DOI: https://doi.org/10.15276/aait.09.2026.09
UDC 004.412:519.237.5

Estimating the number of lines of code of the latest releases of open-source applications using nonlinear regression models

Yurii O. Gunchenko[1)]
ORCID: https://orcid.org/0000-0003-4423-8267; gunchenko@onu.edu.ua. Scopus Author ID: 57193069126
Volodymyr I. Meshcheryakov[1)]
ORCID: https://orcid.org/0000-0003-0499-827X; meshcheryakovvi48@gmail.com. Scopus Author ID: 57192640885
Sergiy B. Prykhodko[2),3)]
ORCID: https://orcid.org/0000-0002-2325-018X; sergiy.prykhodko@nuos.edu.ua. Scopus Author ID: 55225622100
Alla L. Rachinska[1)]
ORCID: https://orcid.org/0000-0003-2430-9603; rachinskaya@onu.edu.ua. Scopus Author ID: 35422819700
[1)] Odesa I. I. Mechnikov National University, 2, Vsevoloda Zmiienka Str. Odesa, 65082, Ukraine
[2)] Admiral Makarov National University of Shipbuilding, 9, Heroes of Ukraine Ave. Mykolaiv, 54007, Ukraine
[3)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The problem of early estimating the number of lines of code of the latest releases of software, including open-source applications, is important because it directly affects the effort prediction for their development and subsequent improvement. The research aim is to build several regression models for early estimating the number of lines of code of the latest releases of open-source applications. The research object is the process of early estimating the number of lines of code of the latest releases of open-source applications. The research subject is the regression models for early estimating the number of lines of code of the latest releases of open-source applications. For early estimating the number of lines of code of the latest releases of open-source applications, the models, confidence, and prediction intervals of two nonlinear regressions with two predictors were constructed using the Box-Cox three-variate normalizing transformation and specialized techniques. These techniques, relying on multiple nonlinear regression analyses incorporating the use of multivariate normalizing transformations, account for the presence of outliers in multidimensional non-Gaussian data. In the paper, we built two nonlinear regression models for early estimating the number of lines of code of the latest releases of open-source applications that depend on two predictors: both the number of classes of their latest and first releases. The first model has good quality, but the second one can only be used for estimating the conditional mean, and it has poor quality for predicting the response as the dependent random variable. We have compared the quality of two constructed regression models and two linear support vector regressions. The quality of the above models is similar. An analysis has been carried out to compare the constructed models with nonlinear regression models that only depend on the number of classes for their current releases, which were built based on the Box-Cox bivariate transformation. Compared to such nonlinear regression models, the constructed models demonstrate a larger multiple coefficient of determination, a smaller value of the mean magnitude of relative error, a larger percentage of predictions that fall within 25 percent of the actual values, and narrower confidence and prediction intervals. The comparison results indicate better quality of the constructed models with two predictors. The prospects for further research may include the use of other data sets to construct the nonlinear regression models for early estimating the number of lines of code of the latest releases of open-source applications, for other restrictions on predictors.

Keywords: Regression model; estimation; lines of code; open-source application; normalizing transformation; Mardia criterion; outlier; Mahalanobis distance; test statistic; support vector regression; machine learning

***For citation*: Gunchenko Y. O., Mescheryakov V. I., Prykhodko S. B., Rachinska A. L. "Early estimating the number of lines of code of latest releases of open-source applications using nonlinear regression models". *Applied Aspects of Information Technology*. 2026; Vol.9 No.1: 122–133. DOI: https://doi.org/10.15276/aait.09.2026.09**

INTRODUCTION

Recently, more and more attention has been paid to the development and improvement of open-source software [1], [2], [3]. Therefore, the task of early estimation of the number of lines of code (NLOC) of different releases of open-source applications (apps) is important, as it directly affects the prediction of efforts for their development and subsequent modification [4], [5], [6].

Although a large number of various mathematical models have already been proposed for early estimation of the NLOC of software apps, they do not allow for the app release to be taken into account. In addition, the existing models take into account only the metrics of the current app release. And as studies [7], [8] show, various software metrics, including the NLOC, also depend on the release of the app. This requires the construction of mathematical models for estimating the NLOC of different releases of open-source software apps, including the latest releases that depend on the software metrics of their latest and previous releases.

LITERATURE REVIEW AND PROBLEM STATEMENT

Nowadays, despite the availability of a fairly large number of methods and models for estimating the NLOC of software at the early stage of development, increasing the accuracy of the NLOC estimation remains a pressing task [9], [10]. At the same time, various approaches are used for the corresponding estimation, which are usually based on metrics that can be defined before coding using either a conceptual data model [11], a use case diagram [12], a sequence model [13], or a class diagram [14], [15], [16], [17], [18].

Despite the widespread use of machine learning (ML) [19], [20], [21], [22], both linear [17], [23], [24] and nonlinear [25], [26], [27], [28], [29] regression analysis methods continue to be used for appropriate estimation. In [17], the authors constructed early software size estimation models based on four analysis-to-design adjustment factors (ADAF)-adjusted analysis class diagram metrics (i.e., ADAF-adjusted number of classes, ADAF-adjusted number of attributes, ADAF-adjusted number of methods, and ADAF-adjusted number of relationships) using stepwise multiple linear regression. Furthermore, in [17], the prediction accuracy of the best-performing proposed model is also compared with the model based on objective class points. According to [17], "The results of this comparison reveal that the proposed method reduces errors significantly (i.e., on average, 16% reduction in mean absolute residual and 24% reduction in mean squared error)". However, the use of linear regression models demands executing the assumption of the error distribution normality that is validated in particular cases [25]. This and other assumptions lead to the need to apply nonlinear regression models.

In [25], the authors applied their proposed technique to construct a nonlinear regression model for estimating the NLOC of open source PHP-based apps depending on three predictors: the number of classes, the average number of methods per class, and the sum of average afferent coupling and average efferent coupling per class. This model was built by the Johnson four-variate transformation for the S_B family. The above model demonstrates good prediction results since the values of the multiple coefficient of determination R^2, a mean magnitude of relative error MMRE, and prediction percentage at the level of magnitude of relative error of 0.25, PRED(0.25), are respectively 0.9812, 0.1753, and 0.750 for it. In [16] and [26], the authors constructed three nonlinear regression models for estimating the NLOC of open source Web apps created using the CodeIgniter and CakePHP frameworks, respectively, that depend on three predictors, the first two of which are the same as in [25], and an average of the Depth of Inheritance Tree (DIT) per class was chosen as the third predictor. These models from [16] were built based on the four-variate Box-Cox transformation, the decimal logarithm, and the univariate Box-Cox transformation, respectively. Although all three models have good values of quality indicators R^2, MMRE, and PRED(0.25), the models based on the four-variate Box-Cox transformation demonstrate the best results. Note, values of all predictors for the models from [16], [25], and [26] can be calculated from the class diagram.

In [27], the authors constructed a three-factor nonlinear regression model for estimating the KLOC (Kilo Lines Of Code) of open source Java-based apps by the decimal logarithm, like the model from [16], with the difference that instead of an average number of methods per class and DIT, “the total quantity of visible methods (VMQ)” and “average quantity of fields per class (aTFQ)” were used as factors. Even though the model from [27] has good values of R^2, MMRE, and, PRED(0.25), the use of the total visible methods quantity as a factor will lead to the problem of calculating the value of this metric from the class diagram because the accuracy of such calculations depends on the level of detail of the diagram itself (whether it is a conceptual diagram or a specification-level diagram).

In [28], like [25], the authors proposed to apply the Johnson multivariate transformation for the S_B family for building the three-factor nonlinear regression model for early estimation of KLOC in Java software. The model from [28] depends on “total quantity of classes (CLASS), total quantity of responses for class (RFC), and average value of public and protected methods per class (aVMQ)”. Although the authors [28] conclude they "obtained the three-factor nonlinear regression model for early estimation of KLOC in Java-software using appropriate techniques for constructing non-linear regression models on the basis of multivariate normalizing Johnson S_B family transformation", they do not provide its final expression, as that was done in [25].

In [29], like [16] and [26], the authors proposed to apply the multivariate Box-Cox transformation for building the five-factor nonlinear regression model for early estimation of KLOC in Java apps. The model from [29] depends on metrics “CLS, INFC, aVMQ, aTFQ, and aCBO”. Here, according to [28], CLS is the total number of actual classes, INFC is the total number of interfaces, aVMQ is “average visual methods quantity” (as in [28]), aTFQ is “average quantity of fields per class” (as in [27]), and aCBO is “average CBO”. Although the authors [28] conclude they “obtained five-factor nonlinear regression model on the basis of multivariate Box-Cox normalizing transformation”, they do not provide its final expression, as that was done, for example, in [16] and [26].

Though the above models demonstrate good prediction results of the NLOC estimation, none of them allow for taking into account the app release. Although, as studies [7] and [8] show, various software metrics, including the NLOC, also depend on the release of the app. In addition, all existing models depend on the project metrics of the current app releases. And to provide an earlier estimate of the size of the latest app release, it is desirable to have mathematical models that depend on the software metrics of previous releases. This requires the construction of mathematical models, including nonlinear regression ones, to estimate the NLOC of different releases of open-source apps, including the latest ones, that depend on the software metrics of their latest and previous releases.

RESEARCH AIM AND OBJECTIVES

The **research aim** is to build several regression models for early estimating the NLOC of the latest releases of open-source apps that depend on the software metrics of the first ones. To achieve the above aim, the following **research objectives** must be performed.

1) We need to construct regression models for early NLOC estimation of the latest releases of open-source apps that depend on the software metrics of the latest and first releases.

2) We need to check the quality of the constructed regression models for early NLOC estimation of the latest releases of open-source apps.

3) We need to compare the quality of the constructed models and nonlinear regression models that only depend on the number of classes for their current releases.

The research object is the process of early estimating the NLOC of the latest releases of open-source apps. The research subject is the regression models for early NLOC estimation of the latest releases of open-source apps.

FORMULATION OF THE PROBLEM

Suppose given the original sample as the three-dimensional non-Gaussian data set: the NLOC of the latest releases of open-source apps *Y*, the total number of classes of open-source apps of the latest X_1 and first releases X_2, respectively. Suppose that there are three-variate normalizing transformation of a non-Gaussian random vector $\mathbf{P}=\{Y,X_1,X_2\}^T$ to a Gaussian random vector $\mathbf{T}=\{Z_Y,Z_1,Z_2\}^T$ is given by

$$\mathbf{T}=\boldsymbol{\psi}(\mathbf{P}) \qquad (1)$$

and the inverse transformation for (1)

$$\mathbf{P}=\boldsymbol{\psi}^{-1}(\mathbf{T}). \qquad (2)$$

It is required to build the nonlinear regression models in the form $Y=Y(X_1,X_2,\varepsilon)$ based on the transformations (1) and (2). Here, Z_Y, Z_1, and Z_2 are the components of a Gaussian random vector **T**, which are defined by the normalizing transformation (1) using the variables *Y*, X_1, and X_2, respectively; ε is a Gaussian random variable that defines residuals, $\varepsilon \sim N(0,\sigma_\varepsilon)$, where σ_ε is the standard deviation of ε.

MATERIALS AND RESEARCH METHODS

To construct nonlinear regression models for early NLOC estimation of the latest releases of open-source apps that depend on two predictors, we apply the technique from [25]. This technique to build nonlinear regression models based on multivariate normalizing transformations and prediction intervals comprises six steps. The first step involves normalizing multivariate non-Gaussian data through transformation (1). To do this, like [16], we apply the three-variate Box-Cox transformation with components

$$Z_j = x(\lambda_j)=\begin{cases}\left(X_j^{\lambda_j}-1\right)/\lambda_j, & \text{if} \quad \lambda_j \neq 0;\\ \ln\left(X_j\right), & \text{if} \quad \lambda_j = 0.\end{cases} \qquad (3)$$

Here Z_j is a Gaussian variable; λ_j is a parameter of the Box-Cox transformation, $j=1,2$. The variable Z_Y is defined analogously (3) with the only difference that instead of Z_j, X_j, and λ_j should be put respectively Z_Y, *Y*, and λ_Y.

In the second step, according to [25], we determine whether one three-dimensional data point of a multivariate non-Gaussian data set is a three-dimensional outlier. If there is a three-dimensional outlier in a three-variate non-Gaussian data set, then we cut off the one and go to step 1, continue.

To determine whether one data point of a three-variate non-Gaussian data set is a three-dimensional outlier, we use the statistical technique based on the normalizing transformations and the squared Mahalanobis distance (SMD) as in [25].

In the third step, we construct a linear regression model for normalized data in the form

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \varepsilon \,. \qquad (4)$$

In the fourth step, according to [25], we test the normality of the residuals distribution in (4). If the residuals distribution in (4) is not Gaussian, then we discard the three-variate data point for which the modulus of the residual ε in the model (4) is the maximum, and go to step 1; otherwise, continue.

According to [25], the nonlinear regression model using the transformation (1) and (2) for the linear regression model for normalized data (4) is constructed in the fifth step. For the three-variate Box-Cox transformation with components (3), the nonlinear regression model has the form [16]

$$Y = \left[\hat{\lambda}_Y \left(\hat{Z}_Y + \varepsilon\right) + 1\right]^{1/\hat{\lambda}_Y}, \qquad (5)$$

where $\hat{Z}_Y$ is a prediction result by the linear regression equation for normalized data, $\hat{Z}_Y = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2$, which are transformed by the three-variate Box-Cox transformation with components (3).

Finally, in the sixth step, according to [25], we define the prediction interval of nonlinear regression and determine whether one or more values of the response Y (dependent random variable) are outliers (their values are outside the prediction interval). If there are outliers in the data for the nonlinear regression model (5), then we discard these and go to step 1; otherwise, we complete constructing the nonlinear regression model. We define the prediction interval of nonlinear regression as in [25]

$$\psi_Y^{-1}\left(\hat{Z}_Y \pm t_{\alpha/2,\nu} S_{Z_Y} \left\{1 + \frac{1}{N} + \left(\mathbf{z}_X^{+}\right)^T \mathbf{S}_Z^{-1} \left(\mathbf{z}_X^{+}\right)\right\}^{1/2}\right), (6)$$

where $t_{\alpha/2,\nu}$ is a student's *t*-distribution quantile with $\alpha/2$ significance level and ν freedom degrees; $\nu = N - k - 1$; k is the number of independent variables (in our case, k is 2); $\mathbf{z}_X^{+}$ is a vector with components $Z_{1_i} - \bar{Z}_1, \ \ldots, \ Z_{k_i} - \bar{Z}_k$ for *i*-row; $\bar{Z}_j = \frac{1}{N}\sum_{i=1}^{N} Z_{j_i}$, $S_{Z_Y}^2 = \frac{1}{\nu}\sum_{i=1}^{N}\left(Z_{Y_i} - \hat{Z}_{Y_i}\right)^2$, $j = 1, 2, \ldots, k$, $\nu = N - k - 1$; $\mathbf{S}_Z$ is a $k \times k$ matrix

$$\mathbf{S}_Z = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \cdots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \cdots & S_{Z_2 Z_k} \\ \cdots & \cdots & \cdots & \cdots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \cdots & S_{Z_k Z_k} \end{pmatrix}. \qquad (7)$$

In (7), $S_{Z_q Z_r} = \sum_{i=1}^{N}\left[Z_{q_i} - \bar{Z}_q\right]\left[Z_{r_i} - \bar{Z}_r\right], \ q, r = 1, 2, \ldots, k$.

According to [25], we used the above formulas to construct nonlinear regression models for early NLOC estimation of the latest releases of open-source apps that depend on two predictors.

RESEARCH RESULTS

We constructed nonlinear regression models for early NLOC estimation of open-source apps by the above technique from the software metrics of 40 open-source apps from [8]. We focused on the following variables: the NLOC of the latest releases of open-source apps Y, and the total number of classes of open-source apps of the latest X_1 and first releases X_2, respectively.

We checked the three-dimensional data from [8] for three-variate outliers. Before analyzing the three-dimensional data from [8] for three-variate outliers, we assessed the normality of the three-variate data. This preliminary check was essential, as common statistical methods, including multivariate outlier detection based on the squared Mahalanobis distance (SMD), are designed to identify outliers assuming a Gaussian distribution. We applied a multivariate normality test proposed by Mardia and based on measures of the multivariate skewness β_1 and kurtosis β_2. According to this test, the distribution of three-dimensional data from [8] is not Gaussian since the test statistic for three-variate skewness $N\beta_1/6$, which equals 164.79, is greater than the quantile of the Chi-Square distribution, which is 25.19 for 10 degrees of freedom and for a significance level of 0.005, and also the test statistic for their three-variate kurtosis β_2, which equals 39.97, is greater than the value of the Gaussian distribution quantile, which is 19.46 for 15 mean, 3.00 variance, and 0.005 significance level.

Because, as in [25], to detect multivariate outliers in the three-dimensional non-Gaussian data from [8], we used the statistical technique based on the multivariate normalizing transformations and the SMD for normalized data. To normalize the data from [8], the three-variate Box-Cox transformation with components (3) was applied. The parameter estimates of the three-variate Box-Cox transformation for the data from [8] are calculated by the maximum likelihood method and are $\hat{\lambda}_Y = 0.0495399$, $\hat{\lambda}_1 = 0.0601632$, $\hat{\lambda}_2 = 0.0185304$.

Next, we assessed the normality of the three-variate normalized data by the Mardia test. According to this test, the distribution of three-dimensional normalized data is approximately Gaussian since the test statistic for their three-variate skewness $N\beta_1/6$, which equals 20.40, is less than the quantile of the Chi-Square distribution, which is 25.19 for 10 degrees of freedom and for a significance level of 0.005, and also the test statistic for their three-variate kurtosis β_2, which equals 14.69, is less than the value of the Gaussian distribution quantile, which is 19.46 for 15 mean, 3.00 variance, and 0.005 significance level.

There are no three-variate outliers among 40 rows of three-dimensional non-Gaussian data since their SMD values for normalized data (SMD$_Z$) do not exceed 12.84, which is the quantile of the Chi-Square distribution, applicable for 3 degrees of freedom and for a significance level of 0.005.

According to [25], in the third step, we build the linear regression model (4) for 40 rows of normalized data from [8]. The estimates $\hat{b}_0$, $\hat{b}_1$, and $\hat{b}_2$ equal 4.39740, 1.36876, and -0.183294, respectively. The estimate $\hat{\sigma}_\varepsilon$ of standard deviation of ε is 0.7554.

In the fourth step, according to [25], we test the normality of the residuals distribution in (4). Since the residuals distribution in (4) is not Gaussian, we discard the three-variate data point for which the modulus of the residual ε in the model (4) is the maximum. This point is row 1 in Table 1. After that, we go to step 1 for the second iteration of building the nonlinear regression model (5). There were eight such iterations in total and nine outliers in the process of building the model (5), which are listed in Table 1.

Table 1. **Data that were outliers in the process of building the first model (5)**

No.	Y	X_1	X_2	LB	UB
1	25953	116	85	1490.2	100318.9
2	18086	103	97	497.4	66438.7
3	13687	540	490	3655.7	173438.9
4	127207	851	465	13494.3	287034.2
5	27648	247	222	2552.8	121348.9
6	155734	917	261	26992.0	617630.1
7	607	26	19	0.0	22121.9
8	30347	307	187	7434.4	202699.9
9	59569	620	534	4381.2	188687.4

Source: **compiled by the authors**

Finally, the first model in form (5) was built based on 31 data points. The parameter estimates of this model (model 1) are listed in Table 2.

The resulting outliers when building the first model indicate that this model cannot describe the data from Table 1. That is why we additionally constructed the second model in form (5) based on nine data points from *Table 1* that were outliers in the process of building the first model (5). There were no outliers in the process of building the second model in form (5). The parameter estimates of the second model (model 2) are listed in Table 2.

Table 2. **The parameter estimates of the models in form (5)**

Esti-mates	Model 1	Model 2	Model 3	Model 4
$\hat{\lambda}_Y$	0.075335	0.21354	0.079335	0.30775
$\hat{\lambda}_1$	0.098819	-0.21383	0.102703	0.38009
$\hat{\lambda}_2$	0.022159	-0.03635	-	-
$\hat{b}_0$	4.92759	-85.9527	4.90872	13.3855
$\hat{b}_1$	1.35474	63.2100	1.25144	3.02638
$\hat{b}_2$	-0.18422	-16.7796	-	-
$\hat{\sigma}_\varepsilon$	0.4163	4.5943	0.4460	16.1373

Source: **compiled by the authors**

Also, the lower (LB) and upper (UB) bounds of the prediction interval of the second nonlinear regression are listed in Table 1. These bound values indicate large interval widths. We also calculated the LB and UB bounds of the prediction interval of the first nonlinear regression, whose values indicate significantly smaller interval widths compared to the corresponding widths of the second regression. We defined the prediction interval of both nonlinear regressions using (6).

To assess the predictive accuracy of the nonlinear regression models (5), we utilized standard metrics, namely R^2, MMRE, and PRED(0.25). The acceptable values of MMRE and PRED(0.25) are not more than 0.25 and not less than 0.75, respectively. To calculate the values of MMRE and PRED(0.25), we used well-known formulas

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i\ ; \tag{8}$$

$$PRED(0{,}25) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 & if\ MRE_i \le 0{,}25\\ 0 & otherwise\end{cases} \tag{9}$$

In (8) and (9), MRE_i is the Magnitude of Relative Error for data point i

$$MRE_i = \left|\left(Y_i - \hat{Y}_i\right)/Y_i\right|.$$

For the first model in form (5) with the above parameter estimates, predicated upon the three-variate Box-Cox transformation applied to the dataset of the 31 apps, the computed values for R^2, MMRE, and PRED(0.25) are listed in Table 3 and indicate good model quality. For the second model in form (5) with the above parameter estimates, predicated upon the three-variate Box-Cox transformation applied to the dataset of nine apps, the computed values for R^2, MMRE, and PRED(0.25) are listed in Table 3. They indicate good model quality for only estimating the conditional mean and poor model quality for predicting the dependent random variable.

Table 3. **The computed values of R^2, MMRE, and PRED(0.25) for models 1 and 2**

Metrics	Training sets		Test datasets	
	Model 1	Model 2	Model 1	Model 2
R^2	0.9541	0.8369	0.9358	0.3159
MMRE	0.1677	0.4933	0.2301	0.6233
PRED	0.7742	0.2222	0.6000	0.3333

Source: **compiled by the authors**

Note that the data on which we built the two models are training sets. To avoid the problem of overfitting models [30], the predictive accuracy of two models in form (5) should be checked on the test datasets, the data of which were not used to build the above models. That we did next.

The test datasets were obtained from [7] around the corresponding metrics of different releases of three popular open-source apps in Java: FreeMind, jEdit, and TuxGuitar. Table 4 contains the test datasets.

Also in Table 4, the SMD values for normalized data (SMD_Z) by the three-variate Box-Cox transformation with parameter estimates, which were used for constructing the first model (model 1) and the second model (model 2), are listed. These values indicate which data from Table 4 we can use to test the nonlinear regression models in form (5).

Note that only those data for which the SMD_Z values are less than 12.84, and the values of predictors belong to fixed intervals, can be included in the test data sets. Predictor 1 for the first model must be in the interval from 8 to 4397, and for the second model, from 26 to 917. Predictor 2 for the first model must be in the interval from 5 to 3697, and for the second model, from 19 to 534. Then, taking into account the above, for testing the first model, we can use only the first five rows (rows 1-5) of data from Table 4, and for testing the second model, six (rows 2, 6-10).

Table 4. **The test datasets**

No.	Latest/pre-vious app releases	Y	X_1 / X_2	SMD_Z model 1 / 2
1	FreeMind 0.7.1/0.0.3	18988	199/53	8.49/ 34.52
2	FreeMind 0.9.0Beta17/0.7.1	56752	577/199	9.33/ 9.14
3	TuxGuitar 1.3.0/1.2	91481	1234/736	3.04/ 2.32
4	TuxGuitar 1.5.2/1.2	108495	1618/736	3.01/ 1.84
5	TuxGuitar 1.5.2/1.3.0	108495	1618/1234	2.42/ 8.79
6	jEdit 5.5.0/2.3pre2	151672	952/322	32.99/ 3.32
7	jEdit 5.5.0/2.6final	151672	952/453	33.58/ 2.85
8	jEdit 5.5.0/3.0final	151672	952/282	33.03/ 4.90
9	FreeMind 0.8.0/0.7.1	84199	718/199	16.39/ 11.01
10	FreeMind 0.8.1/0.7.1	84089	718/199	16.34/ 11.02

Source: **compiled by the authors**

For the first model in form (5), predicated upon the three-variate Box-Cox transformation applied to the test dataset of five apps (rows 1-5) from Table 4, the computed values for R^2, MMRE, and PRED(0.25) are listed in Table 3. These values indicate satisfactory model quality. Only the PRED(0.25) value is 20 % worse than the acceptable. For the second model in form (5), predicated upon the three-variate Box-Cox transformation applied to the test dataset of six rows (rows 2, 6-10) of data from Table 4, the computed values for R^2, MMRE, and PRED(0.25) are listed in Table 3. These values indicate satisfactory model quality for only estimating the conditional mean and poor model quality for predicting the response as the dependent random variable.

We also compared the prediction results $\hat{Y}$ using the two models for data row 2 from Table 4. In this case, the $\hat{Y}$ values for the first and the second models are 34332.8 and 127400.1, respectively. In this case, the MRE values of $\hat{Y}$ for the first and the second models are 0.395 and 1.245, respectively. Also, we compared the confidence intervals of $\hat{Y}$ values for data row 2 from Table 4. In this case, the confidence intervals of $\hat{Y}$ based on the first and the second models are from 30420.7 to 38705.7 and from 43224.4 to 92181.1, respectively. These MRE values and the confidence intervals indicate a greater accuracy of estimation of $\hat{Y}$ by the first model compared to the second.

To compare the constructed models with a well-known ML algorithm, we used support vector regression (SVR) [19]. We built two linear SVRs based on the same training sets, which were used for constructing models 1 and 2. For two linear SVRs (SVR 1 and SVR 2), the computed values of R^2, MMRE, and PRED(0.25) are listed in Table 5.

Table 5. **The computed values of R^2, MMRE, and PRED(0.25) for SVR 1 and SVR 2**

Metrics	***Training sets***		***Test datasets***	
	SVR 1	*SVR 2*	*SVR 1*	*SVR 2*
R^2	0.9576	0.9267	0.9486	0.8602
MMRE	0.2231	0.8525	0.2057	0.2880
PRED	0.5806	0.4444	0.6000	0.5000

Source: **compiled by the authors**

The values from Table 5 for SVR 1 indicate satisfactory model quality. Only the PRED(0.25) values are 23% and 20% worse than the acceptable values for the training and test sets, respectively. The comparison of model 1 and SVR 1 indicates better quality of model 1 due to the MMRE and PRED(0.25) values for the training set. The values from Table 5 for SVR 2 indicate satisfactory model quality for only estimating the conditional mean, as for model 2.

We also built the linear SVR based on the whole training set (40 data points) without spliting on two clusters. For this model, the computed values of R^2, MMRE, and PRED(0.25) for the training set are 0.8237, 0.3339, and 0.4500, respectively. For this model, the computed values of R^2, MMRE, and PRED(0.25) for the test set from Table 4 are 0.5122, 0.3762, and 0.3000, respectively. These values indicate satisfactory model quality for only estimating the conditional mean and poor model quality for predicting the response as the dependent random variable. In addition, in our case, these results underline the need to split the dataset into parts (clusters), which we have done above.

We also compared the constructed models with two nonlinear regression models that only depend on one predictor X_1. These two nonlinear regression models were also built in form (5) based on the same data: 31 and 9 data rows, respectively. For the model in form (5) with one predictor (model 3) based on 31 data rows, the computed values for R^2, MMRE, and PRED(0.25) are 0.9556, 0.1755, and 0.7097, respectively. These values indicate satisfactory model quality since the PRED(0.25) value is 5.4% worse than the acceptable. Note, the above values are worse than R^2, MMRE, and PRED(0.25) values for the first model with two predictors based on 31 data rows (see Table 3).

For another model in form (5) with one predictor (model 4), predicated upon the two-variate Box-Cox transformation applied to nine data rows from Table 1, the computed values for R^2, MMRE, and PRED(0.25) are 0.7646, 0.9254, and 0.4444, respectively. These values indicate satisfactory model quality for only estimating the conditional mean and poor model quality for predicting the dependent random variable. Note, the above values are worse than R^2 and MMRE values for the first model with two predictors based on nine data rows (see *Table 3*). The parameter estimates of models 3 and 4 are listed in Table 2.

The above comparison indicates the model quality of the models with two predictors is better than that of the models with one predictor. In other words, the NLOC estimation accuracy of the latest releases of open-source apps by the models that depend on two predictors is higher than that of the models that only depend on one predictor X_1.

DISCUSSION OF RESULTS

Utilizing the modified technique [25] based on multivariate normalizing transformations, we employ the three-variate Box-Cox transformation to construct two nonlinear regression models for the early NLOC estimation of the latest releases of open-source apps that depend on two predictors: the total number of classes of open-source apps of the latest and first releases. This technique is chosen due to four reasons. Firstly, the three-variate distribution of the data is not Gaussian according to the Mardia multivariate normality test based on multivariate skewness and kurtosis. That is why, to determine whether a data point of a three-variate non-Gaussian data set is a three-dimensional outlier, we use the statistical technique based on normalizing transformations and the squared Mahalanobis distance for the normalized data as in [25]. Secondly, the assumption of normality of the error distribution for two linear regression models is not validated. That leads to the need to build nonlinear regression models. Thirdly, there are outliers due to residuals in the process of constructing the nonlinear regression model. And finally, in the process of building a nonlinear regression model, outliers arise due to the fact that the response values are outside the prediction interval.

To detect outliers in three-variate data and define the prediction interval of the nonlinear regression, we use respectively 0.005 and 0.05 significance levels as the appointed ones usually, although these values may be discussed.

Note that in the process of building the first model based on 40 data points, there were nine outliers, from which the second data cluster was formed, like [31], using which the second model was built. There were no outliers in the process of constructing the second model.

The advantages of the proposed models (5) include the possibility of the early NLOC estimation of the latest releases of open-source apps using the values of two metrics (the total number of classes of open-source apps of the latest and first releases). The first metric (the total number of classes of open-source apps of the latest release) can be measured from the class diagram, as is usually the case for other well-known models. But the second metric (the total number of classes of open-source apps of the first release) value is known as a rule. Note, using these metrics as model predictors allows us to take into account both the increase and decrease in the number of classes in a new app release compared with the previous one in the early NLOC estimate, which cannot be done in existing models.

The disadvantages of the two proposed models include, first of all, the fact that both models, the first and second, are based on small data samples: 31 and 9 data points, respectively. This primarily applies to the second model. That is why the second model has satisfactory quality for only estimating the conditional mean and poor model quality for predicting the response as the dependent random variable. We recommend using the second model when the SMD_Z value for the first model parameters is greater than 12.84. Unlike the second model, the first model has good quality, as evidenced by the calculated values of metrics R^2, MMRE, and PRED(0.25).

The proposed models (5) are limited to the early NLOC estimation of open-source apps for which there are the following restrictions on predictors: the interval for X_1 is from 8 to 4397 and the interval for X_2 is from 5 to 3697 for the first model, the interval for X_1 is from 26 to 917 and the interval for X_2 is from 19 to 534 for the second model. In addition, the proposed models (5), compared to the well-known multifactor models, are limited to the number of predictors. As we know, the early NLOC estimates may depend on other metrics, such as, for example, the weighted methods per class and DIT. That is why in the future, we will need to research the influence of other metrics on the early NLOC estimates.

CONCLUSIONS

1. Utilizing the modified technique based on multivariate normalizing transformations, we have employed the three-variate Box-Cox transformation to construct two nonlinear regression models for the early NLOC estimation of the latest releases of open-source apps that depend on the total number of classes of open-source apps of the latest and first releases, respectively. Using these metrics as model predictors allows us to take into account both the increase and decrease in the number of classes in a new app release compared with the previous one in the early NLOC estimate, which cannot be done in existing models.

2. We have checked the quality of the constructed regression models for early NLOC estimation of the latest releases of open-source apps. Unlike the second model, the first model has good quality, as evidenced by the calculated values of metrics R^2, MMRE, and PRED(0.25). The second model has satisfactory quality for only estimating the conditional NLOC mean and poor quality for predicting the NLOC response as the dependent random variable.

The width of the confidence and prediction intervals for the first model is less than that of the second one. That also indicates better quality of the first model compared to the second one. We recommend using the second model only in the case when the SMD_Z value for the first model parameters is greater than 12.84 and when the values of the predictors are within the specified restrictions (X_1 is from 26 to 917 and X_2 is from 19 to 534).

3. We have compared the quality of two constructed regression models (models 1 and 2) and two linear SVRs (SVR 1 and SVR 2) based on the same training sets. The quality of models 1 and SVR 1 is similar, with the only difference being that model 1 has better MMRE and PRED(0.25) values for the training set (by 33 and 25 percent, respectively). The quality of models 2 and SVR 2 is similar. Both models have satisfactory quality for only estimating the conditional NLOC mean and poor quality for predicting the NLOC response as the dependent random variable.

4. Also, we have checked the quality of the constructed regression models with two predictors and the nonlinear regression models that only depend on the number of classes for their current releases. The comparison of the above models indicates that the quality of the models with two predictors is better than that of the models with one predictor.

5. In the future, to validate strong conclusions derived from data analysis based on the nonlinear regression models for the early NLOC estimation of the latest releases, further research needs to be carried out for other software metrics and data sets. Also, in the future, to construct nonlinear regression models for the early NLOC estimation of the latest releases, it is planned to apply other multivariate normalizing transformations, for example, the Johnson transformation.

REFERENCES

1. Himansh, M. & Manikandan, V. M. "A statistical study and analysis to identify the importance of open-source software". In *International Conference on Innovative Trends in Information Technology (ICITIIT)*. 2022. p. 1–6, https://www.scopus.com/pages/publications/85128660804. DOI: https://doi.org/10.1109/ICITIIT54346.2022.9744176.

2. Madaehoh, A. & Senivongse, T. "OSS-AQM: An open-source software quality model for automated quality measurement". In the *International Conference on Data and Software Engineering (ICoDSE)*. 2022. p. 126–131, https://www.scopus.com/pages/publications/85145779848. DOI: https://doi.org/10.1109/ICoDSE56892.2022.9972135.

3. Haider, S., Khalil, W., Al-Shamayleh, A.S., Akhunzada, A. & Gani, A. "Risk factors and practices for the development of open source software from developers' perspective". *IEEE Access*. 2023; 11, 63333–63350, https://www.scopus.com/pages/publications/85153335021.
DOI: https://doi.org/10.1109/ACCESS.2023.3267048.

4. Brar, P. & Nandal, D. "A systematic literature review of machine learning techniques for software effort estimation models". In *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)*. 2022. p. 494–499, https://www.scopus.com/pages/publications/85141694707. DOI: https://doi.org/10.1109/CCiCT56684.2022.00093.

5. Kumar, S., Arora, M., Sakshi & Chopra, S. "A review of effort estimation in agile software development using machine learning techniques". In *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2022. p. 416–422,https://www.scopus.com/pages/publications/85146489004. DOI: https://doi.org/10.1109/ICIRCA54612.2022.9985542.

6. Rahman, M., Sarwar, H., Kader, M. A., Gonçalves, T. & Tin, T. T. "Review and empirical analysis of machine learning-based software effort estimation". *IEEE Access*. 2024; 12: 85661–85680, https://www.scopus.com/pages/publications/85194075870. DOI: https://doi.org/10.1109/ACCESS.2024.3404879.

7. Molnar, A. J., Neamţu, A. & Motogna, S. "Evaluation of software product quality metrics". In *E. Damiani, G. Spanoudakis, & L. Maciaszek (eds). Communications in Computer and Information Science: Evaluation of Novel Approaches to Software Engineering*. 2020; 1172: 163–187, https://www.scopus.com/pages/publications/85080917482. DOI: https://doi.org/10.1007/978-3-030-40223-5_8.

8. Gradišnik, M., Beranič, T. & Karakatič, S. "Impact of historical software metric changes in predicting future maintainability trends in open-source software development". *Applied Sciences*. 2020; 10 (13): 4624, https://www.scopus.com/pages/publications/85087877670. DOI: https://doi.org/10.3390/app10134624.

9. Daud, M. & Malik, A. A. "Improving the accuracy of early software size estimation using analysis-to-design adjustment factors (ADAFs)". *IEEE Access*. 2021; 9, 81986–81999, https://www.scopus.com/pages/publications/85107333843. DOI: https://doi.org/10.1109/ACCESS.2021.3085752.

10. Dewi, R. S., Araynawa, T. K., Prasanna, F. M., et al. "Improving software size estimation using data complexity (Case study: Research and community service monitoring apps)". In the *11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. 2024. p. 315–319, https://www.scopus.com/pages/publications/85214644467.
DOI: https://doi.org/10.1109/EECSI63442.2024.10776530.

11. Dewi, R. S., Zahrah, F. A., Nugraha, D. A., Prabowo, P. S., Safitri, A. & Jayadi, P. "Predicting software size based on conceptual data model (Case study: Shrimp pond system management)". In *2024 International Conference on Electrical Engineering and Computer Science (ICECOS)*. 2024. p. 175–178, https://www.scopus.com/pages/publications/85215302488.
DOI: https://doi.org/10.1109/ICECOS63900.2024.10791154.

12. Nassif, A. B., AbuTalib, M. & Capretz, L. F. "Software effort estimation from Use Case diagrams using nonlinear regression analysis". In *IEEE Canadian Conference on Electrical and Computer*

Engineering. 2020. p. 1–4, https://www.scopus.com/pages/publications/85097838683. DOI: https://doi.org/10.1109/CCECE47787.2020.9255712.

13. Hussain, I. & Malik, A. A. "Determining the utility of use case points and class points in early software size estimation". In *2023 18th International Conference on Emerging Technologies (ICET)*. 2023. p. 171–175, https://www.scopus.com/pages/publications/85183321717.
DOI: https://doi.org/10.1109/ICET59753.2023.10374977.

14. Tan H. B. K., Zhao, Y. & Zhang, H. "Conceptual data model-based software size estimation for information systems". *Transactions on Software Engineering and Methodology*. 2009; 19 (2): 1–37, https://www.scopus.com/pages/publications/70350214668. DOI: https://doi.org/10.1145/1571629.1571630.

15. Kiewkanya, M. & Surak, S. "Constructing C++ software size estimation model from class diagram". In *13th International Joint Conference on Computer Science and Software Engineering*. 2016. p. 1–6, https://www.scopus.com/pages/publications/85006857342.
DOI: https://doi.org/10.1109/JCSSE.2016.7748880.

16. Prykhodko, S. B., Shutko, I. S. & Prykhodko, A. S. "Early size estimation of web apps created using Codeigniter framework by nonlinear regression models". *Radio-Electronic and Computer Systems*. 2022; 103 (3): 84-94, https://www.scopus.com/pages/publications/85139983764. DOI: https://doi.org/10.32620/reks.2022.3.06.

17. Daud, M. & Malik, A. A. "Construction and validation of early software size estimation models based on ADAF-adjusted ACD metrics". *The Computer Journal*. 2023; 66 (9): 2123–2137, https://www.scopus.com/pages/publications/85174173242. DOI: https://doi.org/10.1093/comjnl/bxac065.

18. Daud, M. & Malik, A. A. "Exploring the impact of security-based non-functional requirements on early software size estimation". *Computing and Informatics*. 2024; 43 (5): 1234–1255, https://www.scopus.com/pages/publications/85209942535. DOI: https://doi.org/10.31577/cai_2024_5_1234.

19. Manisha & Rishi, R. "Early size estimation using machine learning". In *8th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2021. p. 757–762. DOI: https://ieeexplore.ieee.org/document/9441329.

20. Molla, Y. S., Alemneh, E. & Yimer, S. T. "COSMIC-based early software size estimation using deep learning and domain-specific BERT". *IEEE Access*. 2025; 13: 28463–28475, https://www.scopus.com/pages/publications/85217911784. DOI: https://doi.org/10.1109/ACCESS.2025.3540548.

21. Rajput, Y., Razi, M. & Sharma, A. K. "A comparative analysis of different machine learning techniques used in software effort estimation". In *2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*. 2025. p. 385–393, https://www.scopus.com/pages/publications/105002686758.
DOI: https://doi.org/10.1109/CICTN64563.2025.10932574.

22. Vifert Jenuben, D. V. & Rithani, M. "Enhancing software effort estimation with machine and deep learning strategies". In *2025 7th International Conference on Intelligent Sustainable Systems (ICISS)*. 2025. p. 1217–1222, https://www.scopus.com/pages/publications/105012160791.
DOI: https://doi.org/10.1109/ICISS63372.2025.11076303.

23. Nhung, H. L. T. K., Hai, V. V., Silhavy, R., Prokopova, Z. & Silhavy, P. "Parametric software effort estimation based on optimizing correction factors and multiple linear regression". *IEEE Access*. 2022; 10: 2963–2986, https://www.scopus.com/pages/publications/85122291307.
DOI: https://doi.org/10.1109/ACCESS.2021.3139183.

24. Sahoo, P., Behera, D. K., Mohanty, J. R. & Kumar Dash, C. S. "Effort estimation of software products by using UML sequence models with regression analysis". In *OITS International Conference on Information Technology (OCIT)*. 2022. p. 97–101, https://www.scopus.com/pages/publications/85150271198. DOI: https://doi.org/10.1109/OCIT56763.2022.00028.

25. Prykhodko, S. & Prykhodko, N. "A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations". *CEUR Workshop Proceedings, Germany*. 2022; 3179: 156–166, https://www.scopus.com/pages/publications/85136303475.

26. Prykhodko, S., Prykhodko, A. & Shutko, I. "Estimating the size of web apps created using the CakePHP framework by nonlinear regression models with three predictors". In *IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*. 2021. p. 333–336, https://www.scopus.com/pages/publications/85124793057. DOI: https://doi.org/10.1109/CSIT52700.2021.9648680.

27. Oriekhov, O., Farionova, T., Chernova, L. S., Chernova, L. & Vorona, M. “Nonlinear regression models for software size estimation of data science and machine learning Java-applications”. *CEUR Workshop Proceedings*, *Germany*. 2024; 3709: 54–66, https://www.scopus.com/pages/publications/85197217634.

28. Oriekhov, O., Farionova, T. & Chernova, L. “Three-factor nonlinear regression model of estimating the size of java-software”. *CEUR Workshop Proceedings*. Germany. 2024; 3790: 506–518, https://www.scopus.com/pages/publications/85207834181.

29. Oriekhov, O., Farionova, T., Chernova, L. & Vorona, M. “A five-factor nonlinear regression model for Java applications size estimation”. *CEUR Workshop Proceedings*. Germany. 2025; 4023: 1–10, https://www.scopus.com/pages/publications/105017566451.

30. Frost, J. “Overfitting regression models: Problems, detection, and avoidance”. *Statistics By Jim*. – Available from: https://statisticsbyjim.com/regression/overfitting-regression-models. – [Accessed: 23 Nov 2025].

31. Prykhodko, S. & Prykhodko, N. “Building nonlinear regression models for estimating the number of clusters and their initial centroids”. In *IEEE 18th International conference on computer sciences and information technologies (CSIT)*. 2023. p. 1–4, https://www.scopus.com/pages/publications/85179850953. DOI: https://doi.org/10.1109/CSIT61576.2023.10324095.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship, or other interests, which could influence the research and its results presented in this article

Received 07.12.2025
Received after revision 12.02.2026
Accepted 18.022.2026

DOI: https://doi.org/10.15276/aait.09.2026.09
УДК 004.412:519.237.5

Оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом за допомогою нелінійних регресійних моделей

Гунченко Юрій Олександрович[1)]
ORCID: https://orcid.org/0000-0003-4423-8267; gunchenko@onu.edu.ua. Scopus Author ID: 57193069126
Мещеряков Володимир Іванович[1)]
ORCID: https://orcid.org/0000-0003-0499-827X; meshcheryakovvi48@gmail.com. Scopus Author ID: 57192640885
Приходько Сергій Борисович[2),3)]
ORCID: https://orcid.org/0000-0002-2325-018X; sergiy.prykhodko@nuos.edu.ua. Scopus Author ID: 55225622100
Рачинська Алла Леонідівна[1)]
ORCID: https://orcid.org/0000-0003-2430-9603; rachinskaya@onu.edu.ua. Scopus Author ID: 35422819700
[1)] Одеський національний університет імені І. І. Мечникова, вул. Змієнка Всеволода, 2. Одеса, 65082, Україна
[2)] Національний університет кораблебудування імені адмірала Макарова. пр. Героїв України, 9. Миколаїв, 54007, Україна
[3)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Проблема раннього оцінювання кількості рядків коду останніх релізів програмного забезпечення, включаючи застосунки з відкритим кодом, є важливою, оскільки вона безпосередньо впливає на прогнозування зусиль для їх розробки та подальшого вдосконалення. Метою дослідження є побудова кількох регресійних моделей для раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом. Об'єктом дослідження є процес раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом. Предметом дослідження є регресійні моделі для раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом. Для раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом, моделі, довірчі інтервали та інтервали прогнозування двох нелінійних регресій з двома предикторами були побудовані за допомогою тривимірного нормалізуючого перетворення Бокса-Кокса та спеціалізованих методів. Ці методи, які спираються на множинний нелінійний регресійний аналіз, що включає використання багатовимірних нормалізуючих перетворень, враховують наявність викидів у багатовимірних негаусівських даних. У статті ми побудували дві нелінійні регресійні моделі для раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом, які залежать від двох предикторів: кількості класів їх останнього та першого релізів. Перша модель має хорошу якість, але друга може бути використана лише для оцінювання умовного середнього, і вона має низьку якість для прогнозування реакції як залежної випадкової величини. Ми порівняли якість двох побудованих регресійних моделей та двох лінійних регресій з опорними векторами. Якість вищезгаданих моделей є подібною. Було проведено аналіз для порівняння побудованих моделей з нелінійними регресійними моделями, які залежать лише від кількості класів для їхніх поточних релізів, та побудованими на основі двовимірного перетворення Бокса-Кокса. Порівняно з такими нелінійними регресійними моделями, побудовані моделі демонструють більший коефіцієнт детермінації, менше значення середньої величини відносної похибки, більший відсоток прогнозів, що потрапляють у межах 25 відсотків від фактичних значень, та вужчі довірчі інтервали та інтервали прогнозування. Результати порівняння вказують на кращу якість побудованих моделей з двома предикторами. Перспективи подальших досліджень можуть включати використання інших наборів даних для побудови нелінійних регресійних моделей для раннього оцінювання кількості рядків коду останніх релізів програмних застосунків з відкритим кодом, для інших обмежень на предиктори.

Ключові слова: регресійна модель; оцінювання; рядки коду; застосунок з відкритим кодом; нормалізуюче перетворення; критерій Мардії; викид; відстань Махаланобіса; тестова статистика; регресія опорних векторів; машинне навчання

ABOUT THE AUTHORS

Yurii O. Gunchenko - Doctor of Engineering Sciences, Professor, Head of Department of Computer Systems and Technologies, Odesa I. I. Mechnikov National University, 2, Vsevoloda Zmiienka Str. Odesa, 65082, Ukraine
ORCID: https://orcid.org/0000-0003-4423-8267; gunchenko@onu.edu.ua. Scopus Author ID: 57193069126
***Research field*:** Mathematical modeling; data science

Гунченко Юрій Олександрович – доктор технічних наук, професор, завідувач кафедри Комп'ютерних систем та технологій. Одеський національний університет імені І. І. Мечникова, вул. Всеволода Змієнка, 2. Одеса, 65082, Україна

Volodymyr I. Meshcheryakov - Doctor of Engineering Sciences, Professor of Department of Information Technologies, Odesa I. I. Mechnikov National University, 2, Vsevoloda Zmiienka Str. Odesa, 65082, Ukraine
ORCID: https://orcid.org/0000-0003-0499-827X; meshcheryakovvi48@gmail.com. Scopus Author ID: 57192640885
***Research field*:** Information Technologies; Mathematical modeling; data science; machine learning

Мещеряков Володимир Іванович – доктор технічних наук, професор кафедри Інформаційних технологій. Одеський національний університет імені І. І. Мечникова, вул. Всеволода Змієнка, 2. Одеса, 65082, Україна

Sergiy B. Prykhodko - Doctor of Engineering Sciences, Professor, Head of Department of Software for Automated Systems. Admiral Makarov National University of Shipbuilding. 9, Heroes of Ukraine Ave. Mykolaiv, 54007, Ukraine
ORCID: https://orcid.org/0000-0002-2325-018X; sergiy.prykhodko@nuos.edu.ua. Scopus Author ID: 55225622100
***Research field*:** Mathematical modeling; computer simulation; stochastic processes; multivariate statistical analysis; system identification; parameter estimation, nonlinear stochastic differential equations; software metrics estimation

Приходько Сергій Борисович - доктор технічних наук, професор, завідувач кафедри Програмного забезпечення автоматизованих систем. Національний університет кораблебудування імені адмірала Макарова, пр.Героїв України, 9. Миколаїв, 54007, Україна

Alla L. Rachinska - PhD, Associate Professor, Head of Department of Mechanics, Automation and Information Technologies, Odesa I. I. Mechnikov National University. 2, Vsevoloda Zmiienka Str. Odesa, 65082, Ukraine
ORCID: 0000-0003-2430-9603. rachinskaya@onu.edu.ua. Scopus Author ID: 35422819700
***Research field*:** Information Technologies; Mathematical modeling

Рачинська Алла Леонідівна - кандидат фізіко-математичних наук, доцент, завідувач кафедри Механіки, автоматизації та інформаційних технологій. Одеський національний університет імені І. І. Мечникова, вул. Всеволода Змієнка, 2. Одеса, 65082, Україна