ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Multi-Objective Software Project Cost Estimation Using Recent Machine Learning Approaches

Doğay DERYA[1], Osman Berkcan DERYA[2], Tansel DÖKEROĞLU[3*]
[1]Çankaya University, Graduate School of Natural and Applied Sciences, Turkey; ORCID 0009-0006-1680-0530
[2]Çankaya University, Graduate School of Natural and Applied Sciences, Turkey; ORCID 0009-0000-8541-8446
[3]Çankaya University, Software Engineering Department, Ankara, Turkey; ORCID 0000-0003-1665-5928
*Corresponding Author: tdokeroglu@cankaya.edu.tr

Received: 26 March 2023; Accepted: 17 July 2024

Abstract

The success of software projects for organizations heavily relies on accurate workforce and cost estimates. Initially, effort estimation was based on non-algorithmic methods, but with technological advancements, algorithmic approaches such as regression emerged. In recent years, there has been a growing interest in using Machine Learning and Artificial Intelligence for software cost estimation. In this study, Linear Regression, Multilayer Perceptron, Bagging, SMOreg, IBk, KStar, RandomTree, and RandomForest algorithms were trained with four open-source datasets. Firstly, models were trained with original feature sets, then six different hybrid feature selection methods were proposed to eliminate low-impact features and prevent overfitting. These hybrid feature selection methods, developed using evaluation methods like Relief, Classifier, and Correlation, along with search methods like RandomSearch, PSO, GA, and Ranker. And trained models tested by the 10-fold cross-validation technique. The results showed the ability to quickly obtain adaptable models and the effectiveness of feature selection. KStar, SMOreg, Multilayer Perceptron, and Linear Regression algorithms, along with PSO and GA search methods, yielded satisfactory results even with different feature subsets.

Keywords: Software Cost Estimation, Software Effort Estimation, Artificial Intelligence, Machine Learning, Feature Selection

1. Introduction

Effective project management becomes indispensable for software projects that increase in importance and scope in parallel with the increase in trust in electronic technologies.

Project predictability is a critical factor in software project management, as it makes possible to mitigate potential risks by enabling precise cost and workforce planning. Accurate software effort estimation is a crucial component of software development, providing essential inputs for feasibility analysis, planning, budgeting,bidding. Deviating significantly from the required effort causes losses in terms of cost and quality. Thus, it isparticularly important to estimate development time accurately in the highly competitive software industry, where quality is highly valued.

Currently, the most prevalent methods for effort estimation rely on expert judgment. However, these methodsmay lack reliability as they can be influenced by various factors. Additionally, relying solely on human judgment can be burdensome and time-consuming when dealing with numerous estimation items.

In recent years, the dynamic nature of the market has led to a growing adoption of agile methods in software project management, replacing traditional approaches. Within the agile project management methodology, the most commonly used metric for effort prediction is story scores. Presently, these estimations are typically made intuitively by relevant individuals for each request, with subsequent review by unit managers. However, this process lacks consistency and continuity, despite consuming significant human resources. On the other hand, machine learning-based models, by quickly analyzing complex relationships between inputs and outputs even in large datasets through iterative cycles of training, increase the chance of producing accurate predictions.

The objective of this study is to propose a machine learning-based approach for effort estimation, aiming to accurately and swiftly predict effort. The study will handle machine learning approach that establish models by learning from past data to predict development efforts. Furthermore, innovative feature selection techniques will be employed to enhance the accuracy and effectiveness of the estimation process. The open-source WEKA platform has been preferred to enable the rapid and efficient training and testing of the selected techniques, aiming to provide a widely applicable approach.

In the study, algorithms from WEKA (Waikato Environment for Knowledge Analysis) were tested and compared for their performance based on data characteristics in the Functions, Lazy Classifiers, Meta, and Tree categories. For this analysis, Functions-based Linear Regression, Multilayer Perceptron, SMOreg, Lazy Classifiers-based IBk, KStar, Meta-based Bagging, and Tree-based RandomTree, Random Forest, M5P algorithms were selected and trained and tested with both the original feature set and after applying feature selection to enhance model performance and prevent overfitting by focusing on unnecessary inputs. Hybrid approaches of evaluation and search methods were used together in different configurations for feature selection. Search methods such as RandomSearch, PSO, GA, Ranker were selected, and their capabilities in searching optimized subsets were utilized.

Knowing the approximate cost of a project at the beginning of the project is important for the reasons for starting the project. The customers of the project or the top management decides whether or not to carry outthe project according to the predictive values. Incorrect estimations make the institutions or organizations inthe position of customers economically and strategically affects. For example, 60% of large projects exceededtheir project budgets. It has been observed that some projects were never completed due to a 15% cost overrun [1].

Software effort estimation is difficult, mainly for two reasons. The first reason is that software is intangible and is outside the definition of conventional physical product. The second reason is that the software development job is an intellectual rather than a physical job. Software startups are easy, but as the software size increases, the workforce estimation process becomes more difficult. It is possible to write a program thatis close to a few thousand lines in a week. But then the speed slows down as the program grows. When this program reaches several tens of thousands of lines, adding a line is worth a few days' effort, maybe even months. Therefore, it has become difficult to follow the side effects of the addition [2]. The dynamically fluctuating technology environment in the software development industry also makes effort estimation confusing [3]. Contributions of this study are:

- High-performance approaches were emphasized by training, testing and comparing 9 different machine learning algorithms with 6 different feature selection methods in four different datasets.
- With the WEKA tool, which is easily accessible due to its open source nature, alternatives to lowexecution time, high predictive models have been presented.
- When the estimation error rates obtained were compared with the results in the literature, it was observed that successful performances were achieved.

2. Datasets Used in Our Study

In this study, Finnish, Kemerer, Maxwell and China datasets were examined for software cost estimation from the Promise Data repository [4] The primary objective behind utilizing these datasets is their widespread recognition, simplicity, and accessibility to the public. This facilitates easy replication and verification of results, and potentially encourages further exploration and expansion. It is important to note that the approachis not limited to any specific dataset or model but can be applied across various datasets and models. Relateddatasets ‘s information.

Table 1: The related datasets‘s information which includes China, Finnish, Kemerer, Maxwell

Dataset	Project Number	Feature Number	Size (Measure Unit)	Cost (Measure Unit)
China	499	19	Function Point	Man-Hour
Finnish	38	9	Function Point	Man-Hour
Kemerer	15	8	KSLOC	Man--Month
Maxwell	62	27	Function Point	Man-Hour

3. Computation Environment

This study was conducted utilizing the WEKA platform, which is an open-source application written in Java. It was originally developed by a PhD student at the University of Waikato in New Zealand and is governed by the General Public License. WEKA offers a range of algorithms for performing machine learning and data engineering tasks, including classification, clustering, visualization, estimation, correlation analysis, feature selection, and data preprocessing for scientific research. The version utilized in this study was WEKA 3.8.6 (WEKA, 2022).

While WEKA is installed, it presents the weka.jar file, which includes the necessary libraries. WEKA Jar allows for the development of projects by accessing WEKA classes from other platforms such as Java or C#. Within WEKA, datasets are typically in the arff (Attribute Relationship File Format) extension, although it also supports other formats such as textual csv, dat, libsvm, json, and xrff.

4. Feature Selection

Estimating the cost in software projects relies on various factors, including the technology employed, the expertise of developers, the team's past project experiences in a similar domain, and the specific characteristics of the functions being developed. Software workforce estimation is a challenging task due to the multitude of parameters involved, and accurately predicting the relationships between these parameters is not always feasible. To address these ongoing challenges, techniques are continuously evolving to mitigate their impact. Numerous approaches and methods have been suggested to enhance the accuracy and success rate of effort estimation values.

In general, useful features are unpredictable, and features with low correlation and missing data can affect classification performance. Including low-impact variables in model training reduces the model's ability to generalize and may also reduce the overall accuracy of a classifier. Also, adding more variables to a model increases the overall complexity of the model. Therefore, deciding on the optimum features to include in model training is critical in obtaining a generically high-performing model. Various techniques are used in various fields to eliminate unnecessary features.

Various techniques are used in various fields to eliminate unnecessary features. The techniques for feature selection in machine learning can be broadly classified into the following categories:

➢ Feature selection based on combining the features for evaluation

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

➢ Feature selection based on the supervised learning algorithm

Feature selection based on combining the features for evaluation: They are classified into feature subset-based and feature ranking-based methods. In the feature subset-based method, the features are combined as possible combinations of feature subsets using any one of the searching strategies. Then, the feature subsets are evaluated using any one of the statistical measures or the supervised learning algorithms to observe the significance of each subset and the most significant subset is selected as the significant feature subset for a given dataset. If the subset is evaluated using the supervised learning algorithm, then this method is known as wrapper method [5] PSO, GA are heuristic searching strategies. One of the widely accepted fundamental benefits of metaheuristic algorithms is that they provide mechanisms to solve large or intractable problems in reasonable execution times while the exact algorithms fail to succeed due to time limitations [6]. Numerous research works on feature selection have utilized the genetic algorithm to create subsets of features for evaluation, with the supervised machine learning algorithm employed to assess these subsets. For instance, Erguzel et al. utilized the genetic algorithm and artificial neural network to classify electroencephalogram signals [7]. Oreski & Oreski proposed an approach for feature selection that combined GA with neural networks for credit risk assessment [8]. Additionally, Wang et al. applied the GA to generate subsets alongside SVM in the process of feature selection for data classification applications [9]. In their research, Yang et al. created a feature selection method for land cover 16 classification using PSO [10]. Feature ranking-based methods involve weighting each feature in a dataset based on statistical or information-theoretic measures and then ranking them according to their weights. The significant features are selected using a predetermined threshold that determines how many features will be chosen from the dataset. Since these methods do not require a supervised learning algorithm to assess feature significance, they follow a filter-based approach. As a result, feature ranking-based methods are more versatile and computationally efficient, regardless of the specific supervised learning algorithm used. Hence, they are a viable choice for selecting important features from datasets with high dimensions. From a taxonomic point of view, these techniques are classified into filter, wrapper, embedded, and hybrid methods. Hybrid methods are a fusion of filter and wrapper-based approaches. Dealing with high-dimensional data can be challenging when using the wrapper method. To address this, Bermejo et al. devised a hybrid feature selection method called the filter wrapper approach. In this method, they initially employ a statistical measure to rank the features based on their relevance. The higher-ranked features are then passed on to the wrapper method, which significantly reduces the number of evaluations required, making it a linear process. As a result, this hybrid approach reduces the computational complexity when applied to medical data classification tasks. The hybrid algorithms are developed by combining the current metaheuristics or classical algorithms. The main purpose of hybrid algorithms is to combine the skills of diverse algorithms to obtain better results. Therefore, hybrid metaheuristic algorithms have significant improvements compared to single metaheuristic algorithms [11]. Ruiz et al developed a feature selection algorithm for selecting the significant genes for the medical diagnosis system. They used a statistical ranking approach to filter the features from high-dimensional space and the filtered features are fed into the wrapper approach. This combination of the filter and wrapper approach was used to distinguish the significant genes causing cancer disease in the diagnosis process [12].

Hybrid methods are a fusion of filter and wrapper-based approaches. Dealing with high-dimensional data can be challenging when using the wrapper method. To address this, Bermejo et al. devised a hybrid feature selection method called the filter-wrapper approach. In this method, they initially employ a statistical measure to rank the features based on their relevance. The higher-ranked features are then passed on to the wrapper method, which significantly reduces the number of evaluations required, making it a linear process. As a result, this hybrid approach reduces the computational complexity when applied to medical data classification tasks. The hybrid algorithms are developed by combining the current metaheuristics or classical algorithms. The main purpose of hybrid algorithms is to combine the skills of diverse algorithms to obtain better results. Therefore, hybrid metaheuristic algorithms have significant improvements compared to single metaheuristic algorithms [11]. Ruiz et al developed a feature selection algorithm for selecting the significant genes for the medical diagnosis system. They used a statistical ranking approach to filter the features from high-dimensional space and the filtered features are fed into the wrapper approach. This combination of the filter and wrapper approach was used to distinguish the significant genes causing cancer disease in the diagnosis process [12].

5. Machine Learning Algorithms

In this section, the ML algorithms used in our study and included in the classification area of the WEKA tool are presented.

ML algorithms in WEKA are listed under the following headings and the algorithms used in model training in our study are listed under the relevant headings.

a. Functions
   - o LinearRegression
   - o Multilayer Perceptron
   - o SMOreg (Sequential Minimal Optimization Regression)

b. Lazy Classifiers
   - o IBk (K-nearest neighbors classifier)
   - o KStar (Instance-based classifier)

c. Meta
   - o Bagging

d. Tree
   - o M5P (M5 Model trees)
   - o RandomForest
   - o RandomTree

6. Feature Selection Techniques

Attribute selection in WEKA is performed by the Attribute Evaluator and Search method working together. Attribute Evaluator evaluates the importance of the attributes and tries to find the best set of attributes, guidedby the Search method. This approach is used to evaluate the quality of features and to eliminate unimportantfeatures, so that a smaller and more meaningful set of features can be obtained. This can provide the model with a better generalization ability and a faster training time. Feature selection can reduce the dimensionality to enable many data mining algorithms to work effectively on data with large dimensionality [13].

Selecting Attribute Evaluator: The first step is to select the Attribute Evaluator method. The Attribute Evaluator measures the effect of each attribute on classification or regression. Weka has various Attribute Evaluator methods, such as Information Gain, Gain Ratio, ReliefF, Chi-Square, etc. Choosing one of these methods determines the evaluator who will rate the importance of the features.

Search Method Selection: The second step is the selection of the Search method to be used in the feature selection. Search methods try to find the best set of attributes based on the importance rating generated by theAttribute Evaluator. Various Search methods are available in Weka, for example GreedyStepwise, BestFirst,GeneticSearch, etc. Choosing one of these methods determines a search strategy to find the best feature set.

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Attribute Selection: Attribute selection is performed using the selected Attribute Evaluator and Search method. In this step, the necessary parameters for feature selection are set and the selection process is started. Evaluation and selection of features are performed on a specific criterion or threshold value. As a result, the best featureset is determined.

In this section, the Attribute Evaluators and Search Methods used in our study and included in the SelectAttributes area of the WEKA tool are presented.

Attribute Evaluators:

- CfsSubsetEval
- ClassifierAttEval
- Corr. Att.Evaluation
- Relief

Att.Evaluation Search
Methods:

- Random Search
- Particle Swarm Optimization (PSO)
- Genetic Algorithm (GA)
- Ranker

7. Performance Measures

Correlation Coefficient:

The Correlation Coefficient is a statistical value that measures the strength and direction of the relationship between two variables. It is often called the Pearson Correlation Coefficient and takes values between -1 and +1.The formula for the Pearson Correlation Coefficient is expressed as:

$$r = (\Sigma((xi - x) * (yi - \bar{y}))) / \sqrt{((\Sigma(xi - x)\ 2) * (\Sigma(yi - \bar{y})2))} \tag{1}$$

Formula:

- r represents the Correlation Coefficient.
- xi and yi represent the values of the data points.
- x and ȳ represent the mean values of xi and yi.

Mean Absolute Error (MAE):

Mean Absolute Error (MAE) is a method of evaluating the accuracy of a prediction model by calculating the mean of the absolute differences between the measured and predicted values. MAE measures how close a model's predictions are to the true values and represents the mean errors of the predictions.

The formula for MAE is expressed as follows:

$$MAE = (1/n) * \Sigma|yi - xi| \tag{2}$$

Formula:

- MAE stands for Mean Absolute Error.
- n stands for the total number of data points.
- yi represents the true value.

- xi represents the predicted value.

Relative Absolute Error (RAE):

Relative Absolute Error (RAE) calculates the accuracy of a predictive model. RAE can be used in machine learning. Furthermore, RAE is expressed as the ratio; it computes the mean error (residual) of errors producedby a trivial or naive model. The model is considered non-trivial if the result is less than 1. This is the model for a dataset (k):

$$R_k = \frac{\sum_{i=1}^{n} \left| E_{ki} - D_i \right|}{\sum_{i=1}^{n} \left| D_i - \bar{D} \right|} \tag{3}$$

where Ei's is prediction, Di's is actual values, and Rae is the measure of forecast accuracy. D is the mean of Di's; n is the size of the dataset (in data points)

8. Findings

At this stage, considering the Finnish, Kemerer, China, Maxwell datasets implemented and choosing the 10-fold cross-validation technique.

- Firstly, models were created with the original datasets,
- In the second part, by using the hybrid configurations of given below evaluation and search methods among feature selection methods for the the same datasets, optimized and formed most effective features subsets. And these subsets were used to create models.

1. CFS+ RandomSearch
2. CFS+ PSO
3. CFS+ GA
4. ClassifierAttEval + Ranker
5. Corr. Att.Evaluation + Ranker
6. Relief Att.Evaluation + Ranker

Therefore, each discussed algorithm was initially tested with the original data, and then the most effective feature subsets obtained from the same datasets were evaluated with nine algorithms using six different hybrid methods for each subset. In the first stage, the results obtained with the original dataset will be examined, and in the second stage, the findings obtained as a result of the feature selection applied dataset will be presented. Finally, by examining the performance criteria reached with the results of methods obtained without feature selection and with different feature selection methods:

▪ The highest performances achievable with the original datasets,

▪ Dataset-specific and holistic analysis of algorithms that demonstrate the highest performance in the model formed with the original data,

▪ Highest achievements after attribute selection,

▪ Analysis of which feature selection is superior compared to the others,

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Ultimately, specific to the dataset and holistically, the aim is to obtain a generic approach that is not reliant on the specific dataset by considering the overall evaluation of these results.Specific to the dataset and holistically, Ultimately, the goal is to obtain a generic approach that is not reliant on the specific dataset by considering the overall evaluation of these results.The performance evaluation of the models was carried out by considering the Correlation Coefficient as well as several error metrics, including MAE and RAE.

In order not to be affected by small deviations while examining the results, the values close to the best and the worst results with a small percentage difference were added to the table. In addition, due to its higher resistance to overfitting, models with less number of feature subsets and close to the best results are also included. Table 1 presents Finnish model performance measurements.

Table 2. Finnish Model Performance Measurements

Finnish Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	Kstar	9	Original Feature Set	0.9889	0.1344	13.1344
The Lowest Result Without Feature Selection	IBk	9	Original Feature Set	0.7697	0.539	52.6711
The Hightest Results	Kstar	6	ClassifierAttEval+ Ranker	0.9948	0.0873	8.5274
	RandomForest	5	CFS + PSO	0.9942	0.0976	9.5354
	RandomForest	5	CFS+ GA	0.9942	0.0976	9.5354
The Lowest Results	Ibk	6	ClassifierAttEval+ Ranker	0.7421	0.5756	56.2528

Table 2 presents Finnish model performance measurements.

Table 3. China Model Performance Measurements

China Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	Linear Regression	19	Original Feature Set	0.9889	362.939	9.809
The Lowest Result Without Feature Selection	IBk	19	Original Feature Set	0.8918	1571.1824	42.4638
The Hightest Results	MultilayerPerceptron	16	Relief. Att.Evaluation + Ranker	0.9914	370.1846	10.0048
	MultilayerPerceptron	16	Corr. Att.Evaluation + Ranker	0.9912	406.1195	10.976
	SMOreg	16	Relief. Att.Evaluation + Ranker	0.9898	269.3637	7.28
	SMOreg	7	Random Search	0.9866	351.7668	9.5071
	LinearRegression	9	CFS + PSO	0.986	395.7794	10.6966
	LinearRegression	10	CFS+ GA	0.9859	411.7442	11.1281
The Lowest Results	IBk	16	Corr. Att.Evaluation + Ranker	0.8396	1418.9499	38.3495
	Random Tree	16	Relief. Att.Evaluation + Ranker	0.8098	1263.9482	34.1603

Table 3 shows China model performance measurements.

Table 4. Maxwell Model Performance Measurements

Maxwell Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	SMOreg	27	Original Feature Set	0.8191	3812.9653	60.6894
The Lowest Result Without Feature Selection	Ibk	27	Original Feature Set	0.463	5517.129	87.8139
The Hightest Results	Kstar	20	CFS+ GA	0.8596	4078.3244	64.913
	LinearRegression	20	CFS+ GA	0.8544	3395.0666	54.0379
	Kstar	9	CFS + PSO	0.85	4040.6726	64.3137
The Lowest Results	Ibk	24	Corr. Att.Evaluation + Ranker	0.4487	5720.7419	91.0547
	Random Tree	20	CFS+ GA	0.4398	5223.2222	83.1359

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Table 4 presents Maxwell model performance measurements.

Table 5. Kemerer Model Performance Measurements

Kemerer Dataset						
	Machine Learning Algortihm	**Number Of Selected Features**	**Feature Selection Technique**	**Correlation Coefficient**	**MAE**	**RAE (%)**
The Highest Result Without Feature Selection	SMOreg	8	Original Feature Set	0.5737	114.3301	71.0419
The Lowest Result Without Feature Selection	Random Tree	8	Original Feature Set	-0.0271	250.9131	155.9111
The Hightest Results	SMOreg	5	Corr. Att.Evaluation + Ranker	0.7171	103.4371	64.2732
	SMOreg	5	CFS + PSO	0.6946	96.4073	59.9051
	SMOreg	5	CFS+ GA	0.6946	96.4073	59.9051
The Lowest Results	Bagging	5	CFS+ RandomSearch	0.1168	182.9247	113.6648
	Random Tree	5	CFS+ RandomSearch	0.1189	194.63	120.94

Table 5 presents Kemerer model performance measurements.

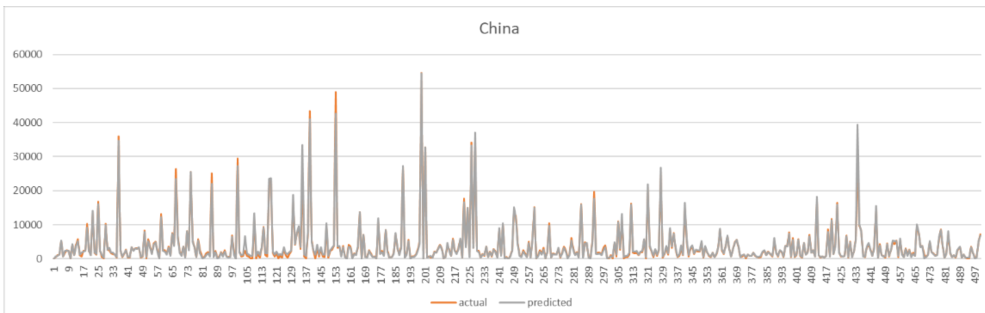

Figure 1. China Dataset Actual And Predicted Values

Figure 1 shows the efforts comparison for the actual and predicted by the model for in China dataset. Multilayer Perceptron algorithm was found to be the most successful to achieve best estimation. The Relief Att. Evaluation and Ranker methods were utilized during the analysis.

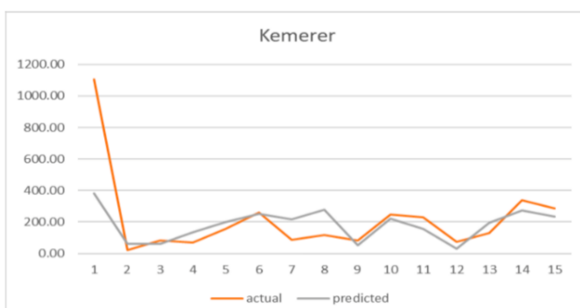

Figure 2. Kemerer Dataset Actual And Predicted Values

Figure 2 shows Kemerer dataset actual and predicted values. The KStar algorithm was found to be the most successful to achieve best estimation. The CFS and Genetic Algorithm methods were utilized during the analysis. The efforts comparison for the actual and predicted by the model are depicted.

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

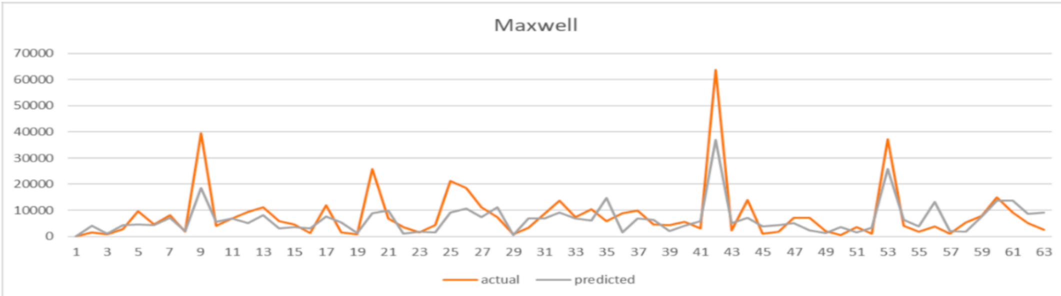

Figure 3. Maxwell Dataset Actual And Predicted Values

Figure 3 shows Maxwell dataset actual and predicted values. it was determined that the highest performance measurements as algorithms were obtained when KStar, SMOreg, MultilayerPerceptron and LinearRegression were used. It has been noted that models created using IBk, RandomTree and Bagging algorithms tend to give low results.

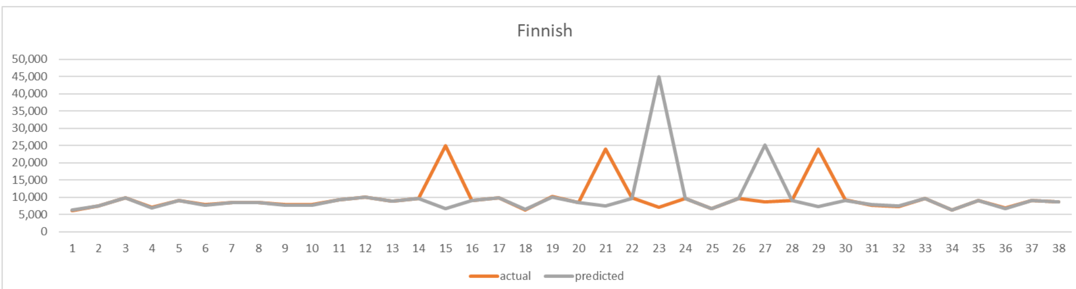

Figure 4. Finnish Dataset Actual And Predicted Values

Figure 4 shows Finnish dataset actual and predicted values. KStar algorithm was found to be the most successful to achieve best estimation. The ClassifierAttEval and Ranker methods were utilized during the analysis.

Table 6. Model Performance Results with Feature Selection

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Dataset	Orginal Feature Set	Model	FeatureSelection	Selected Feature Set	Correlation Coefficient
Finnish	9	KStar	CFS+ RandomSearch	5	0.9916
Finnish	9	RandomForest	CFS+ PSO	5	0.9942
Finnish	9	RandomForest	CFS+ GA	5	0.9942
Finnish	**9**	**KStar**	**ClassifierAttEval+ Ranker**	**6**	**0.9948**
Finnish	9	KStar	Corr. Att.Evaluation + Ranker	6	0.9912
Finnish	9	KStar	Relief. Att.Evaluation + Ranker	6	0.9916
China	19	SMOreg	CFS+ RandomSearch	7	0.9866
China	19	SMOreg	CFS+ PSO	9	0.9853
China	19	LinearRegression	CFS+ GA	10	0.9859
China	19	SMOreg	ClassifierAttEval+ Ranker	16	0.9887
China	19	MultilayerPerceptron	Corr. Att.Evaluation + Ranker	16	0.9912
China	**19**	**MultilayerPerceptron**	**Relief. Att.Evaluation + Ranker**	**16**	**0.9914**
Maxwell	27	LinearRegression	CFS+ RandomSearch	16	0.8354
Maxwell	27	K Star	CFS+ PSO	9	0.85
Maxwell	**27**	**K Star**	**CFS+ GA**	**20**	**0.8596**
Maxwell	27	M5p	ClassifierAttEval+ Ranker	24	0.8515
Maxwell	27	SMOreg	Corr. Att.Evaluation + Ranker	24	0.8336
Maxwell	27	M5p	Relief. Att.Evaluation + Ranker	24	0.8472
Maxwell	27	SMOreg	Relief. Att.Evaluation + Ranker	14	0.838
Kemerer	8	SMOreg	CFS+ RandomSearch	5	0.6795
Kemerer	**8**	**SMOreg**	**CFS+ PSO**	**5**	**0.6946**
Kemerer	**8**	**SMOreg**	**CFS+ GA**	**5**	**0.6946**
Kemerer	8	SMOreg	ClassifierAttEval+ Ranker	5	0.5405
Kemerer	8	RnndomTree	Corr. Att.Evaluation + Ranker	5	0.6658
Kemerer	8	MultilayerPerceptron	Relief. Att.Evaluation + Ranker	5	0.6295

Table 6 shows that even if the best result is obtained with a large number of feature sets, close to the best results can also be obtained with a less numbered feature set.

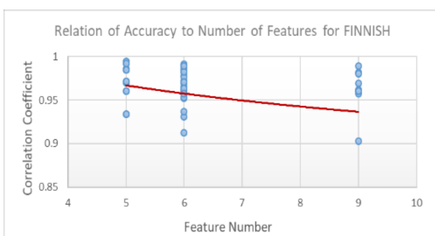

Figure 5. Relation of Accuracy To Number of Features For Finnish

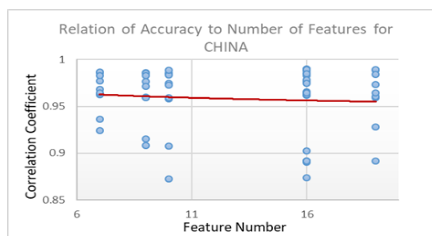

Figure 6. Relation of Accuracy To Number of Features For China

Figure 5 shows the relation of accuracy to number of features for Finnish and Figure 6 shows the relation of accuracy to number of features for China.

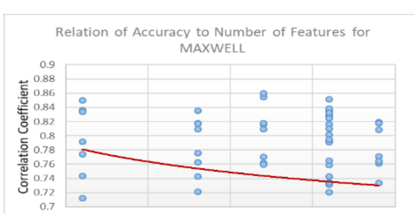

Figure 7.Relation of Accuracy To Number of Features For Maxwell

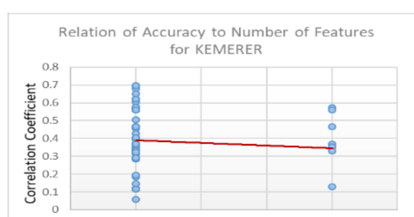

Figure 8. Relation of Accuracy To Number of Features For Kemerer

Figure 7 shows the relation of accuracy to number of features for Maxwell and Figure 8 shows the relation of accuracy to number of features for Kemerer.

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Table 7. Comparative Analysis of Gained Results with Literature

Dataset	Author(s)	Intelligent method	MMRE	PRED	Correlation Coefficient	MAE	RAE
China	(Rehal & Sharma, 2021)	SMOReg			0.9897	270.4561	7.3095
	(Kumar, Behera, Kumari, Nayak & Nail, 2020)	Spiking Neural Network	0.23				
		fuzzy c-means clustering-Functional Link Artificial Neural Networks	0.45				
		intuitionistic fuzzy c-means clustering-Functional Link Artificial Neural Networks	0.33				
		Long short-term memory	0.41				
		Output layer self-connection recurrent neural networks	0.32				
	Proposed Model	MLP & Relief Att.Eval. + Ranker	0.2655	0.0847	0.9914	370.1846	10.005
Finnish	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	1.7974	0.52			
		AnalogyBased Estimation - Extreme Learning Machines	2.3929	0.15			
		AnalogyBased Estimation - Artificial Neural Networks	2.124	0.32			
	Proposed Model	Kstar & ClassifierAttEval + Ranker	0.2521	0.0104	0.9948	0.0873	8.5274
Maxwell	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	1.1529	0.42			
		AnalogyBased Estimation - Extreme Learning Machines	4.2891	0.16			
		AnalogyBased Estimation - Artificial Neural Networks	4.4466	0.12			
	(Kumar, Behera, Kumari, Nayak & Nail, 2020)	Artificial Neural Network	1.32				
		Functional Link Artificial Neural Networks	0.42				
		Elman neural network	1.3748				
		Long short-term memory	0.37				
		Output layer self-connection recurrent neural networks	0.31				
	Proposed Model	Kstar & CFS + GA	0.7644	0.1274	0.8596	4078.324	64.913
Kemerer	(Benala & Bandarupalli, 2016)	AnalogyBased Estimation - Least Squares Support Vector Machin	0.66412	0.4			
		AnalogyBased Estimation - Extreme Learning Machines	1.8071	0.13			
		AnalogyBased Estimation - Artificial Neural Networks	2.0333	0.08			
	Proposed Model	SMOReg & Corr. Att.Evaluation + Ranker	0.5940	0.1289	0.7171	103.4371	64.273

Table 7 shows the best results obtained and the literature studies found with Artificial Neural Network methods applied to the same datasets and Machine Learning methods without feature selection are given. It is clear that high performance can be achieved with machine learning models by applying the low-cost and sustainable model feature selection targeted in the study. In the model outputs created with the relevant datasets, it was determined that the highest performance measurements as algorithms were obtained when KStar, SMOreg, MultilayerPerceptron and LinearRegression were used. It has been noted that models created using IBk, RandomTree and Bagging algorithms tend to give low results.

As a result, it seems that Machine Learning Based Approaches can be used as a high-performance method for software cost estimation and it is an open area for improvement.

9. Conclusion

The main goal of a successful software project is to produce software that will meet the expectations of the customer with a predetermined budget at a predetermined time. The failure of many software projects is due to the fact that the estimates made at the initial planning stage were not correct. For this reason, it can be said that the most basic and first project management activity in the success of a software project is the appropriate and effective allocation of necessary resources. In other words, it is critical to determine the resources that will be needed in the realization of the relevant project by making the planning on the right basis. Cost is the crux of these resources and is highly dependent on the effort within the project. In this case, estimating the effort needed is important in determining the cost.

For the software cost estimation process, which is a very important step in software project management, traditionally and predominantly manual input and expert opinion are still used today. However, these techniques cannot handle to estimate the cost of large and complex software. Therefore, to improve the software cost estimation process has aimed in this thesis. For this purpose, a machine learning- based approach has been adopted to make the software cost estimation process faster, more consistent and repeatable accurately. By leveraging machine learning techniques, the goal is to automate and optimize the software cost estimation process, reducing the reliance on manual and subjective judgements.

During the development of a machine learning-driven approach, the Finnish, Kemerer, China, and Maxwell datasets provided were utilized for software cost estimation. Models were constructed using the algorithms outlined and the validation technique employed was 10-fold cross-validation.

The study generally showed that machine learning-based models are applicable in software development effort estimation by quickly adapting to different data types, unlike traditional methods. Additionally, it is clear that better results can be obtained by applying feature selection to the data. It has been observed that the proposed hybrid feature selection methods can achieve better results compared to studies in the literature. In addition to all these, it has been observed that there are algorithms and feature selection methods that give the best results in different data sets, and it has been observed that the Kstar, SMOReg, LineaeRefression and MulitilayerPerceptron methods, which have achieved the best results with more data, are open to testing in order to reach a general conclusion.

As a result, it seems that Machine Learning Based Approaches can be used as a high-performance method for software cost estimation, and it is an open area for improvement. In future studies, similar methods can be studied with more and different datasets in order to generalize the obtained inferences and improve performance with different parameter values.

Contribution of Researchers

All researchers have contributed equally to writing this paper.

Conflicts of Interest

The authors declare no conflict of interest.

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 4, No. 1, July 2024
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

References

[1] LIU Qin and MINTRAM Robert C. (2005), "Preliminary Data Analysis", Software Quality Journal, Volume 13, pp. 91-115.

[2] SONER Taner (2014), Parametrik Tahmin Modellerinin Yazılım Projelerine Uygulanmasına Yönelik Bir Yazılım Paketinin Geliştirilmesi (Master's Thesis), Ankara University Institute of Science, Ankara.

[3] RIJWANI Poonam and JAIN Sonal (2016), "Enhanced Software Effort Estimation using Multi Layered Feed Forward Artificial Neural Network Technique", Procedia Computer Science, Volume 89, pp. 307-312.

[4] SAYYAD Shirabad and MENZIES Tim (2005), The PROMISE Repository of Software Engineering Databases, School of Information Technology and Engineering University of Ottawa, Canada, http://promise.site.uottawa.ca/SERepository, DoA. 05.03.2023.

[5] DANASINGH Asir Antony, BALAMURUGAN Suganya and EPIPHANY Jebamalar Leavline (2016), "Literature Review on Feature Selection Methods for High-Dimensional Data", International Journal of Computer Applications, Volume 136, No 10, pp. 9-17.

[6] DOKEROGLU Tansel, SEVINC Ender, KUCUKYILMAZ Tayfun and COSAR Ahmet (2019), "A survey on new generation metaheuristic algorithms", Computers & Industrial Engineering, Volume 137, pp. 106040, DOI: 10.1016/j.cie.2019.106040. DoA. 08.07.2023.

[7] ERGUZEL Turker, OZEKES Serhat, TAN Oguz and GULTEKIN Selahattin (2015), "Feature Selection and Classification of Electroencephalographic Signals an Artificial Neural Network and Genetic Algorithm Based Approach", Clinical EEG and Neuroscience, Volume 46, No 4, pp. 321-326.

[8] ORESKI Stjepan and ORESKI Goran (2014),"Genetic algorithm-based heuristic for feature selection in credit risk assessment", Expert systems with applications, Volume 41, No 4, pp. 2052-2064.

[9] WANG Yanqiu, CHEN Xiaowen, JIANG Wei, LI Li, LI Wei, YANG Lei, LIAO Mingzhi, LIAN Baofeng, LV Yingli, WANG Shiyuan, WANG Shuyuan and LI Xia (2011), "Predicting human microRNA precursors based on an optimized feature subset generated by GA–SVM", Genomics, Volume 98, No 2, pp. 73-78.

[10] YANG He, DU Qian and CHEN Genshe (2012), "Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Volume 5, No 2, pp. 544-554.

[11] DOKEROGLU Tansel, DENIZ Ayca and KIZILOZ Hakan Ezgi (2022), "A comprehensive survey on recent metaheuristics for feature selection", Neurocomputing, Volume 494, No 14, pp. 269-296.

[12] RUIZ Roberto, RIQUELME Jose C. and AGUILAR-RUIZ Jesus S. (2006), "Incremental wrapper-based gene selection from microarray data for cancer classification", Pattern Recognition, Volume 39, No 12, pp. 2383-2392.

[13] LIU Huan and YU Lei (2005), "Toward integrating feature selection algorithms for classification and clustering", IEEE Transactions on Knowledge and Data Engineering, Volume 17, No 4, pp. 491-502.