AdamOptimizer for the Optimisation of Use Case Points Estimation

Huynh Thai Hoc[1], Vo Van Hai[1], Ho Le Thi Kim Nhung[1]

[1] Tomas Bata University in Zlin, Faculty of Applied Informatics, Nad Stranemi 4511, Zlin 76001, Czech Republic
{huynh_thai,vo_van, lho}@utb.cz

Abstract. Use Case Points is considered to be one of the most popular methods to estimate the size of a developed software project. Many approaches have been proposed to optimise Use Case Points. The Algorithmic Optimisation Method uses the Multiple Least Squares method to improve the accuracy of Use Case Points by finding optimal coefficient regressions, based on the historical data. This paper aims to propose a new approach to optimise the Use Case Points method based on Gradient Descent with the support of the TensorFlow package. The significance of its purpose is to conduct a new approach that might lead to more accurate prediction than that of the Use Case Points and the Algorithmic Optimisation Method. As a result, this new approach outweighs both the Use Case Points and the Algorithmic Optimisation Methods.

Keywords: Software Effort Estimation, Algorithmic Optimisation Method, Use Case Points, Gradient Descent, Tensorflow, Adam, AdamOptimizer

1 Introduction

Software Effort Estimation (SEE) is considered to be the first main phase of the software development process [1, 2, 3, 4]. The purpose of this estimation is to estimate resources - including cost, participate in the steps of a project like development phases or maintain phases [2]. These are the key to understanding budgets or completion time or to maintain project activities.

However, SEE is not expected to be 100% accurate [2, 5]. Instead of finding the real size of the project, feasible solutions, or how to reduce the project risks, or the surprises the project might reveal, might be more acceptable. A feasible solution is to discover modifications so as to improve estimation accuracy based on the sample dataset.

Regression models are often one of the most popular ways used to improve effort estimation - (the response variable), by adding suitable weights to the corresponding technical factors - (predictor variables) [6]. Silhavy R. et al. [7] have proposed using the Algorithmic Optimisation Method (AOM) algorithm - a regression model, in order to optimise the Use Case Points (UCP) method, based on the tested dataset [7, 8, 9]. It is also a kind of supervised machine learning technique, and the TensorFlow package - the most popular machine learning package for Python, has a wide range of classifiers that make it useful for various applications, including regression models. This is an open-source library for numerical computation, implemented by the Google Brain team [10, 11]. Its name is derived from neural network operations that perform on tensors, and multidimensional data arrays [11]. It supports a list of optimisation algorithms in the tf.train module - for instance, the AdamOptimizer. The AdamOptimizer module is installed based on the Adam algorithm [10].

The idea of the Adam algorithm is to minimise loss functions based on the gradient descent concept [10, 12, 13]. The loss function is a function (Eq.1) that depicts the total square error function between a real value (Y_{real}), and an estimated value calculated from the $f(W)$ function [10, 12, 13]; where W is denoted as the learning weights vector. Gradient descent will be applied to minimise the loss function based on the repetitive following of the negative gradient [10], and the suitable learnable weights rule can be written with the rule in Eq.2; where α is the step-size - used to adjust weight matrix W, and ∇W denotes the directional change in W [10, 12, 13].

In 2004, Kingma & Ba proposed the Adam (Adaptive Moment Estimation) algorithm (Fig.1.) in order to resolve this problem [12, 13]. It is an optimisation algorithm based on adaptive estimates of the first and second moments of the gradients to the first-order gradient-based improvement of stochastic objective functions [12, 13]. Step size - also known as learning rate [10], is considered to be the significant property of Adam; it is invariant to the magnitude of the gradient and is approximately constrained by the step-size hyper-parameter [12, 13].

The Adam Algorithm

Input requires: Step-size (α); $\beta_1, \beta_2 \in [0,1)$; ϵ; $W_0: initial\ weights\ for\ training$
$m_0 \leftarrow 0$
$v_0 \leftarrow 0$
$t \leftarrow 0$
While W_t not converged do:
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_W L(W_{t-1})_t$ $(Get\ gradients\ at\ timestep\ t)$
    $m_t \leftarrow \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t$
    $v_t \leftarrow \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2$
    $\widehat{m_t} \leftarrow m_t / (1 - \beta_1^t)$
    $\widehat{v_t} \leftarrow v_t / (1 - \beta_2^t)$
    $W_t \leftarrow W_{t-1} - \alpha \times \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}}} + \epsilon$ $(Update\ unknown\ weights)$
End while
Return W_t (Resulting learnable weights)

Fig. 1. AdamOptimizer Algorithm Pseudo-code [12, 13]

This algorithm does not require an immobile goal since it works with sparse gradients and naturally performs a form of step-size annealing [12, 13]. It also requires less memory as well as computational efficiency [12, 13].

$$L(W) = \sum (f(W) - Y_{real})^2 \tag{1}$$

$$W = W - \alpha \nabla W \tag{2}$$

This paper presents a new approach to optimising the UCP method by finding the suitable weights $W = (\mathrm{w}_0, \mathrm{w}, \mathrm{w}_2)$ in the regression model (Eq.8), based on the AdamOptimizer module. The balance of the paper is structured as follows: Section 2 presents the research questions; Section 3 discusses the evaluation criteria; Section 4 proposes the new approach - the Adam-optimizer for the optimisation of the UCP, (AdamUCP); Section 5 presents the comparisons of the proposed algorithm with the UCP and the AOM; and Section 6 presents the conclusions.

2 Research Questions

This study evaluates the following research questions:

RQ1: Does the real size of the project fit with a new approach to optimising the UCP method? This question will answer whether the new approach fits the actual size or not; and a higher R^2 value means that the better the proposed approach, the better the fit with the observed data [14].

RQ2: Is this a better-proposed approach than the UCP or the AOM methods? The answer to this question is to determine the MMRE, the minimised SSR, and the maximised $PRED(0.25)$. In addition, the paired t-test was tested to decide the mean MRE measurement difference from the AdamUCP and the UCP/AOM [15].

H_o: $MRE_{AdamUCP} = MRE_{UCP/AOM}$: There is no estimation error difference between these methods.

H_1: $MRE_{UCP/AOM} > MRE_{AdamUCP}$: Estimation capability of project size by the AdamUCP might be more feasible than by the other methods. Put another way, the estimation error raised by the UCP or the AOM might be greater than the AdamUCP.

3 Evaluation Criteria

The Mean Magnitude of Relative Error - (MMRE) and the Prediction level - (PRED(x) are well-known evaluation criteria. All criteria were employed in order to examine the models´ prediction accuracy [7, 16, 4]. Both parameters are based on the quantity call Magnitude of Relative Error (MRE) method. The Residual Sum of Squares - (RSS), is a metric in regression analysis used to measure modeling error variation [7], it is also known as the Sum of Square Residuals - (SSR) or the Sum of Squared Estimate of Errors - (SSE) method. The coefficient of determination (R^2) is then used to assess the "goodness-of-fit" measure in the regression model [7, 16, 4]. After fitting the regression model, we use R^2 to determine how well the model fits the dataset. The equations are given as follows:

$$MRE = \frac{|\hat{y}_i - y_i|}{y_i} \tag{3}$$

$$MMRE = \frac{\sum_i^N MRE}{N} \tag{4}$$

$$\text{PRED}(x) = \frac{1}{N}\sum_{i}^{N}\begin{cases}1, if\ MRE_i\ \leq x\\ 0, otherwise\end{cases} \tag{5}$$

$$SSR = \sum_{i}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

$$R^2 = 1 - \left[\frac{\sum_{i}^{N}(y_i - \hat{y}_i)^2}{\sum_{i}^{N}(y_i - \bar{y})^2}\right] \tag{7}$$

Where N is the number of observations, y_i the known real value, $\hat{y}_i$ the predicted/estimated value, $\bar{y}$ the mean of known real value; and the x value is considered to be 0.25 - as recommended in many studies [7, 17, 18]. Some authors also use PRED(0.20) or PRED(0.30) with little differing results [18].

4 Our Proposed Method – AdamUCP

In publications [7, 8, 16], the mathematical equation used to estimate the size of the project is presented as Eq.8; where TUAW is Total Unadjusted Actor Weight [7, 8, 16]; UUCW is the Unadjusted Use Case Weight [7, 8, 16]; TCF is the Technical Complexity Factor [7, 8, 16]; and ECF is the Environmental Complexity Factor [7, 8, 16].

$$UCP = \text{TUAW x TCF x ECF} + \text{UUCW x TCF x ECF} \tag{8}$$

This paper intends to further enhance the accuracy of UCP based on the historical dataset [7]; so the UCP equation in Eq.8 is re-written (Eq.9) as a regression model - [3, 6] by adding $\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2$as the unknown parameters (or unknown weights).

$$UCP_{Predicted} = \mathrm{w}_0 + \mathrm{w}_1 \text{x } TUAW \text{ x TCF x ECF} + \mathrm{w}_2 \text{x } UUCW \text{ x TCF x ECF} \tag{9}$$

Where w_0 is the intercept and $\mathrm{w}_1, \mathrm{w}_2$ are the weights - (regression coefficients), of TUAW x TCF x ECF and UUCW x TCF x ECF, respectively.

As discussed in Section 1, the AdamOptimizer module will be used to find suitable unknown weights in Eq.9. It is a relatively robust algorithm and a good default to select [12]; it also requires little memory, calculates effectively, and fits problems that have large parameters or large datasets [12, 11]. As recommended from the Adam Algorithm [12], good default values: $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, will be selected as the default input values for the AdamOptimizer in this research paper. The suitable learnable parameters - $(\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2)$ in Eq.9, will be found through adopting the AdamOptimizer module with the historical dataset [7, 8, 9]. The AdamUCP is denoted as the equation to estimate the size of the project corresponding with $(\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2)$.

5 Experiment

5.1 Dataset Description

As mentioned in Section 4, the AdamUCP method will use Dataset1 for evaluation purposes - which is also used by Silhavy et al. in their publication [7] that includes data from 28 projects. These dataset characteristics are illustrated in Table 1.

Table 1. Dataset Characteristics

	Median Real_P20	Standard Deviation	Minimum Real_P20	Maximum Real_P20	N
Dataset	100.42	57.06	13.85	197.50	28

5.2 Experiment Design

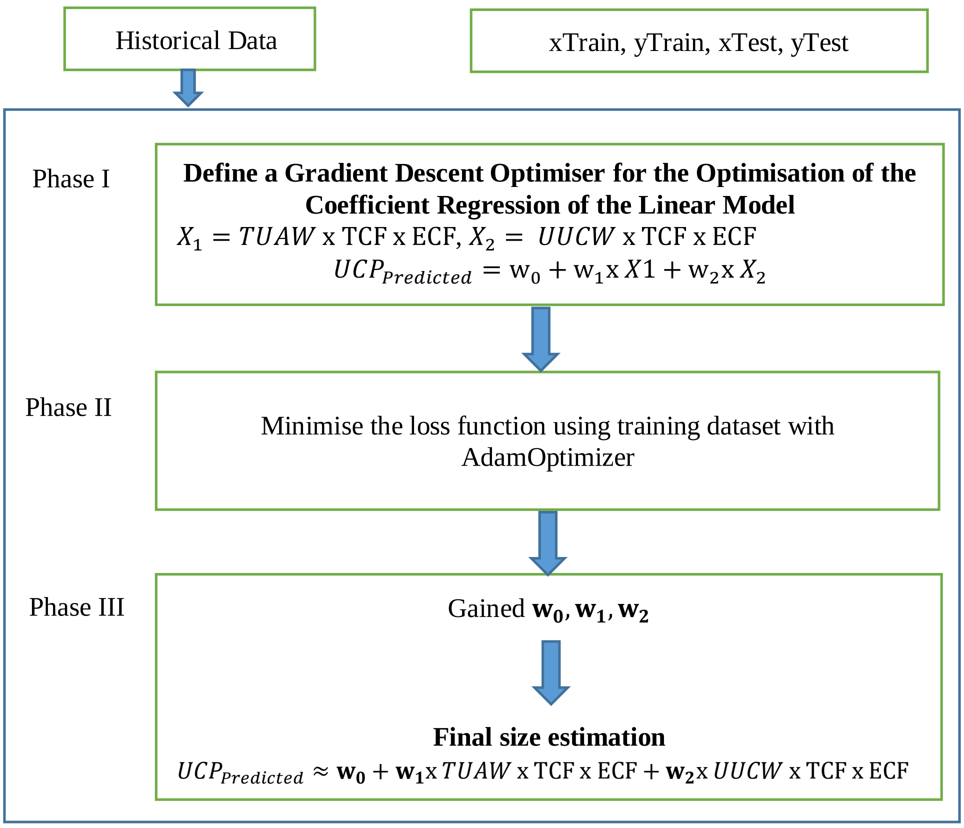

Fig. 2. The AdamUCP Flow Diagram

The AdamUCP method will be processed in three phases - shown in Fig. 2. The first phase will identify the gradient descent optimiser for the optimisation of Linear Model Coefficient Regressions; and then, the "Minimising Training Dataset Loss Functions with AdamOptimiser" will be handled in the ensuing step. In this way; $\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2$ - the optimal value, will finally be attained.

In Phase I, a placeholder for $X1 = TUAW$ x TCF x ECF $X2 = UUCW$ x TCF x ECF will be created for the input arguments; training weighting $W = (\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2)$ is thus transformed into variable objects with initial default values. In this way, the $UCP_{Predicted}(W)$ equation is given as Eq.10, identified as the linear regression model [3, 6]:

$$UCP_{Predicted}(\mathrm{W}) = \mathrm{w}_0 + \mathrm{w}_1 \mathrm{x}\, \mathrm{X}_1 + \mathrm{w}_2 \mathrm{x}\, \mathrm{X}_2 \tag{10}$$

The following step of the first phase will define the Loss Function - (Eq. 11), as well as a gradient descent optimiser - (Eq. 12). Often used for unknown weights training - including $\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2$.

$$loss(W) = REDUCE\_MEAN(SQUARE(UCP_{predicted}(W) - Real\_size) \tag{11}$$

$$train\ op = AdamOptimizer(learing\ rate).minimize(loss(W)) \tag{12}$$

The training model is executed in Phase II. The AdamOptimizer method will be used for training purposes based on historical data. The first issue - the training function, will be run in order to optimise Loss Consumption. This might have massive improvements in the first few epochs, and continues to decline so far [10]; it also might get worse if there were any changes.

5.3 Model Evaluation

In order to evaluate the accuracy of the AdamUCP, 10% of the projects will be used to evaluate the model; and 90% of the remaining projects used to train the AdamUCP model. All of these will be randomly selected. The testing dataset different from the training data. Due to the fact that there are only 28 projects in Dataset1 [7], if we were to increase the percentage of testing projects, the rest of the projects might not be good enough for training purposes - and might lead to inaccuracies in the learning process. The proportion of projects concentrated on testing should be increased if we are to have a large dataset.

There are three main experiments conducted to evaluate AdamUCP. Dataset1 - [7] is primarily normalised based on its mean and standard deviation [19]; outlier numbers are less affected by such normalisation - [20]. Table 2, presents the characteristics of the normalised Dataset1. The next step of 90% of projects will involve AdamUCP method training processes. In conclusion, 10% of the remaining project models will be evaluated.

Moreover, all Input Requirement Variables, including $\alpha, \epsilon, \beta_1, \beta_2$, use the default values mentioned in Section 4; and Initial Training Weights - (w_0, w_1, w_2) might be randomly assigned by the random Python's Function.

Table 2. Normalised Dataset1 Characteristics

	Median Real_P20	Standard Deviation	Minimum Real_P20	Maximum Real_P20	N
Dataset	-1.75	1.00	-1.52	1.70	28

Fig. 3 presents the loss curve of the AdamUCP method. It rapidly reaches the lowest point of approximately zero for several first iterations. Relying on the loss function concept [10], the training model should cease to obtain suitable unknown weights - w_0, w_1, w_2. As a result, the AdamUCP model obtained based on Dataset1 [7] is given by Eq.13; R^2 stands that the good fitness of the model is 0.6489; and MMR, MMRE, PRED (0.25), SSR are presented in Table 3.

$$UCP \approx 0.176866 + (0.250590 \times TUAW + 0.637755 \times UUCW) \times \mathrm{TCF} \times \mathrm{ECF} \quad (13)$$

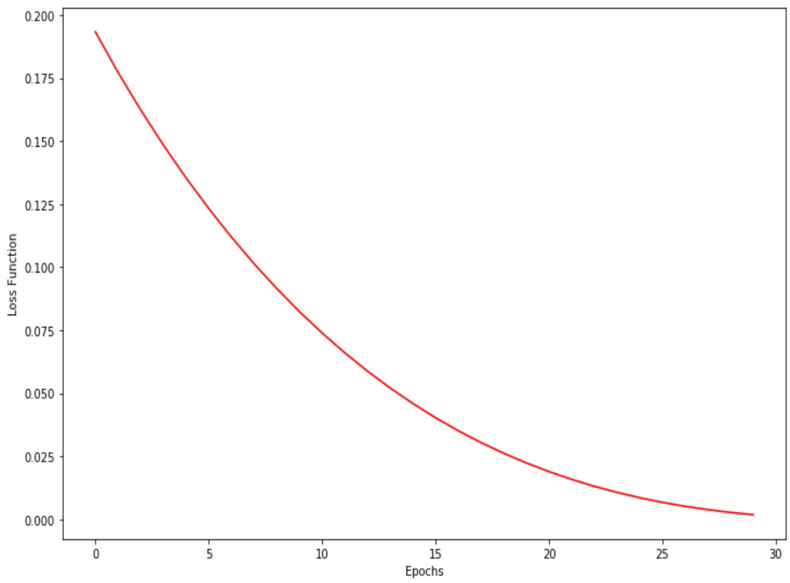

Fig. 3. The Loss Function Curve Gained from the AdamUCP

In conclusion, comparing all criteria - including MMRE, SSR, $PRED(0.25)$ shown in Table 3 and obtained from the AdamUCP method, are overall, better than those from the AOM and the UCP methods - matching the RQ2 requirement in Section 2. Thus, we might conclude that this might produce more accurate estimations.

Table 3. The Performance Estimation Comparison Method

	AdamUCP	**AOM**	**UCP**
MRE	15.6989	17.2002	32.3685
MMRE	0.5606	0.6142	1.1560
PRED(0.25)	0.4642	0.3929	0.1786
SSR	63,284.73	69,296.94	268,616.61
N	28	28	28

In addition, the paired t-test result of the hypothesis is given in Table 4. There is a significant difference in MRE between AdamUCP and UCP - (pvalue $= 0.0002 < 0.05$); AOM (pvalue $= 0.0274 < 0.05$). Based on the paired t-test results, one might reject the null hypothesis and accept the alternative hypothesis.

$MRE_{UCP/AOM}$, is associated with a larger mean than $MRE_{AdamUCP}$. This might reveal that AdamUCP is the best in the Estimation Accuracy Field.

Table 4. AdamUCP Hypothesis t-test Results

	Degree of Freedom	t-value	p-value
UCP	27	4.3580	0.0002
AOM	27	2.3315	0.0274

6 Conclusion

This paper presents research that proposes a new approach - (AdamUCP) to optimise project size estimation using the TensorFlow AdamOptimizer package. The AdamUCP performance was analysed and compared with the UCP and the AOM by adopting the evaluation criteria presented in Section 2 - including MMRE, SSE, $PRED(0.25)$. Two major questions were posed. Based on the analysed results in Section 5.3, we conclude the following:

(1) With $R^2 = 0.6489$, it is clear that the predicted size of projects obtained from the AdamUCP, overall, is close to the actual value [14]. In other words, the real size of the project is considered to slightly fit with the estimated size received from the AdamUCP.

(2) Table 3 presents the results of all criteria - (MRE, MMRE, $PRED(0.25)$, SSR) of the AdamUCP as compared with the UCP and the AOM. It is clear that the MRE, MMRE, and SSR obtained from the AdamUCP are all less than those obtained from the AOM and UCP; while the AdamUCP $PRED(0.25)$ attains the maximum when compared with the others. In addition, Table 4 demonstrates that the new approach might likely be better than other options. So, one might conclude that the AdamUCP model is more accurate than the UCP and the AOM models.

The disadvantage of this paper is that the training dataset has only 28 projects; so the number of projects using for training and testing is not diverse enough. More training datasets will be tested in the future.

Acknowledgment. This work was supported by the Faculty of Applied Informatics, Tomas Bata University in Zlín, under Project SV13202001020-PU30, Project IGA/CebiaTech/2020/001, and Project RVO/FAI/2020/002.

References

1. Muketha, Geoffrey, "A Review of Agile Software Effort Estimation Methods," *International Journal of Computer Applications Technology and Research*, vol. 5, no. 9, pp. 612-618, 2016.
2. Adam Trendowicz, Ross Jeffery, Software Project Effort Estimation, Springer, 2014.
3. Hoc H.T., Hai V.V., Nhung H.L.T.K., "A Review of the Regression Models Applicable to Software Project Effort Estimation," in *Silhavy R., Silhavy P., Prokopova Z. (eds) Computational Statistics and Mathematical Modeling Methods in Intelligent Systems. CoMeSySo 2019. Advances in Intelligent Systems and Computing, vol 1047. Springer, Cham*, 2019.
4. Hai V.V., Nhung H.L.T.K., Hoc H.T., "A Review of Software Effort Estimation by Using Functional Points Analysis," in *Silhavy R., Silhavy P., Prokopova Z. (eds) Computational Statistics and Mathematical Modeling Methods in Intelligent Systems. CoMeSySo 2019. Advances in Intelligent Systems and Computing, vol 1047. Springer, Cham*, 2019.
5. M. Jørgensen, D.I.K. Sjøberg, "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy," *Information and Software Technology 45*, p. 23–136, 2003.
6. Samprit Chatterjee, Ali S.Hadi, Regression Analysis By Example, Fifth ed., John Wiley & Sons, 2012.
7. Silhavy R., Silhavy P., Prokopova Z, "Algorithmic Optimisation Method for Improving Use Case Points Estimation," *PLoS ONE 10(11): e0141887*, 2015.
8. Silhavy P., Silhavy R., Prokopova Z, "Analysis and selection of a regression model for the Use Case Points method using a stepwise approach," *Journal of Systems and Software*, vol. 125, pp. 1-14, March 2017.
9. Silhavy P., Silhavy R., Prokopova Z, "Evaluating subset selection methods for use case points estimation," *Information and Software Technology*, vol. 97, pp. 1-9, 2018.
10. Ramsundar, B.;Zadeh, R. B., "TensorFlow for Deep Learning : From Linear Regression to Reinforcement Learning," O'Reilly Media., 2018, pp. 43-73.
11. Aaswad Sawant, Mayur Bhandari, Ravikymar Yadav, Rohan Yele, Mrs Sneha Bendale, "Brain cancer detection from MRI: A machine learning approach (TensorFlow),"

International Research Journal of Engineering and Technology (IRJET), vol. Volume 05, no. April 2018, p. 2092, 2018.

12. Diederik P Kingma, Jimmy Ba. Adam, "Adam: A method for stochastic," in *Proceedings of the 3rd International Conference on Learning Representations*, 2014.
13. Sebastian Bock, Josef Goppold, Martin Weiß, "An improvement of the convergence proof of the ADAM-Optimizer," in *Conference paper at OTH Clusterkonferenz* , 2018.
14 Moore, D. S., Notz, W. I, & Flinger, M. A., in *The basic practice of statistics*, 6th ed., New York, NY: W. H. Freeman and Company, 2013, p. 138.
15. Ross A., Willson V.L., "Paired Samples T-Test.," in *Basic and Advanced Statistical Tests.*, Rotterdam, SensePublishers, 2017.
16. Nhung H.L.T.K., Hoc H.T., Hai V.V., "A Review of Use Case-Based Development Effort Estimation Methods in the System Development Context," in *Silhavy R., Silhavy P., Prokopova Z. (eds) Intelligent Systems Applications in Software Engineering. CoMeSySo 2019. Advances in Intelligent Systems and Computing, vol 1046. Springer, Cham*, 2019.
17. Silhavy P., Silhavy R., Prokopova Z, "Stepwise Regression Clustering Method in Function Points Estimation. In: Silhavy R., Silhavy P., Prokopova Z. (eds)," *CoMeSySo 2018. Advances in Intelligent Systems and Computing*, vol. 859, pp. 333-340, 2019.
18. O Fedotova, L Teixeira, H Alvelos, "Software Effort Estimation with Multiple Linear Regression: Review and Practical Application," *J. Inf. Sci. Eng*, vol. 29, no. 5, pp. 925-945, 2013.
19. H. Yeh and M. Sun, "Coal Price Prediction Using Financial Indices," in *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Kaohsiung, Taiwan, 2019.
20. Berthold M.R., Borgelt C., Höppner F., Klawonn F, "Data Preparation," in *Guide to Intelligent Data Analysis*, London, Springer, 2010, pp. 115-143.