https://doi.org/10.31891/2307-5732-2025-349-71
УДК 004.412:519.237.5

ПРИХОДЬКО СЕРГІЙ
Національний університет кораблебудування імені адмірала Макарова
https://orcid.org/0000-0002-2325-018X
e-mail: sergiy.prykhodko@nuos.edu.ua
ШУТКО ІВАН
Національний університет кораблебудування імені адмірала Макарова
https://orcid.org/0000-0001-8584-113X
e-mail: ishutko@gmail.com

РАННЄ ОЦІНЮВАННЯ КІЛЬКОСТІ РЯДКІВ КОДУ ВЕБ-ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ PHP ФРЕЙМВОРКІВ

В роботі запропоновано для раннього оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою двох відомих PHP фреймворків (CakePHP та Codeigniter), використовувати регресійні моделі в залежності від трьох факторів (кількість класів, середня кількість методів на клас та метрика DIT на рівні застосунку), значення яких можуть бути визначені за діаграмою класів.

Ключові слова: оцінювання, рядки коду, веб-застосунок, PHP фреймворк, регресійна модель, перетворення Бокса-Кокса.

PRYKHODKO SERHII
SHUTKO IVAN
Admiral Makarov National University of Shipbuilding

RESEARCH OF STRUCTURAL AND MECHANICAL PROPERTIES OF MEAT AS AN OBJECT OF PROCESSING IN MEAT COMMINUTOR

The problem of early estimation of lines of code count in software projects holds significant importance, as it directly influences the prediction of software development effort, including web applications created using the well-known PHP frameworks, as the CakePHP and Codeigniter. The object of the study is the process of early estimating the lines of code count of web applications created using the CakePHP and Codeigniter frameworks. The subject of the study is the regression models for early estimating the lines of code count of web applications created using the CakePHP and Codeigniter frameworks. The goal of the work is to build some regression models with three factors for early estimating the lines of code count of web applications created using the CakePHP and Codeigniter frameworks depending on the factors that can be found in the class diagram. In the work, we built one nonlinear and two linear regression models for early estimating the lines of code count of web applications created using the CakePHP and Codeigniter frameworks depending on three factors: the number of classes, the average number of methods per class, metric DIT (Depth of Inheritance Tree) on the application level. These factors were chosen for two reasons. First, the values of these factors can be found from the class diagram, and, second, there is no problem of multicollinearity for them. The parameter estimates of the obtained models were found by the method of least squares. The above models are constructed based on splitting the four-dimensional data set (the actual size in thousands of lines of code, the number of classes, the average number of methods per class, the DIT metric on the application level) into two clusters. This data of metrics for 88 open-source web applications created using the CakePHP and Codeigniter frameworks was obtained using the PhpMetrics tool. These applications are hosted on the GitHub platform. The comparison of the constructed models with the other non-linear regression models is performed. The constructed regression models, in comparison with existing non-linear ones, allow us to describe all the four-dimensional data on which they were built.

Keywords: estimation, lines of code, web application, PHP framework, regression model, Box-Cox transformation.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Задача раннього оцінювання кількості рядків коду програмного забезпечення (ПЗ), включаючи і веб-застосунки, є важливою, оскільки ця інформація використовується для визначення зусиль створення ПЗ за допомогою різноманітних методів та моделей [1, 2]. Вказане оцінювання є необхідною умовою для планування проекту ПЗ та складання його бюджету [3, 4].

В той же час для розробки веб-застосунків широко використовують PHP фреймворки, які прискорюють створення цих застосунків. Серед таких відомих фреймворків широкою популярністю користуються CakePHP (https://cakephp.org/) та Codeigniter (https://www.codeigniter.com/). Хоча для раннього оцінювання кількості строк коду веб-застосунків, що створюються за допомогою зазначених фреймворків CakePHP та Codeigniter, вже були запропоновані відповідні нелінійні регресійні моделі - [5] та [6], але вони не дозволяють описувати частину даних, що були відкинуті як аномальні при їх побудові. Це потребує використання декількох математичних моделей для раннього оцінювання кількості строк коду веб-застосунків, що створюються за допомогою фреймворків CakePHP та Codeigniter, які дозволяли би описувати весь набір даних. Для побудови зазначених моделей ми скористалися результатами роботи [7], за якими здійснюється розбиття чотиривимірного набору даних (фактичний розмір у тисячах рядків коду; кількість класів; середня кількість методів на клас; метрика DIT (Depth of Inheritance Tree) на рівні застосунку) на два кластери.

Аналіз досліджень та публікацій

В наш час не зважаючи на існування достатньо великої кількості методів та моделей для оцінювання кількості строк коду ПЗ на ранній стадії розробки, підвищення точності оцінювання залишається актуальною задачею [8, 9]. При цьому для відповідного оцінювання використовують різноманітні підходи, які, як правило, базуються на метриках, що можуть бути визначені до кодування із

застосуванням або концептуальної моделі даних [10], або діаграми варіантів використання [11], або моделі послідовності [12], або діаграми класів [5, 6].

Не зважаючи на широке застосування в наші дні машинного навчання [1, 2, 13], методи як лінійного [14], так і нелінійного регресійного аналізу [5, 6, 11] продовжують використовуватися для відповідного оцінювання. Так, на сьогодні відомі нелінійні регресійні моделі для раннього оцінювання кількості рядків коду веб-застосунків, які створюються за допомогою певних PHP фреймворків, у тому числі і CakePHP [5] та Codeigniter [6]. Але розроблені в роботах [5] та [6] трифакторні нелінійні регресійні моделі на основі чотиривимірного перетворення Бокса-Кокса хоча і мають добрі показники якості для набору даних, за яким вони були створені, однак ці моделі не дозволяють описувати частину даних, що були відкинути як аномальні при їх побудові. Це призводить до необхідності побудови декількох моделей, які надавали би можливість здійснювати раннє оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою фреймворків CakePHP та Codeigniter, для всього набору даних.

Формулювання цілей статті

Метою роботи є: побудова декількох регресійних моделей для раннього оцінювання кількості рядків коду веб-застосунків, які створюються за допомогою фреймворків CakePHP та Codeigniter, в залежності від факторів, що можуть бути знайдені за діаграмою класів.

Виклад основного матеріалу

Спочатку було здійснено вибір даних метрик веб-застосунків, що створювалися на основі фреймворків CakePHP та Codeigniter. Ми взяли дані з робот [5] та [6] для 88 веб-застосунків за такими метриками: розмір веб-застосунку Y у тисячах рядків коду, кількість класів X_1, середня кількість методів на клас X_2 та метрика DIT на рівні застосунку X_3. Ми вибрали наведені вище фактори X_1, X_2 та X_3 тому, що, по-перше, їх значення можна отримати з діаграми класів, а, по-друге, між ними відсутня мультиколінеарність.

Далі згідно з результатами роботи [7] обрані дані були розбиті на два кластери. До кластеру 1 увійшло 48 точок даних, а до кластеру 2 – 40 точок даних зазначених вище метрик. Значення зазначених метрик для 48 веб-застосунків (кластер 1) наведені в таблиці 1. При чому в таблиці 1 в рядках 1-32 розташовані дані веб-застосунків, що створювалися на основі фреймворку CakePHP, а в рядках 33-48 – Codeigniter. Значення метрик для 40 веб-застосунків (кластер 2) наведені в таблиці 2. При чому в таблиці 2 в рядках 1-34 розташовані дані веб-застосунків, що створювалися на основі фреймворку Codeigniter, а в рядках 35-40 – CakePHP.

Таблиця 1

Дані з кластеру 1

№	Y	X_1	X_2	X_3	SMD_Z	№	Y	X_1	X_2	X_3	SMD_Z
1	0,448	4	4,75	1,4	5,33	25	12,433	66	6,52	1,79	6,61
2	4,345	42	4,19	1,81	1,62	26	3,5	40	4,18	1,63	1,24
3	0,212	1	7	2	7,45	27	1,494	22	3,41	1,35	5,48
4	1,149	14	3,86	1,73	0,47	28	3,735	49	3,57	1,7	2,78
5	0,477	8	2,63	2,2	2,18	29	2,954	45	3,13	2,13	6,16
6	0,124	2	2,5	2	2,77	30	4,029	58	3,38	1,54	4,49
7	0,365	4	4	1,5	3,41	31	7,846	90	3,86	1,71	5,06
8	2,332	23	4	1,79	0,67	32	2,717	60	2,2	2,11	14,11
9	0,948	10	4,3	1,78	0,64	33	2,28	32	0,31	1,71	4,63
10	1,826	20	4,15	1,6	0,24	34	2,265	31	0,19	1,75	5,30
11	0,131	3	2	1,67	3,46	35	32,39	97	13,31	1,26	3,52
12	2,227	23	4,22	1,46	1,12	36	33,368	101	13,45	1,25	3,69
13	0,906	5	7,8	1,6	3,35	37	28,333	83	13,41	1,24	3,64
14	2,681	23	4,78	1,71	0,43	38	2,361	35	0,23	1,77	4,05
15	0,142	3	3	2	4,28	39	0,939	10	3,8	1,67	0,62
16	1,717	26	2,54	1,44	3,08	40	28,653	83	13,4	1,29	3,45
17	3,486	29	3,62	2	4,30	41	32,778	100	13,1	1,26	3,44
18	0,358	2	3,5	2	9,81	42	29,738	94	12,84	1,24	3,39
19	1,538	11	5,82	1,8	1,27	43	33,373	97	13,75	1,26	3,83
20	0,882	9	3,56	1,6	1,45	44	34,23	106	12,86	1,31	3,47
21	0,471	5	3,8	2	1,91	45	2,097	30	0,2	1,75	4,56
22	0,521	2	8,5	3	13,69	46	2,347	34	0,18	1,77	4,45
23	0,349	8	2,25	1,38	8,14	47	31,365	98	12,89	1,25	3,36
24	3,567	23	6,83	1,24	6,11	48	32,836	101	13,19	1,26	3,47

За критерієм Мардіа чотиривимірний розподіл даних для 48 веб-застосунків (48 строк даних з таблиці 1) відхиляється від гаусівського тому, що значення тестової статистики для багатовимірної

асиметрі більше за квантиль $\chi^2_{20,0,005}$ з 20 ступенями свободи та 0,005 рівнем значущості, а саме: 48,27>40,00. Тому для визначення наявності чотиривимірних викидів у даних таблиці 1 ми використовували критерій на основі квадрату відстані Махаланобіса для нормалізованих даних. Для нормалізації даних, як і в [7], ми використовували чотиривимірне перетворення Бокса-Кокса з компонентами

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

де Z_j – це j-та випадкова змінна з розподілом Гаусу (нормалізована змінна), λ_j – параметр перетворення Бокса-Кокса для змінної j, $j = 1,2,3$. Компонента Z_Y визначається аналогічно (1) з тою різницею, що замість Z_j, X_j та λ_j слід підставити відповідно Z_Y, Y та λ_Y.

Таблиця 2

Дані з кластеру 2

№	Y	X_1	X_2	X_3	SMD$_Z$	№	Y	X_1	X_2	X_3	SMD$_Z$
1	54,052	187	11,41	1,29	9,09	21	41,347	150	10,49	1,34	0,40
2	128,355	341	11,38	1,20	7,68	22	38,654	132	11,10	1,29	1,35
3	127,696	330	11,52	1,19	8,03	23	39,817	152	10,33	1,32	0,34
4	119,424	274	12,70	1,17	9,07	24	41,236	150	10,49	1,30	0,22
5	42,068	142	11,66	1,33	2,48	25	38,867	135	11,01	1,31	0,69
6	37,940	132	10,89	1,29	1,79	26	39,404	146	10,44	1,33	0,32
7	39,073	138	10,80	1,31	0,64	27	38,763	134	11,01	1,31	0,84
8	40,487	143	10,76	1,33	0,42	28	44,253	155	11,25	1,31	2,62
9	38,994	140	10,72	1,31	0,48	29	39,205	148	10,26	1,31	0,28
10	39,269	146	10,38	1,30	0,41	30	39,191	142	10,62	1,35	0,60
11	41,850	142	11,23	1,30	0,48	31	39,611	143	10,65	1,31	0,36
12	155,74	420	11,33	1,21	10,79	32	39,565	142	10,89	1,31	0,45
13	38,912	141	10,60	1,31	0,52	33	38,184	135	10,79	1,30	1,12
14	38,326	134	10,85	1,31	1,10	34	41,800	154	10,49	1,33	0,49
15	39,699	139	10,81	1,31	0,63	35	47,610	289	5,33	1,33	18,81
16	45,540	177	9,91	1,38	2,29	36	61,269	487	4,93	1,79	15,69
17	37,754	134	10,79	1,31	1,04	37	24,040	303	3,67	1,75	8,84
18	40,908	140	11,04	1,33	0,52	38	24,660	367	3,36	1,64	11,65
19	39,276	144	10,44	1,33	0,43	39	24,347	328	3,59	1,24	12,77
20	38,431	147	10,43	1,38	1,36	40	20,327	308	3,94	1,20	22,91

В таблиці 1 наведені значення квадрату відстані Махаланобіса (SMD) для нормалізованих даних SMD$_Z$. Всі вони менше за квантиль $\chi^2_{4,0,005}$ з 4 ступенями свободи та 0,005 рівнем значущості, що дорівнює 14,86. Це вказує на відсутність чотиривимірних викидів у даних таблиці 1.

За критерієм Мардіа чотиривимірний розподіл даних для 40 веб-застосунків (40 строк даних з таблиці 2) відхиляється від гаусівського тому, що значення тестової статистики для багатовимірної асиметрі більше за квантиль $\chi^2_{20,0,005}$ з 20 ступенями свободи та 0,005 рівнем значущості, а саме: 298,88>40,00. Тому для визначення наявності чотиривимірних викидів у даних таблиці 2 ми також використовували критерій на основі квадрату відстані Махаланобіса для нормалізованих даних. Для нормалізації даних ми також використовували чотиривимірне перетворення Бокса-Кокса з компонентами (1).

В таблиці 2 також наведені значення SMD$_Z$. Дані рядка 40 є чотиривимірним викидом і вони були видалені тому, що значення SMD$_Z$ для них більше за 14,86 та є максимальним серед інших значень SMD$_Z$. Далі ми знову виконали нормалізацію для 39 точок даних (рядки 1-39) з таблиці 2 та обчислили для них нові значення SMD$_Z$. Дані рядка 39 виявилися чотиривимірним викидом і вони були також видалені. І так ми діяли доки не видалили з таблиці 1 всі шість викидів: рядки 35-40. В результаті первинний кластер 2 був поділений на дві частки. Перша частка містить 34 точки даних з таблиці 2 (рядки 1-34), а друга – 6 точок даних (рядки 35-40).

За даними таблиці 1 та двох часток даних таблиці 2 були побудовані три лінійні регресійні моделі

$$Y = \hat{Y} + \varepsilon = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon, \tag{2}$$

де оцінки параметрів $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ визначалися за методом найменших квадратів. Причому в (2) ε, яка характеризує відхилення значення залежної випадкової величини Y від лінії регресії або умовного вибіркового середнього $\hat{Y}$, повинна бути випадковою величиною з розподілом Гаусу з нульовим математичним сподіванням та дисперсією σ_ε^2, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$.

Нульова гіпотеза про нормальність розподілу ε для трьох моделей виду (2) була перевірена за критерієм Колмогорова-Смірнова. Згідно вказаного критерію для моделі (2), яка побудована за даними таблиці 1, – 48 точками даних (модель 1), ця гіпотеза відхиляється для рівня значущості 0,05. Те, що в цій моделі 1 розподіл ε не може вважатися гаусівським, призводить до відсутності теоретичного обґрунтування застосування лінійної регресійної моделі (моделі 1) для даних з таблиці 1.

Згідно критерію Колмогорова-Смірнова гіпотеза про нормальність розподілу ε для двох моделей виду (2), які побудовані за двома частками даних таблиці 2, – 34 точками даних (модель 2) та 6 точками даних (модель 3) не може бути відхилена для рівня значущості 0,05. Те, що в (2) розподіл ε може вважатися гаусівським, є одним з теоретичних обґрунтувань застосування двох лінійних регресійних моделей 2 та 3 для відповідних часток даних з таблиці 2. Оцінки параметрів двох лінійних регресійних моделей 2 та 3 виду (2) наведені в таблиці 3. В таблиці 3 також наведені значення трьох показників якості двох побудованих лінійних регресійних моделей 2 та 3 виду (2).

Таблиця 3

Оцінки параметрів та показників якості лінійних регресійних моделей

№	Модель	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{\sigma}_\varepsilon$	R^2	*MMRE*	*PRED*(0,25)
1	Модель 2	–1,4454	1,9693	1,4694	–0,0008	0,00005	0,9905	0,0152	1,0000
2	Модель 3	–77,220	0,0895	16,1602	8,7234	2,479	0,9780	0,0685	1,0000

Якість двох побудованих лінійних регресійних моделей виду (2) було перевірено за коефіцієнтом детермінації R^2 (the coefficient of determination), середньою величиною відносної помилки *MMRE* (Mean Magnitude of Relative Error) і відсотком прогнозованих результатів *PRED* (Percentage of Prediction), для яких величини відносної помилки *MRE* (Magnitude of Relative Error) менші за 0,25, *PRED*(0,25). Ці показники зазвичай використовуються для оцінювання якості прогнозування за допомогою регресійних моделей.

Значення показників якості з таблиці 3 свідчать про прийнятну якість двох побудованих лінійних регресійних моделей виду (2). Нагадаємо, зазвичай $MMRE \leq 0{,}25$ та $PRED(0{,}25) \geq 0{,}75$ вважається прийнятною точністю прогнозування за допомогою регресійних моделей. А високі значення R^2 вказують на те, що більшість значень Y знаходяться або на лінії регресії або біля неї.

За даними таблиці 1, як і в [7], була побудована нелінійна регресійна модель на основі чотиривимірного перетворення Бокса-Кокса

$$Y = \left[\hat{\lambda}_Y\left(\hat{Z}_Y + \varepsilon\right) + 1\right]^{1/\hat{\lambda}_Y}, \tag{3}$$

де $\hat{Z}_Y$ – умовне вибіркове середнє за лінійним регресійним рівнянням $\hat{Z}_Y = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3$ для нормалізованих даних, які визначаються за чотиривимірним перетворенням Бокса-Кокса з компонентами (1) та відповідними оцінками параметрів $\hat{\lambda}_Y = 0{,}176406$, $\hat{\lambda}_1 = 0{,}223415$, $\hat{\lambda}_2 = 1{,}072463$, $\hat{\lambda}_3 = -2{,}316493$. Для (3) оцінки параметрів $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ визначалися за методом найменших квадратів і мають відповідно такі значення: –2,50158, 0,59750, 0,17715 та 0,16543. Оцінка $\hat{\sigma}_\varepsilon$ дорівнює 0,2290.

Для нелінійної регресійної моделі виду (3) показники R^2, *MMRE* та *PRED*(0,25) відповідно дорівнюють 0,9966, 0,1516 та 0,7708. Ці значення свідчать про прийнятну якість нелінійної регресійної моделі виду (3).

Нелінійна регресійна модель (3) з наведеними вище параметрами дозволяє оцінювати кількість рядків коду веб-застосунків, які створюються за допомогою фреймворків CakePHP та Codeigniter, в залежності від факторів у таких діапазонах: $1 \leq X_1 \leq 106$, $0{,}18 \leq X_2 \leq 13{,}75$ і $1{,}24 \leq X_3 \leq 3{,}00$. Лінійна регресійна модель 2 виду (1) дозволяє оцінювати кількість рядків коду веб-застосунків, які створюються за допомогою фреймворку Codeigniter, в залежності від факторів у таких діапазонах: $132 \leq X_1 \leq 420$, $9{,}91 \leq X_2 \leq 12{,}70$ і $1{,}17 \leq X_3 \leq 1{,}38$. Лінійна регресійна модель 3 виду (1) дозволяє оцінювати кількість рядків коду веб-застосунків, які створюються за допомогою фреймворку CakePHP, в залежності від факторів у таких діапазонах: $289 \leq X_1 \leq 487$, $3{,}36 \leq X_2 \leq 5{,}33$ і $1{,}20 \leq X_3 \leq 1{,}79$.

Отримані значення показників якості двох удосконалених лінійних моделей виду (2) є практично такими ж, що і нелінійні регресійні моделі, які запропоновані у [5] та [6]. Також зазначимо, що хоча побудована модель (3) має дещо гірші значення *MMRE* та *PRED*(0,25) у порівнянні з нелінійними регресійними моделями, які наведені в [5] та [6], але на відміну від них модель (3) разом з двома моделями виду (2) надають можливість здійснювати раннє оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою фреймворків CakePHP та Codeigniter, для всього набору даних.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Удосконалено нелінійну регресійну модель, яка побудована на основі чотиривимірного перетворення Бокса-Кокса, за рахунок знайдених нових значень оцінок параметрів для раннього оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою фреймворків CakePHP та Codeigniter для діапазону кількості класів від 1 до 106. Удосконалено дві лінійні регресійні моделі за рахунок знайдених нових значень оцінок параметрів для раннього оцінювання кількості рядків коду

відповідно веб-застосунків, що створюються за допомогою фреймворку Codeigniter для діапазону кількості класів від 132 до 420, та веб-застосунків, що створюються за допомогою фреймворку CakePHP для діапазону кількості класів від 289 до 487. На відміну від існуючих моделей ці три удосконалені моделі надають можливість здійснювати раннє оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою фреймворків CakePHP та Codeigniter, для всього набору даних при прийнятній якості відповідного оцінювання. В подальшому планується побудова моделей для раннього оцінювання кількості рядків коду веб-застосунків, що створюються за допомогою інших фреймворків.

Література

1. Brar, P., & Nandal, D. (2022). A systematic literature review of machine learning techniques for software effort estimation models. *Computational Intelligence and Communication Technologies (CCICT): Proceedings of the 2022 Fifth International Conference, Sonepat, India*, 494–499. https://doi.org/10.1109/CCiCT56684.2022.00093

2. Rahman, M., Sarwar, H., Kader, M. A., Gonçalves, T., & Tin, T. T. (2024). Review and empirical analysis of machine learning-based software effort estimation. *IEEE Access, 12*, 85661–85680. https://doi.org/10.1109/ACCESS.2024.3404879

3. Hussain, I., & Malik, A. A. (2023). Determining the utility of use case points and class points in early software size estimation. *Emerging Technologies (ICET): Proceedings of the 2023 18th International Conference, Peshawar, Pakistan*, 171–175. https://doi.org/10.1109/ICET59753.2023.10374977

4. Yuan, X., Su, J., Yu, C., & Ye, S. (2023). Power grid software cost estimation based on improved COCOMO model. *Electronic Technology, Communication and Information (ICETCI): Proceedings of the 2023 IEEE 3rd International Conference, Changchun, China*, 1265–1269. https://doi.org/10.1109/ICETCI57876.2023.10176686

5. Prykhodko, S. B., Shutko, I. S., & Prykhodko, A. S. (2021). A nonlinear regression model to estimate the size of web apps created using the CakePHP framework. *Radio Electronics, Computer Science, Control, 59*(4), 129–139. https://doi.org/10.15588/1607-3274-2021-4-12

6. Prykhodko, S. B., Shutko, I. S., & Prykhodko, A. S. (2022). Early size estimation of web apps created using CodeIgniter framework by nonlinear regression models. *Radio-Electronic and Computer Systems, 103*(3), 84–94. https://doi.org/10.32620/reks.2022.3.06

7. Prykhodko, S., & Prykhodko, N. (2023). Building nonlinear regression models for estimating the number of clusters and their initial centroids. *Computer Sciences and Information Technologies (CSIT): Proceedings of the 2023 IEEE 18th International Conference, Lviv, Ukraine*, 1–4. https://doi.org/10.1109/CSIT61576.2023.10324095

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

9. Dewi, R. S., Araynawa, T. K., Prasanna, F. M., et al. (2024). Improving software size estimation using data complexity (Case study: Research and community service monitoring apps). *Electrical Engineering, Computer Science and Informatics (EECSI): Proceedings of the 2024 11th International Conference, Yogyakarta, Indonesia*, 315–319. https://doi.org/10.1109/EECSI63442.2024.10776530

10. Dewi, R. S., Zahrah, F. A., Nugraha, D. A., et al. (2024). Predicting software size based on conceptual data model (Case study: Shrimp pond system management). *Electrical Engineering and Computer Science (ICECOS): Proceedings of the 2024 International Conference, Palembang, Indonesia*, 175–178. https://doi.org/10.1109/ICECOS63900.2024.10791154

11. Nassif, A. B., AbuTalib, M., & Capretz, L. F. (2020). Software effort estimation from use case diagrams using nonlinear regression analysis. *In Electrical and Computer Engineering: Proceedings of the IEEE Canadian Conference, London, ON, Canada*, 1–4. https://doi.org/10.1109/CCECE47787.2020.9255712

12. Sahoo, P., Behera, D. K., Mohanty, J. R., et al. (2022). Effort estimation of software products by using UML sequence models with regression analysis. *Information Technology (OCIT): Proceedings of the 2022 OITS International Conference, Bhubaneswar, India*, 97–101. https://doi.org/10.1109/OCIT56763.2022.00028

13. Manisha, & Rishi, R. (2021). Early size estimation using machine learning. *Computing for Sustainable Global Development (INDIACom): Proceedings of the 2021 8th International Conference, New Delhi, India*, 757–762. https://doi.org/10.1109/INDIACom51348.2021.00135

14. Nhung, H. L. T. K., Hai, V. V., Silhavy, R., et al. (2022). Parametric software effort estimation based on optimizing correction factors and multiple linear regression. *IEEE Access, 10*, 2963–2986. https://doi.org/10.1109/ACCESS.2021.3139183

References

1. Brar, P., & Nandal, D. (2022). A systematic literature review of machine learning techniques for software effort estimation models. Computational Intelligence and Communication Technologies (CCICT): Proceedings of the 2022 Fifth International Conference, Sonepat, India, 494–499. https://doi.org/10.1109/CCiCT56684.2022.00093

2. Rahman, M., Sarwar, H., Kader, M. A., Gonçalves, T., & Tin, T. T. (2024). Review and empirical analysis of machine learning-based software effort estimation. IEEE Access, 12, 85661–85680. https://doi.org/10.1109/ACCESS.2024.3404879

3. Hussain, I., & Malik, A. A. (2023). Determining the utility of use case points and class points in early software size estimation. Emerging Technologies (ICET): Proceedings of the 2023 18th International Conference, Peshawar, Pakistan, 171–175. https://doi.org/10.1109/ICET59753.2023.10374977

4. Yuan, X., Su, J., Yu, C., & Ye, S. (2023). Power grid software cost estimation based on improved COCOMO model. Electronic Technology, Communication and Information (ICETCI): Proceedings of the 2023 IEEE 3rd International Conference, Changchun, China, 1265–1269. https://doi.org/10.1109/ICETCI57876.2023.10176686

5. Prykhodko, S. B., Shutko, I. S., & Prykhodko, A. S. (2021). A nonlinear regression model to estimate the size of web apps created using the CakePHP framework. Radio Electronics, Computer Science, Control, 59(4), 129–139. https://doi.org/10.15588/1607-3274-2021-4-12

6. Prykhodko, S. B., Shutko, I. S., & Prykhodko, A. S. (2022). Early size estimation of web apps created using CodeIgniter framework by nonlinear regression models. Radio-Electronic and Computer Systems, 103(3), 84–94. https://doi.org/10.32620/reks.2022.3.06

7. Prykhodko, S., & Prykhodko, N. (2023). Building nonlinear regression models for estimating the number of clusters and their initial centroids. Computer Sciences and Information Technologies (CSIT): Proceedings of the 2023 IEEE 18th International Conference, Lviv, Ukraine, 1–4. https://doi.org/10.1109/CSIT61576.2023.10324095

8. Daud, M., & Malik, A. A. (2021). Improving the accuracy of early software size estimation using analysis-to-design adjustment factors (ADAFs). IEEE Access, 9, 81986–81999. https://doi.org/10.1109/ACCESS.2021.3085752

9. Dewi, R. S., Araynawa, T. K., Prasanna, F. M., et al. (2024). Improving software size estimation using data complexity (Case study: Research and community service monitoring apps). Electrical Engineering, Computer Science and Informatics (EECSI): Proceedings of the 2024 11th International Conference, Yogyakarta, Indonesia, 315–319. https://doi.org/10.1109/EECSI63442.2024.10776530

10. Dewi, R. S., Zahrah, F. A., Nugraha, D. A., et al. (2024). Predicting software size based on conceptual data model (Case study: Shrimp pond system management). Electrical Engineering and Computer Science (ICECOS): Proceedings of the 2024 International Conference, Palembang, Indonesia, 175–178. https://doi.org/10.1109/ICECOS63900.2024.10791154

11. Nassif, A. B., AbuTalib, M., & Capretz, L. F. (2020). Software effort estimation from use case diagrams using nonlinear regression analysis. In Electrical and Computer Engineering: Proceedings of the IEEE Canadian Conference, London, ON, Canada, 1–4. https://doi.org/10.1109/CCECE47787.2020.9255712

12. Sahoo, P., Behera, D. K., Mohanty, J. R., et al. (2022). Effort estimation of software products by using UML sequence models with regression analysis. Information Technology (OCIT): Proceedings of the 2022 OITS International Conference, Bhubaneswar, India, 97–101. https://doi.org/10.1109/OCIT56763.2022.00028

13. Manisha, & Rishi, R. (2021). Early size estimation using machine learning. Computing for Sustainable Global Development (INDIACom): Proceedings of the 2021 8th International Conference, New Delhi, India, 757–762. https://doi.org/10.1109/INDIACom51348.2021.00135

14. Nhung, H. L. T. K., Hai, V. V., Silhavy, R., et al. (2022). Parametric software effort estimation based on optimizing correction factors and multiple linear regression. IEEE Access, 10, 2963–2986. https://doi.org/10.1109/ACCESS.2021.3139183