УДК 004.412:519.237.5

С.Б. ПРИХОДЬКО, І.С. ШУТКО
Національний університет кораблебудування імені адмірала Макарова

РЕГРЕСІЙНІ МОДЕЛІ ДЛЯ РАННЬОГО ОЦІНЮВАННЯ КІЛЬКОСТІ РЯДКІВ КОДУ ВЕБЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ CODEIGNITER

Проблема раннього оцінювання кількості рядків коду у проєктах програмного забезпечення має велике значення, оскільки це безпосередньо впливає на прогнозування зусиль із розроблення програмних застосунків, зокрема й вебзастосунків, які створюються за допомогою такого відомого PHP фреймворку, як Codeigniter. Об'єктом дослідження є процес раннього оцінювання кількості рядків коду вебзастосунку, які створюються за допомогою Codeigniter. Предметом дослідження є регресійні моделі для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою Codeigniter.

Метою роботи є побудова декількох трифакторних регресійних моделей для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку "Codeigniter", залежно від чинників, що можуть бути знайдені за діаграмою класів.

У роботі побудовано дві лінійні регресійні моделі для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку "Codeigniter", залежно від трьох чинників: кількості класів, середньої кількості методів на клас і метрики DIT (Depth of Inheritance Tree) на рівні застосунку. Зазначені чинники вибрано із двох причин: їхнє значення можна отримати з діаграми класів, вони не мають проблеми мультиколінеарності. Оцінки параметрів отриманих моделей були знайдені за методом найменших квадратів. Зазначені моделі побудовані на основі розбиття чотиривимірного набору даних (фактичний розмір у тисячах рядків коду; кількість класів; середня кількість методів на клас; метрика DIT на рівні застосунку) на два кластери. Ці дані з метрик для 50 вебзастосунків із відкритим кодом, що створювалися за допомогою фреймворку "Codeigniter", були отримані за допомогою інструменту PhpMetrics (https://phpmetrics.org/). Ці застосунки розміщені на платформі "GitHub" (https://github.com/). Виконано порівняння побудованих лінійних регресійних моделей з існуючими нелінійними регресійними моделями. Ці дві лінійні моделі в порівнянні з існуючими нелінійними моделями дозволяють описувати всі чотиривимірні дані, за якими вони були побудовані. Урахування всього набору даних, зокрема й аномальних значень, дозволяє підвищити універсальність моделей. Аномалії, хоча і можуть впливати на точність окремих прогнозів, часто є важливими для відображення специфічних особливостей реальних проєктів. Використання таких даних сприяє створенню моделей, які є більш адаптивними до різноманітних сценаріїв розробки.

Ключові слова: *регресійна модель, оцінювання, рядки коду, вебзастосунок, PHP, фреймворк, Codeigniter, клас, метод, глибина дерева успадкування, перетворення Бокса – Кокса.*

S.B. PRYKHODKO, I.S. SHUTKO
Admiral Makarov National University of Shipbuilding

REGRESSION MODELS FOR EARLY ESTIMATING THE LINES OF CODE COUNT OF WEB APPLICATIONS CREATED USING THE CODEIGNITER FRAMEWORK

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 such the well-known PHP framework, as the Codeigniter. The object of the study is the process of early estimating the lines of code count of web applications created using the Codeigniter framework. The subject of the study is the regression models for early estimating the lines of code count of web applications created using the Codeigniter framework.

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 Codeigniter framework depending on the factors that can be found in the class diagram.

In the work, we built two linear regression models for early estimating the lines of code count of web applications created using the Codeigniter framework 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 50 open-source web applications created using the Codeigniter framework was obtained

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

using the PhpMetrics tool (https://phpmetrics.org/). These applications are hosted on the GitHub platform (https://github.com/). The comparison of the constructed linear regression models with the non-linear regression models is performed. These two linear models, in comparison with existing non-linear ones, allow us to describe all the four-dimensional data on which they were built.

Key words: *regression model, estimation, lines of code, web application, PHP, framework, Codeigniter, class, method, depth of inheritance tree, Box – Cox transformation.*

Постановка проблеми

Завдання раннього оцінювання кількості рядків коду програмного забезпечення (далі – ПЗ), вебзастосунків також, важливе для керівників проєктів ПЗ, оскільки ця інформація використовується для визначення зусиль створення ПЗ за допомогою різноманітних методів і моделей [1–3]. Зазначене оцінювання є необхідною умовою для планування проєкту ПЗ та складання його бюджету [4; 5].

Водночас для розроблення вебзастосунків широко використовують фреймворки, які прискорюють створення цих застосунків. Серед таких відомих фреймворків дуже популярний Codeigniter (https://www.codeigniter.com/) – потужний PHP-фреймворк із відкритим вихідним кодом для створення багатофункціональних і безпечних вебзастосунків з архітектурою MVC (Model-View-Controller).

Хоча для оцінювання кількості строк коду вебзастосунків, що створюються за допомогою фреймворку “Codeigniter”, уже було запропоновано відповідну нелінійну регресійну модель [6], вона не дозволяє описувати частину даних, що були відкинуті як аномальні під час її побудови. Ігнорування аномальних даних у таких моделях може знижувати їхню універсальність, адже виключені точки можуть відображати реальні, хоч і нетипові сценарії розробки. У цьому контексті побудова лінійних моделей, що враховують увесь набір даних, є обґрунтованою. Вони не лише забезпечують точність, але й дозволяють краще зрозуміти вплив окремих чинників на кількість рядків коду. Це потребує побудови декількох математичних моделей для оцінювання кількості строк коду вебзастосунків, що створюються за допомогою фреймворку “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 останніх досліджень та публікацій

У наш час, незважаючи на існування досить великої кількості методів і моделей для оцінювання кількості строк коду ПЗ на ранній стадії розробки, підвищення точності оцінювання залишається актуальним завданням [8; 9]. Для відповідного оцінювання використовують різноманітні підходи, які зазвичай 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 [6].

Незважаючи на широке застосування в наш час машинного навчання [1; 3; 13], методи як лінійного [14], так і нелінійного регресійного аналізу [6; 11] продовжують використовуватися для відповідного оцінювання. Так, натепер відомі нелінійні регресійні моделі для оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою окремих PHP фреймворків, зокрема й Codeigniter [6]. Але розроблена в роботі [6] трифакторна нелінійна регресійна модель на основі чотиривимірного перетворення Бокса – Кокса, хоча й має добрі показники якості для набору даних, за яким вона була створена, однак не дозволяє описувати частину даних, що були відкинути як аномальні під час її побудови. Це призводить до необхідності побудови декількох моделей, які надавали би можливість здійснювати оцінювання кількості рядків коду вебзастосунків, що створюються за допомогою фреймворку “Codeigniter”, для всього набору даних після розбиття його на кластери.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

Мета дослідження

Метою дослідження є побудова декількох регресійних моделей для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку “Codeigniter”, залежно від чинників, що можуть бути знайдені за діаграмою класів.

Об’єктом дослідження є процес раннього оцінювання кількості рядків коду вебзастосунки, які створюються за допомогою фреймворку “Codeigniter”. Предметом дослідження є регресійні моделі для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою Codeigniter.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Побудувати дві регресійні моделі для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку “Codeigniter”, що, на відміну від моделей, які існують, дозволяли би описувати всі дані, за якими вони були побудовані.

2. Виконати порівняння якості побудованих лінійних регресійних моделей з наявними нелінійними регресійними моделями для раннього оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку “Codeigniter”.

Виклад основного матеріалу дослідження

Спочатку було здійснено вибір даних метрик вебзастосунків, що створювалися на основі фреймворку “Codeigniter”. Ми взяли дані з роботи [6] для 50 вебзастосунків за такими метриками: розмір вебзастосунку 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; по-друге, між ними відсутня мультиколінеарність.

Далі вибрані дані були розбиті на дві частки (кластери). Для цього було використано результати роботи [7]. До кластера 1 увійшло 16 точок даних, а до кластера 2 – 34 точки даних зазначених вище метрик. Значення зазначених метрик для 16 вебзастосунків (кластер 1) наведені в таблиці 1.

Значення зазначених вище метрик для 34 вебзастосунків (кластер 2) наведені в табл. 2. Також у таблицях 1 і 2 наведені значення квадрата відстані Махаланобіса MD^2.

Відповідно до [6], чотиривимірний розподіл даних для 50 вебзастосунків (50 строк даних із таблиць 1 і 2) відхиляється від гаусівського. Але чотиривимірний розподіл для 16 строк даних із таблиці 1 можна наближено вважати нормальним згідно із критерієм Мардіа та критерієм на основі квадрата відстані Махаланобіса. У даних із таблиці 1 відсутні чотиривимірні викиди. Це підтверджують значення квадрата відстані Махаланобіса з таблиці 1.

Чотиривимірний розподіл для 34 строк даних із таблиці 2 відхиляється від нормального згідно із критерієм Мардіа та критерієм на основі квадрата відстані Махаланобіса. Тому, як і в роботах [6; 7], для визначення наявності чотиривимірних викидів у даних із таблиці 2 ми здійснили їх нормалізацію за допомогою чотиривимірного перетворення Бокса – Кокса.

Таблиця 1

Дані із кластера 1

№	Y	X_1	X_2	X_3	MD^2	№	Y	X_1	X_2	X_3	MD^2
1	2,28	32	0,31	1,71	5,10	9	32,778	100	13,10	1,26	1,48
2	2,265	31	0,19	1,75	4,50	10	29,738	94	12,84	1,24	5,42
3	32,39	97	13,31	1,26	1,08	11	33,373	97	13,75	1,26	1,94
4	33,368	101	13,45	1,25	1,73	12	34,230	106	12,86	1,31	5,47
5	28,333	83	13,41	1,24	3,04	13	2,097	30	0,20	1,75	3,30
6	2,361	35	0,23	1,77	7,83	14	2,347	34	0,18	1,77	4,17
7	0,939	10	3,80	1,67	10,33	15	31,365	98	12,89	1,25	2,33
8	28,653	83	13,4	1,29	4,84	16	32,836	101	13,19	1,26	1,46

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

Таблиця 2

Дані із кластера 2

№	Y	X_1	X_2	X_3	MD^2	№	Y	X_1	X_2	X_3	MD^2
1	54,052	187	11,41	1,29	15,15	18	40,908	140	11,04	1,33	2,16
2	128,355	341	11,38	1,20	7,38	19	39,276	144	10,44	1,33	1,19
3	127,696	330	11,52	1,19	7,61	20	38,431	147	10,43	1,38	8,45
4	119,424	274	12,70	1,17	22,15	21	41,347	150	10,49	1,34	1,60
5	42,068	142	11,66	1,33	10,50	22	38,654	132	11,10	1,29	1,84
6	37,940	132	10,89	1,29	2,00	23	39,817	152	10,33	1,32	1,76
7	39,073	138	10,80	1,31	0,35	24	41,236	150	10,49	1,30	2,36
8	40,487	143	10,76	1,33	0,86	25	38,867	135	11,01	1,31	0,61
9	38,994	140	10,72	1,31	0,42	26	39,404	146	10,44	1,33	0,90
10	39,269	146	10,38	1,30	3,48	27	38,763	134	11,01	1,31	0,65
11	41,850	142	11,23	1,30	1,76	28	44,253	155	11,25	1,31	4,15
12	155,74	420	11,33	1,21	17,95	29	39,205	148	10,26	1,31	3,18
13	38,912	141	10,60	1,31	0,81	30	39,191	142	10,62	1,35	3,19
14	38,326	134	10,85	1,31	0,54	31	39,611	143	10,65	1,31	0,56
15	39,699	139	10,81	1,31	0,34	32	39,565	142	10,89	1,31	0,54
16	45,540	177	9,91	1,38	9,28	33	38,184	135	10,79	1,30	1,08
17	37,754	134	10,79	1,31	0,53	34	41,800	154	10,49	1,33	0,66

За критерієм на основі квадрата відстані Махаланобіса для нормалізованих даних із таблиці 2 чотиривимірні викиди відсутні. Зазначимо, що в разі ігнорування відхилення чотиривимірного розподілу даних від нормального, ми, навпаки, маємо три чотиривимірні викиди в даних із таблиці 2: це рядки 1, 4, 12. Про це свідчать значення квадрата відстані Махаланобіса, які наведені в таблиці 2. Тому під час визначення багатовимірних викидів у даних обов'язково треба враховувати таке: чи не відхиляється багатовимірний розподіл даних від нормального (гаусівського). Остаточно ми прийняли відсутність викидів у даних із таблиць 1 і 2.

За даними таблиць 1 і 2 були побудовані дві лінійні регресійні моделі

$$Y = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon\,, \tag{1}$$

де оцінки параметрів $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ та $\hat{b}_3$ визначалися за методом найменших квадратів. В (1) змінна ε, яка описує відхилення залежної випадкової величини Y від лінії регресії, повинна бути випадковою величиною з розподілом Гауса, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$.

Нульова гіпотеза про нормальність розподілу для двох моделей виду (1) була перевірена за критеріями Пірсона та Колмогорова – Смирнова. Ця гіпотеза не може бути відхилена для рівня значущості 0,05. Те, що в (1) розподіл може вважатися гаусівським, є одним із теоретичних обґрунтувань можливості застосування лінійних регресійних моделей виду (1) для даних із таблиць 1 та 2 відповідно. Оцінки параметрів двох моделей виду (1), які побудовані за даними таблиці 1 (модель 1) і таблиці 2 (модель 2), наведені в таблиці 3.

Таблиця 3

Параметри та показники якості моделей

Модель	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	$\hat{\sigma}_\varepsilon$	R^2	*MMRE*	*PRED*
Модель 1	−23,4197	0,25125	1,40578	9,84847	0,4259	0,9992	0,0776	0,9375
Модель 2	19,9204	0,38757	5,67136	−73,1579	2,4206	0,9938	0,0305	1,0000

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

Якість двох побудованих лінійних регресійних моделей виду (10) 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. У таблиці 3 наведені значення цих трьох показників якості двох побудованих лінійних регресійних моделей виду (1).

Значення показників якості з таблиці 3 свідчать про задовільну якість двох побудованих лінійних регресійних моделей виду (10). Нагадаємо, зазвичай <<Eqn016.eps>> та <<Eqn017.eps>> уважається прийнятною точністю прогнозування за допомогою регресійних моделей. А високі значення R^2 вказують на те, що більшість значень Y перебувають або на лінії регресії або біля неї.

Якість двох побудованих лінійних регресійних моделей виду (1) було також порівняно із трьома нелінійними регресійними моделями з роботи [6], які створені на основі десяткового логарифма, одновимірного та чотиривимірного перетворень Бокса – Кокса відповідно. Усі моделі, як лінійні, так і нелінійні, мають практично однакові високі значення R^2, які різняться у третій значущій цифрі. Найбільше значення R^2, яке дорівнює 0,9992, має модель 1, а найменше – модель 2. Серед нелінійних моделей найбільше значення R^2, яке дорівнює 0,9984, має модель, побудована на основі чотиривимірного перетворення Бокса – Кокса. Найкраще значення *MMRE*, яке дорівнює 0,0167, також має нелінійна модель, побудована на основі чотиривимірного перетворення Бокса – Кокса, а найгірше – лінійна модель 1. Зазначимо, лінійна модель 2 має менше значення *MMRE*, яке дорівнює 0,0305, порівняно із двома нелінійними моделями, що побудовані на основі десяткового логарифма й одновимірного перетворення Бокса – Кокса, значення *MMRE* яких є відповідно такими: 0,0492 та 0,0452. Найбільші однакові значення *PRED* (0,25) мають дві моделі: лінійна модель 2 та нелінійна модель, побудована на основі чотиривимірного перетворення Бокса – Кокса. Найменше значення *PRED* (0,25), яке дорівнює 0,8261, має нелінійна модель, побудована на основі десяткового логарифма.

Виконаний аналіз значень показників якості свідчить про те, що нелінійна регресійна модель, побудована на основі чотиривимірного перетворення Бокса – Кокса, має дещо кращий показник MMRE. Але ця нелінійна модель не дозволяє описувати частину даних (рядки 2, 7, 8 із таблиці 1 і рядок 1 із таблиці 2), що були відкинути як аномальні під час її побудови. Так само не дозволяють описувати частину даних і дві інші нелінійні моделі, що побудовані на основі одновимірних нормалізуючих перетворень (десяткового логарифма й одновимірного перетворення Бокса – Кокса відповідно). І навпаки, дві лінійні моделі виду (1), які мають задовільні показники якості, дозволяють описувати всі дані з таблиць 1 і 2. У цьому є перевага побудованих двох лінійних моделей виду (1) порівняно з нелінійними регресійними моделями з роботи [6]. До недоліків двох лінійних моделей виду (1) можна насамперед віднести те, що є розрив у діапазонах за кількістю класів для цих моделей: модель 1 побудована для діапазону від 10 до 106 класів, а модель 2 – від 132 до 420 класів. До обмежень щодо застосування цих двох лінійних моделей для оцінювання кількості рядків коду вебзастосунків, які створюються за допомогою фреймворку “Codeigniter”, також можна віднести таке. Регресійна модель 1 призначена для здійснення відповідного оцінювання для кількості класів від 10 до 106, середньої кількості методів на клас від 0,18 до 13,75, метрики DIT на рівні застосунку від 1,24 до 1,77. Регресійна модель 2 призначена для здійснення відповідного оцінювання для кількості класів від 132 до 420, середньої кількості методів на клас від 9,91 до 12,70, метрики DIT на рівні застосунку від 1,17 до 1,38. Надалі, на наш погляд, ці обмеження можна подолати побудовою нових моделей на основі більшої кількості даних.

Висновки

Удосконалено дві трифакторні лінійні регресійні моделі для оцінювання кількості рядків коду вебзастосунків, що створюються з використанням фреймворку “Codeigniter”, залежно

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

від кількості класів, середньої кількості методів на клас і метрики DIT на рівні застосунку на основі розбиття набору даних на два кластери. Ці моделі, порівняно з наявними регресійними моделями, дозволяють описувати всі дані, за якими вони були побудовані.

У визначенні наявності багатовимірних викидів у даних треба враховувати, чи не відхиляється багатовимірний розподіл даних від нормального (гаусівського). Якщо багатовимірний розподіл даних відхиляється від нормального, для виявлення викидів можна застосувати метод, заснований на квадраті відстані Махаланобіса. Дані варто попередньо нормалізувати за допомогою багатовимірного перетворення.

Включення аномальних даних у моделі сприяє не лише підвищенню точності для всіх типів сценаріїв, але й дозволяє краще зрозуміти закономірності в розподілі метрик, що є важливим для раннього оцінювання зусиль розробки.

Надалі планується побудова регресійних моделей для оцінювання кількості рядків коду вебзастосунків, що розробляються з використанням Codeigniter, без розриву в діапазонах за кількістю класів.

Список використаної літератури

1. Brar P., Nandal D. A systematic literature review of machine learning techniques for software effort estimation models. *Computational intelligence and communication technologies (CCICT) :* proceedings of 2022 Fifth International conference, Sonepat, India: *IEEE*, 2022. P. 494–499. https://doi.org/10.1109/CCiCT56684.2022.00093.
2. Kumar S., Arora M., Sakshi Chopra S. A review of effort estimation in agile software development using machine learning techniques. *Inventive research in computing applications (ICIRCA) :* proceedings of the 2022 4th International conference, Coimbatore, India: IEEE, 2022. P. 416–422. https://doi.org/10.1109/ICIRCA54612.2022.9985542.
3. 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. Vol. 12. P. 85661–85680. https://doi.org/10.1109/ACCESS.2024.3404879.
4. Hussain I., Malik A.A. Determining the utility of use case points and class points in early software size estimation. *Emerging Technologies (ICET) :* proceedings of 2023 18th International Conference, Peshawar, Pakistan: *IEEE*, 2023. P. 171–175. https://doi.org/10.1109/ICET59753.2023.10374977.
5. Yuan X., Su J., Yu C., Ye S. 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, Los Alamitos: *IEEE*, 2023. P. 1265–1269. https://doi.org/10.1109/ICETCI57876.2023.10176686.
6. 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. Vol. 103. № 3. P. 84–94. https://doi.org/10.32620/reks.2022.3.06.
7. Prykhodko S., Prykhodko N. 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: *IEEE*, 2023. P. 1–4. https://doi.org/10.1109/CSIT61576.2023.10324095.
8. Daud M., Malik A.A. Improving the accuracy of early software size estimation using analysis-to-design adjustment factors (ADAFs). *IEEE Access*. 2021. Vol. 9. P. 81986–81999. https://doi.org/10.1109/ACCESS.2021.3085752.
9. Dewi R.S., Araynawa T.K., Prasanna F.M., Felianasari N., Rahmawati R., Hartantc A.E., … Mazaya Al-K. Improving software size estimation using data complexity (Case study: Research and community service monitoring apps). *Electrical engineering, computer science and informatics (EECSI) :* proceedings of *2024 11th International conference,* Yogyakarta, Indonesia: *IEEE*, 2024. P. 315–319. https://doi.org/10.1109/EECSI63442.2024.10776530.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18

10. 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). *Electrical engineering and computer science (ICECOS) :* proceedings of 2024 International conference, Palembang, Indonesia: *IEEE*, 2024. P. 175–178. https://doi.org/10.1109/ICECOS63900.2024.10791154.
11. Nassif A.B., AbuTalib M., Capretz L.F. Software effort estimation from Use Case diagrams using nonlinear regression analysis. In *On electrical and computer engineering :* proceedings of IEEE Canadian conference, London, ON, Canada: *IEEE*, 2020. P. 1–4. https://doi.org/10.1109/CCECE47787.2020.9255712.
12. 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. *Information Technology (OCIT) :* proceedings of the 2022 OITS International Conference, Bhubaneswar, India, Los Alamitos: *IEEE*, 2022. P. 97–101. https://doi.org/10.1109/OCIT56763.2022.00028.
13. Manisha Rishi R. Early size estimation using machine learning. *Computing for sustainable global development (INDIACom) :* proceedings of the 2021 8th International conference, New Delhi, India, Los Alamitos: *IEEE*, 2021. P. 757–762. https://doi.org/10.1109/INDIACom51348.2021.00135.
14. 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. Vol. 10. P. 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. In *2022 Fifth International conference on computational intelligence and communication technologies (CCICT),* 494–499. Sonepat, India: *IEEE*. https://doi.org/10.1109/CCiCT56684.2022.00093 [in English].
2. Kumar, S., Arora, M., Sakshi, & Chopra, S. (2022). A review of effort estimation in agile software development using machine learning techniques. In *the 2022 4th International conference on inventive research in computing applications (ICIRCA),* 416–422. Coimbatore, India: *IEEE*. https://doi.org/10.1109/ICIRCA54612.2022.9985542 [in English].
3. 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 [in English].
4. Hussain, I., & Malik, A.A. (2023). Determining the utility of use case points and class points in early software size estimation. In *2023 18th International Conference on Emerging Technologies (ICET),* 171–175. Peshawar, Pakistan: *IEEE*. https://doi.org/10.1109/ICET59753.2023.10374977 [in English].
5. Yuan, X., Su, J., Yu, C., & Ye, S. (2023). Power grid software cost estimation based on improved COCOMO model. In *the 2023 IEEE 3rd International conference on electronic technology, communication and information (ICETCI),* 1265–1269. Changchun, China, Los Alamitos: *IEEE*. https://doi.org/10.1109/ICETCI57876.2023.10176686 [in English].
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*, 3 (103), 84–94. https://doi.org/10.32620/reks.2022.3.06 [in English].
7. Prykhodko, S., & Prykhodko, N. (2023). Building nonlinear regression models for estimating the number of clusters and their initial centroids. in *the 2023 IEEE 18th International conference on computer sciences and information technologies (CSIT)*, 1–4. Lviv, Ukraine: IEEE. https://doi.org/10.1109/CSIT61576.2023.10324095 [in English].
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 [in English].
9. Dewi, R.S., Araynawa, T.K., Prasanna, F.M., Felianasari, N., Rahmawati, R., Hartantc, A.E., Kamilah, L.S., & Mazaya, Al-K. (2024). Improving software size estimation using data complexity (Case study: Research and community service monitoring apps). In *2024 11th International conference on electrical engineering, computer science and informatics (EECSI),* 315–319. Yogyakarta, Indonesia: *IEEE*. https://doi.org/10.1109/EECSI63442.2024.10776530 [in English].
10. Dewi, R.S., Zahrah, F.A., Nugraha, D.A., Prabowo, P.S., Safitri, A., & Jayadi, P. (2024). 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),* 175–178. Palembang, Indonesia: *IEEE*. https://doi.org/10.1109/ICECOS63900.2024.10791154 [in English].
11. Nassif, A.B., AbuTalib, M., & Capretz, L.F. (2020). Software effort estimation from Use Case diagrams using nonlinear regression analysis. In *IEEE Canadian conference on electrical and computer engineering,* 1–4. London, ON, Canada: *IEEE*. https://doi.org/10.1109/CCECE47787.2020.9255712 [in English].
12. Sahoo, P., Behera, D.K., Mohanty, J.R., & Kumar Dash, C.S. (2022). Effort estimation of software products by using UML sequence models with regression analysis. In *The 2022 OITS International Conference on Information Technology (OCIT),* 97–101. Bhubaneswar, India, Los Alamitos: *IEEE*. https://doi.org/10.1109/OCIT56763.2022.00028 [in English].
13. Manisha, & Rishi, R. (2021). Early size estimation using machine learning. In *the 2021 8th International conference on computing for sustainable global development (INDIACom),* 757–762. New Delhi, India, Los Alamitos: *IEEE*. https://doi.org/10.1109/INDIACom51348.2021.00135 [in English].
14. Nhung, H.L.T.K., Hai, V.V., Silhavy, R., Prokopova, Z., & Silhavy, P. (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 [in English].

Приходько Сергій Борисович – д.т.н., професор, завідувач кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування імені адмірала Макарова. E-mail: sergiy.prykhodko@nuos.edu.ua, ORCID: 0000-0002-2325-018X.

Шутко Іван Сергійович – аспірант кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування імені адмірала Макарова. E-mail: ishutko@gmail.com, ORCID: 0000-0001-8584-113X.

Prykhodko Sergiy Borisovich – Doctor of Technical Sciences, Professor, Head of the Department of Software for Automated Systems of Admiral Makarov National University of Shipbuilding. E-mail: sergiy.prykhodko@nuos.edu.ua, ORCID: 0000-0002-2325-018X.

Shutko Ivan Serhiyovych – Postgraduate Student at the Department of Software for Automated Systems of Admiral Makarov National University of Shipbuilding. E-mail: ishutko@gmail.com, ORCID: 0000-0001-8584-113X.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-18