6] M. D. Ganggayah, N. A. Taib, Y. C. Har, P. Lio and S. K. Dhillon, "Predicting factors for survival of breast cancer patients using machine learning techniques," BMC Medical Informatics and Decision Making, pp. 1-17, 2019.

[7] V. Vig and A. Kaur, "Test effort estimation and prediction of traditional and rapid release models using machine learning algorithms," Journal of Intelligent & Fuzzy Systems, vol. 35, pp. 1657-1669, 2018. Available at: https://doi.org/10.3233/jifs-169703.

[8] A. Khalid, M. A. Latif, and M. Adnan, "An approach to estimate the duration of software project through machine learning techniques," Gomal University Journal of Research, vol. 33, pp. 1-13, 2017.

[9] T.-H. Yeh and S. Deng, "Application of machine learning methods to cost estimation of product life cycle," International Journal of Computer Integrated Manufacturing, vol. 25, pp. 340-352, 2012. Available at: https://doi.org/10.1080/0951192x.2011.645381.

[10] M. Jørgensen, "A review of studies on expert estimation of software development effort," Journal of Systems and Software, vol. 70, pp. 37-60, 2004/02/01/ 2004.

[11] Topi Haapio, 2011, "Improving Effort Management in Software Development Projects", Dissertation in forestry and natural sciences, university of Eastern Finland.

[12] Karen Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, Erik Monson, 2003, "Handbook for Software Cost Estimation", Jet Propulsion Laboratory, Pasadena, California.

[13] Esra's Z.M.Quba, 2012, " Software Projects Estimation using Neural Networks "

[14] A. A. Soofi and A. Awan, “Classification techniques in machine learning: applications and issues,” Journal of Basic & Applied Sciences, vol. 13, pp. 459-465, 2017.

[15] M. Hasanluo, F. Soleimanian Gharehchopogh, "Software cost estimation by a new hybrid model of particle swarm optimization and k – nearest neighbor algorithms," Journal of Electrical and Computer Engineering Innovations (JECEI), vol. 4, no. 1, pp. 49-55, 2016.

[16] Alaa F. Sheta, Alaa Al-Afeef, 2010, "A GP Effort Estimation Model Utilizing Line of Code and Methodology for NASA Software Projects", IEEE, International Conference on Intelligent Systems Design and Applications.

[17] Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi, 2010,"Neural Network-A Novel Technique for Software Effort Estimation", International Journal of Computer Theory and Engineering.

[18] Parvinder S. Sandhu, Porush Bassi, and Amankanet Singh Brar, 2008, "Software Effort Estimation Using Soft Computing Techniques", World Academy of Science, Engineering and Technology.

[19] Robert T. Futrell, Donald F. Shafer, Linda I. Safer, 2002, "Quality Software Project Management", Prentice Hall PTR.

[20] gerardnico, "Machine Learning - K-Nearest Neighbors (KNN) algorithm - Instance based learning," 2017. [Online]. Available: https://gerardnico.com/.
[21] J. Wang, L.-C. Yu, K. R. Lai and X. Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016.

[22] ANFAL A. FADHIL, RASHA GH. ALSARRAJ and ATICA M. ALTAIE, 2020, "Software Cost Estimation Based on Dolphin Algorithm", IEEE.

[23] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," Comput. Intell. Neurosci., vol. 2019, pp. 1-17, Feb. 2019.