DOI https://doi.org/10.15589/znp2021.1(484).13
УДК 004.412:519.237.5

ESTIMATING THE SIZE OF OPEN-SOURCE PHP-BASED APPS BY NONLINEAR REGRESSION MODELS WITH VARIOUS FACTORS

ОЦІНЮВАННЯ РОЗМІРУ PHP-ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ ЗА НЕЛІНІЙНИМИ РЕГРЕСІЙНИМИ МОДЕЛЯМИ З РІЗНИМИ ФАКТОРАМИ

Sergiy B. Prykhodko
sergiy.prykhodko@nuos.edu.ua
ORCID: 0000-0002-2325-018X
Mykhaylo V. Vorona
mvl.vorona@gmail.com
ORCID: 0000-0003-4288-0096

С. Б. Приходько,
докт. техн. наук, професор
М. В. Ворона,
аспірант

Admiral Makarov National University of Shipbuilding, Mykolaiv
Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв

Abstract. The problem of estimating the software size in the early stage of a software project is important because a software size estimate is used for predicting the software development efforts, including open-source PHP-based apps. The purpose of the work is to increase the prediction accuracy of early software size estimation of open-source PHP-based apps. The object of study is the process of estimating the software size of open-source PHP-based apps. The subject of study is the three-factor nonlinear regression models with various factors to estimate the software size of open-source PHP-based apps. To build the three-factor nonlinear regression models we use the technique based on the multivariate normalizing transformations and prediction intervals. These models are constructed based on the Johnson four-variate normalizing transformation for S_B family of the non-Gaussian data set from 44 apps hosted on GitHub. The data set was obtained using the PhpMetrics tool (https://phpmetrics.org/). The three-factor nonlinear regression models are built around the metrics of class diagrams: the number of classes, the average number of methods per class, the sum of average afferent coupling and average efferent coupling per class, DIT (depth of inheritance tree) mean per class. To compare the prediction accuracy of the three-factor nonlinear regression models we used the well-known prediction accuracy metrics such as a 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). The nonlinear regression model constructed around the number of classes, the average number of methods per class, DIT mean per class has the larger PRED(0.25) value and about the same values of R^2 and MMRE that the model in which the third factor is the sum of average afferent coupling and average efferent coupling per class. The scientific novelty of obtained results is that the three-factor nonlinear regression model for estimating the software size of open-source PHP-based apps has been improved by introducing a new factor – the DIT mean per class. This allowed us to increase the PRED(0.25) value by 8%. The practical importance of obtained results is that the software realizing the constructed model is developed in the sci-language for Scilab.
Key words: software size estimation; PHP-based app; nonlinear regression model; normalizing transformation; non-Gaussian data.

Анотація. Проблема оцінювання розміру програмного забезпечення (ПЗ) на ранній стадії програмного проекту є важливою, оскільки оцінка розміру програмного забезпечення використовується для прогнозування трудомісткості розробки ПЗ, включаючи PHP-застосунки з відкритим кодом. Метою роботи є підвищення точності оцінювання розміру PHP-застосунків з відкритим кодом. Об'єктом дослідження є процес оцінювання розміру PHP-застосунків з відкритим кодом. Предметом дослідження є трьох-факторні моделі нелінійної регресії з різними факторами для оцінювання розміру PHP-застосунків з відкритим кодом. Для побудови трьох-факторних моделей нелінійної регресії ми використовуємо метод, заснований на багатовимірних нормалізуючих перетвореннях та інтервалах прогнозування. Ці моделі побудовані на основі чотирьох-вимірного перетворенні Джонсона для сімейства S_B негаусового набору даних із 44 застосунків, розміщених на GitHub. Набір даних був отриманий за допомогою інструмента PhpMetrics (https://phpmetrics.org/). Трьох-факторні моделі нелінійної регресії побудовані за метриками діаграми класів: кількість класів, середня кількість методів на клас, сума

середнього аферентного та еферентного зв'язків на клас, середнє значення DIT (глибина дерева успадкування) на клас. Для порівняння точності прогнозування трьох-факторних нелінійних регресійних моделей ми використовували відомі показники точності прогнозування, такі як множинний коефіцієнт детермінації R^2, середня величина відносної похибки MMRE та відсоток прогнозування на рівні величини відносної помилки 0,25, PRED(0,25). Нелінійна регресійна модель, що побудована навколо кількості класів, середньої кількості методів на клас, середнього значення DIT на клас, має більше значення PRED(0,25) та приблизно однакові значення R^2 та MMRE, що і модель, в якій третім фактором є сума середнього аферентного та еферентного зв'язків на клас. Наукова новизна отриманих результатів полягає в тому, що удосконалена трьох-факторна нелінійна регресійна модель для оцінювання розміру PHP-застосунків з відкритим кодом шляхом введення нового фактору – середнього значення DIT на клас. Це дозволило збільшити значення PRED(0,25) на 8%. Практична значимість отриманих результатів полягає у розробці ПЗ, що реалізує побудовану модель, sci-мовою для Scilab.
Ключові слова: оцінювання розміру програмного забезпечення; PHP застосунок; нелінійна регресійна модель; нормалізуюче перетворення; негаусові дані.

ПОСТАНОВКА ЗАДАЧІ

Як відомо, PHP – це популярна мова сценаріїв загального призначення, яка особливо підходить для веб-розробки (https://www.php.net/). Однак ця мова дозволяє не тільки швидко писати веб-сторінки, що динамічно генеруються, але і робити набагато більше, включаючи різні фреймворки, конвертери та інші програмні застосунки.

Задача оцінювання розміру програмного забезпечення (ПЗ) на ранній стадії програмного проекту є важливою, оскільки оцінка розміру ПЗ використовується для прогнозування трудомісткості розробки ПЗ за допомогою математичних моделей, таких як COCOMO II [1; 2]. Для цього потрібні відповідні моделі у тому числі і регресійні для оцінювання розміру ПЗ, включаючи PHP-застосунки з відкритим кодом [3–6].

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Незважаючи на достатньо велику кількість існуючих на сьогодні різноманітних методів і моделей для оцінювання розміру ПЗ [3–16], роботи в цьому напрямі не припиняються. Це пов'язано в першу чергу з низькою точністю оцінювання розміру ПЗ на ранніх етапах його розробки. Один із шляхів вирішення цієї проблеми полягає в побудові відповідних моделей для оцінювання розміру ПЗ, яке розробляється як певною мовою програмування [3–6; 8; 11], так і для певного типу застосунків [3–6; 9; 15]. Крім того, для оцінювання розміру ПЗ на ранніх етапах його розробки використовуються метрики UML діаграм, в першу чергу діаграми класів [3–6; 10; 11; 13]. Також, як це зазначено в [5; 6], покращити точність оцінювання розміру ПЗ на ранніх етапах його розробки можна за рахунок відповідних математичних моделей, зокрема нелінійних регресійних моделей, які будуються за допомогою багатовимірних нормалізуючих перетворень та інтервалів передбачення. На відміну від одновимірних, використання багатовимірних нормалізуючих перетворень дозволяє врахувати кореляцію між залежною і незалежними змінними, що і призводить до підвищення точності оцінювання залежної змінної за допомогою відповідної нелінійної регресійної моделі [17].

ВІДОКРЕМЛЕННЯ НЕ ВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

В [5] нелінійна регресійна модель для оцінювання розміру інформаційних PHP-систем з відкритим кодом була побудована із застосуванням чотиривимірного перетворення Джонсона сім'ї S_B на основі трьох метрик діаграми класів, що і в [3; 4]: загальна кількість класів, загальна кількість зв'язків та середня кількість атрибутів на клас. Але для PHP-застосунків з відкритим кодом, що не є інформаційними системами, наприклад, таких як різноманітні фреймворки та конвертори, регресійні моделі можуть залежати в тому числі від інших метрик. Тому в [6] було запропоновано нелінійну регресійну модель для оцінювання розміру PHP-застосунків з відкритим кодом, що не є інформаційними системами, в залежності від трьох факторів: кількості класів; суми середньої кількості класів, на які впливає даний клас (Average Afferent Coupling) і середньої кількості класів, з яких даний клас отримує ефекти (Average Efferent Coupling), та середньої кількості методів. Зазначена модель також була побудована на основі чотиривимірного нормалізуючого перетворення Джонсона сім'ї S_B, що дозволило підвищити достовірність оцінювання залежної змінної нелінійної регресії у порівнянні з використанням одновимірних нормалізуючих перетворень. Але для цієї моделі відсоток прогнозованих результатів, для яких величини відносної помилки менші за 0,25, PRED(0,25) дорівнював всього 68,3%. Це веде до необхідності подальшого удосконалення відповідної моделі для оцінювання розміру PHP-застосунків з відкритим кодом, що не є інформаційними системами.

МЕТОЮ ДОСЛІДЖЕННЯ

Підвищення точності оцінювання розміру PHP-застосунків з відкритим кодом.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єктом дослідження є процес оцінювання розміру PHP-застосунків з відкритим кодом. Предметом

дослідження є трьох-факторні нелінійні регресійні моделі з різними факторами для оцінювання розміру PHP-застосунків з відкритим кодом. Для побудови трьох-факторних нелінійних регресійних моделей ми використовуємо метод, заснований на багатовимірних нормалізуючих перетвореннях та інтервалах передбачення [17]. Суть цього методу [17] є такою. На першому етапі початкові дані перевіряються на наявність викидів і, якщо останні знайдено, то вони відкидаються. Для цього використовується критерій на основі квадрату відстані Махаланобіса для нормалізованих даних із 0,005 рівнем значущості. На другому етапі будується нелінійна регресійна модель із використанням відповідного методу на основі нормалізуючих перетворень [5]. На третьому етапі визначаються границі інтервалу передбачення нелінійної регресії для рівня значущості, що дорівнює 0,05, за відповідним методом [5]. На останньому четвертому етапі перевіряють, чи є серед даних, за якими будувалася нелінійна регресійна модель такі, що виходять за границі інтервалу передбачення. Та, якщо відповідні дані знайдено, то вони відкидаються, і ми повторюємо знову всі етапи, починаючи з першого, для нових даних. Якщо таких викидів не було, то повторення етапів завершується, відповідна нелінійна регресійна модель побудована. Як і в [5, 6, 17], у якості багатовимірного нормалізуючого перетворення ми застосовуємо чотирьох-вимірне перетворення Джонсона для сімейства S_B.

ОСНОВНИЙ МАТЕРІАЛ

Для досягнення зазначеної мети ми брали емпіричні дані з метрик 44 PHP-застосунків з відкритим кодом, що наведені в [6]. Ці дані були доповнені значеннями ще однієї метрики – це середнє значення DIT (глибина дерева успадкування) на клас. Для побудови трьох-факторних нелінійних регресійних моделей для оцінювання розміру PHP-застосунків з відкритим кодом в залежності від різних факторів було застосовано метод покращення нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень та інтервалів передбачення [17]. Трьох-факторні моделі нелінійної регресії були побудовані за такими метриками діаграми класів: кількість класів X_1, середня кількість методів на клас X_2, сума середнього аферентного та еферентного зв'язків на клас X_3, середнє значення DIT на клас X_4. Ми побудували три таких моделі в залежності від трьох різних факторів: першу – в залежності від X_1, X_2 та X_3, другу – в залежності від X_1, X_2 та X_4, і третю – в залежності від X_1, X_3 та X_4.

Спочатку ми перевірили наші чотиривимірні дані на наявність багатовимірних відхилень. Але перед цим ми перевірили нормальність цих багатовимірних даних, оскільки добре відомі статистичні методи (наприклад, виявлення багатовимірних викидів на основі квадрату відстані Махаланобісу) використовуються для виявлення викидів у багатовимірному наборі даних за умови, що дані є гаусовими. Ми застосували тест на нормальність багатовимірних даних, заснований на вимірах багатовимірних асиметрії та ексцесу [18]. Згідно з цим тестом розподіл чотиривимірних даних не є гаусовим, оскільки статистика тесту на багатовимірну асиметрію цих даних перевищує значення квантилю розподілу χ^2, що становить 45,31 для 20 ступенів свободи та 0,001 рівень значущості. Аналогічно, статистика тесту для багатовимірного ексцесу перевищує значення квантилю Гауса, який становить 30,46 для середнього і вибіркової дисперсії, що дорівнюють відповідно 24 і 4,36, та 0,001 рівня значущості. Ці результати вказують нам на необхідність подальшого застосування методу для визначення багатовимірних викидів у багатовимірних негаусових даних на основі багатовимірних нормалізуючих перетворень. Що ми і зробили далі згідно з [17].

Також майбутні фактори ми перевірили на наявність мультиколінеарності. Наявність мультиколінеарності ми визначали за коефіцієнтами впливу дисперсії (VIFs) серед майбутніх факторів в моделі множинної лінійної регресії. Для моделі множинної лінійної регресії з k-факторами $X_i, i = 1, 2, \ldots, k$, VIFs – це діагональні елементи оберненої коваріаційної матриці $k \times k$ k-факторів [19]. Значення VIFs більше за 10 часто сприймаються як сигнал, що дані мають проблеми з мультиколінеарністю. Для наших даних значення VIFs знаходяться у межах від 1 до 5, тому мультиколінеарності немає.

Нелінійна регресійна модель для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_2 та X_3 має вигляд [5]

$$Y = \hat{\varphi}_Y + \hat{\lambda}_Y \left[1 + e^{-(\hat{Z}_Y + \varepsilon - \hat{\gamma}_Y)/\hat{\eta}_Y}\right]^{-1} \qquad (1)$$

де $\hat{Z}_Y = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3$; ε – випадкова величина з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$ з оцінкою середньоквадратичного відхилення $\hat{\sigma}_\varepsilon$ =0,1571; Z_j – випадкова величина з розподілом Гаусу, $Z_j \sim N(0,1)$

$Z_j = \gamma_j + \eta_j \, \text{h} \, \frac{X_j - \varphi_j}{\varphi_j + \lambda_j - X_j}$, $\varphi_j < X_j < \varphi_j + \lambda_j$, $j = 1,2,3$;

оцінки параметрів є такими: $\hat{b}_0 = 0$, $\hat{b}_1 = 1{,}00603$, $\hat{b}_2 = 0{,}533514$, $\hat{b}_3 = -0{,}032605$, $\hat{\gamma}_Y = 3{,}01458$, $\hat{\gamma}_1 = 2{,}87815$, $\hat{\gamma}_2 = 16{,}5902$, $\hat{\gamma}_3 = 0{,}710304$, $\hat{\eta}_Y = 0{,}698176$, $\hat{\eta}_1 = 0{,}652161$, $\hat{\eta}_2 = 2{,}322625$, $\hat{\eta}_3 = 0{,}878073$, $\hat{\varphi}_Y = 0{,}086857$, $\hat{\varphi}_1 = -0{,}063309$, $\hat{\varphi}_3 = 1{,}48156$, $\hat{\varphi}_3 = 1{,}48156$, $\hat{\lambda}_Y = 707{,}4161$, $\hat{\lambda}_1 = 6959{,}670$, $\hat{\lambda}_2 = 11672{,}248$ і $\hat{\lambda}_3 = 11{,}5940$.

Зауважимо, нелінійна регресійна модель (1) для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_2 та X_3 отримана за три ітерації, при чому було три викиди. Тобто ця модель побудована за 41 точками даних.

Нелінійна регресійна модель для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_2 та X_4 також має вигляд (1)

лише з тією різницею, що в (1) індекс 3 потрібно замінити на 4, а відповідні оцінки є такими:
$\hat{\sigma}_\varepsilon$=0,1533, $\hat{b}_0 = 0$, $\hat{b}_1 = 1{,}050805$, $\hat{b}_2 = 0{,}490140$, $\hat{b}_4 = -0{,}011783$, $\hat{\gamma}_Y = 1{,}99105$, $\hat{\gamma}_1 = 2{,}230711$, $\hat{\gamma}_2 = 18{,}2818$, $\hat{\gamma}_4 = 8{,}560105$, $\hat{\eta}_Y = 0{,}612132$, $\hat{\eta}_1 = 0{,}618562$, $\hat{\eta}_4 = 1{,}710109$, $\hat{\eta}_4 = 1{,}710109$, $\hat{\varphi}_Y = 0{,}113140$, $\hat{\varphi}_1 = -0{,}214048$, $\hat{\varphi}_2 = -3{,}303081$, $\hat{\varphi}_4 = 0{,}900$, $\hat{\lambda}_Y = 240{,}494$, $\hat{\lambda}_1 = 3099{,}972$, $\hat{\lambda}_2 = 17083{,}878$ і $\hat{\lambda}_4 = 62{,}6318$.

Зауважимо, нелінійна регресійна модель (1) для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_2 та X_4 отримана за дві ітерації, при чому було два викиди. Тобто ця модель побудована за 42 точками даних.

Нелінійна регресійна модель для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_3 та X_4 також має вигляд лише з тією різницею, що в (1) індекси 2 і 3 потрібно замінити на 3 і 4, а відповідні оцінки є такими:
$\hat{\sigma}_\varepsilon$=0,3882, $\hat{b}_0 = 0$, $\hat{b}_1 = 0{,}885605$, $\hat{b}_3 = -0{,}0227126$, $\hat{b}_4 = -0{,}190975$, $\hat{\gamma}_Y = 1{,}474052$, $\hat{\gamma}_1 = 2{,}364047$, $\hat{\gamma}_3 = 0{,}875184$, $\hat{\gamma}_4 = 6{,}287214$, $\hat{\eta}_Y = 0{,}519960$, $\hat{\eta}_1 = 0{,}728907$, $\hat{\eta}_3 = 0{,}953635$, $\hat{\eta}_4 = 1{,}631979$, $\hat{\varphi}_Y = 0{,}210403$, $\hat{\varphi}_1 = -7{,}908353$, $\hat{\varphi}_3 = 1{,}450813$, $\hat{\varphi}_4 = 0{,}900$, $\hat{\lambda}_Y = 181{,}2127$, $\hat{\lambda}_1 = 2739{,}2845$, $\hat{\lambda}_3 = 12{,}3043$ і $\hat{\lambda}_4 = 20{,}7263$.

Зауважимо, нелінійна регресійна модель (1) для оцінювання розміру PHP-застосунків у тисячах строк коду в залежності від факторів X_1, X_3 та X_4 отримана за дві ітерації, при чому був лише один викид. Тобто ця модель побудована за 43 точками даних.

Для порівняння точності оцінювання розміру PHP-застосунків за допомогою побудованих трьохфакторних нелінійних регресійних моделей в залежності від різних факторів ми використовували відомі показники точності прогнозування, такі як множинний коефіцієнт детермінації R^2, середня величина відносної похибки MMRE та відсоток прогнозування на рівні величини відносної помилки 0,25, PRED(0,25). Ці показники зазвичай використовуються для оцінювання якості прогнозування за допомогою регресійних моделей і в інженерії програмного забезпечення [20; 21]. Допустимі значення MMRE і PRED(0,25) складають не більше 0,25 і не менше 0,75 відповідно. Прийнятне значення R^2 приблизно також ж дорівнює 0,75.

Нелінійна регресійна модель, що побудована навколо кількості класів, середньої кількості методів на клас, суми середнього аферентного та еферентного зв'язків на клас, має наступні значення R^2, MMRE та PRED(0,25): 0,9776, 0,1795 та 0,6829 відповідно. Значення R^2 та MMRE для цієї моделі є кращими у порівнянні з іншими моделями.

Нелінійна регресійна модель, що побудована навколо кількості класів, середньої кількості методів на клас, середнього значення DIT на клас, має наступні значення R^2, MMRE та PRED(0,25): 0,9754, 0,1831 та 0,7381 відповідно. Ця модель має на 8% більше значення PRED(0,25) та приблизно однакові значення R^2та MMRE (з різницею у 0,22% та 2% відповідно), що і попередня модель, в якій третім фактором є сума середнього аферентного та еферентного зв'язків на клас.

Нелінійна регресійна модель, що побудована навколо кількості класів, суми середнього аферентного та еферентного зв'язків на клас, середнього значення DIT на клас має найгірші значення R^2, MMRE та PRED(0,25), які дорівнюють 0,8646, 0,5659 та 0,2558 відповідно. Значення MMRE та PRED(0,25) для цієї моделі вказують на погану точність оцінювання залежної випадкової величини *Y*. Лише значення R^2 є добрим і говорить про можливість використання цієї моделі (1) для оцінювання вибіркового середнього величини *Y*, коли є прямує у нуль.

Підкреслимо, що для нелінійної регресійної моделі, що побудована в залежності від факторів X_1, X_2 та X_4, а саме – кількості класів, середньої кількості методів на клас, середнього значення DIT на клас, ширина інтервалу передбачення кількості тисяч строк коду для PHP-застосунків з розміром понад 40 KLOC (thousand lines of code) є суттєво меншою у порівняні з моделлю, що побудована в залежності від факторів X_1, X_2 та X_3. Так, для застосунку 1 за даними, що наведені в [6, таблиця 2], – X_1=2075, X_2=4,809 та X_3=6,425 для моделі, що побудована в залежності від факторів X_1, X_2 та X_3, маємо такий інтервал передбачення залежної змінної *Y*: [126,334; 267,189]. Причому актуальне значення *Y* для цього застосунку (а це Symfony-master), яке було отримано за допомогою PhpMetrics, дорівнює 174,927 KLOC. Для цього ж застосунку 1 за значеннями факторів X_1=2075, X_2=4,809 та X_4=1,25 для моделі, що побудована в залежності від факторів X_1, X_2 та X_4, маємо такий інтервал передбачення залежної змінної *Y*: [145,090; 199,384], ширина якого на 86% менша за ширину відповідного інтервалу для моделі, що побудована в залежності від факторів X_1, X_2 та X_3. Подібний результат ми маємо і для довірчого інтервалу. Так, для значень зазначених вище факторів застосунку 1 для моделі, що побудована в залежності від факторів X_1, X_2 та X_4, маємо такий довірчий інтервал вибіркового середнього змінної *Y*: [163, 733; 186, 529], ширина якого на 58% менша за ширину відповідного інтервалу для моделі, що побудована в залежності від факторів X_1, X_2 та X_3, якій є таким [162, 487; 217, 004]. Зменшення ширин відповідних інтервалів також вказує на підвищення точності оцінювання розміру PHP-застосунків з відкритим кодом понад 40 KLOC за допомогою нелінійної регресійної моделі, що побудована в залежності від кількості класів, середньої кількості методів на клас, середнього значення DIT на клас.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Отримані результати свідчать про таке. До факторів, які найбільшим чином впливають на точність оцінювання розміру PHP-застосунків з відкритим

кодом, насамперед потрібно віднести кількість класів. У найбільшому впливі цього фактору можна пересвідчитися порівнявши за абсолютною величиною значення оцінок параметрів b_j: значення $\hat{b}_1$ є найбільшим для всіх моделей. Другим за впливом фактором є середня кількість методів на клас. Зазначимо, що неврахування цього фактору у моделі, що була побудована в залежності від факторів X_1, X_2 та X_4, призводить до поганої точності оцінювання залежної випадкової величини Y, на що вказують значення MMRE та PRED(0,25) для цієї моделі. Тому для підвищення точності оцінювання розміру PHP-застосунків з відкритим кодом, кількість строк якого буде перевищувати 40 KLOC, на наш погляд, може використовуватися нелінійна регресійна модель, що побудована в залежності від факторів X_1, X_2 та X_4, а саме – кількості класів, середньої кількості методів на клас, середнього значення DIT на клас. А у разі, коли кількість строк коду PHP-застосунку буде менше за 40 KLOC може бути застосована нелінійна регресійна модель, що побудована в залежності від факторів X_1, X_2 та X_3, а саме – кількості класів, середньої кількості методів на клас, суми середнього аферентного та еферентного зв'язків на клас. У випадку, коли ми заздалегідь не знаємо, чи буде розмір PHP-застосунку більше або менше за 40 KLOC, то, на нашу думку, слід використовувати трьох-факторну нелінійну регресійну модель, що побудована в залежності від кількості класів, середньої кількості методів на клас та середнього значення DIT на клас тому, що ця модель має на 8% більше значення PRED(0,25) та приблизно однакові значення R^2 та MMRE, що і попередня модель, в якій третім фактором є сума середнього аферентного та еферентного зв'язків на клас.

ВИСНОВКИ

У роботі удосконалена трьох-факторна нелінійна регресійна модель для оцінювання розміру PHP-застосунків з відкритим кодом шляхом введення нового фактору – середнього значення DIT на клас. Це дозволило підвищити точність і достовірність відповідного оцінювання у порівнянні з наявними трьох-факторними нелінійними регресійними моделями. У подальшому планується побудова нелінійної регресійної моделі залежно від чотирьох факторів для оцінювання розміру PHP-застосунків.

REFERENCES

[1] Boehm B.W., Abts, C., & Brown, A. W. et al. (2000). *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall PTR.

[2] Yahya, M. A., Ahmad, R., & Lee, S. P. (2008). Effects of software process maturity on COCOMO II's effort estimation from CMMI perspective. Proceedings from RIVFCCT'08: IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies. (pp. 255-262). Ho Chi Minh City, Vietnam. DOI: 10.1109/RIVF.2008.4586364.

[3] Tan, H. B. K., Zhao, Y., & Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models. Proceedings from ICSE'06: *Software Engineering: the 28th International Conference*. (pp. 321-330). Shanghai, China. DOI: 10.1145/1134285.1134331.

[4] Tan, H. B. K., Zhao, Y., & Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*, *19 (2)*, article no. 4. DOI: 10.1145/1571629.1571630.

[5] Prykhodko, N.V., & Prykhodko, S.B. (2018). Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*, *40 (6).* 101-110. DOI: 10.15407/emodel.40.06.101.

[6] Prykhodko, S. B., Prykhodko, N. V., Farionova T. A., & Vorona M.V. (2020). Trokhfaktorna neliniina rehresiina model dlia otsiniuvannia rozmiru Php-zastosunkiv z vidkrytym kodom кодом. [Three-factor non-linear regression model to estimate the size of open source PHP-based applications]. *Naukovyi zhurnal «Vcheni zapysky Tavriiskoho natsionalnoho universytetu imeni V. I. Vernadskoho. Seriia: Tekhnichni nauky» – Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences, 31, (70), no. 1*. 124-131. [in Ukrainian]. DOI: 10.32838/2663-5941/2020.1-1/23.

[7] Hastings, T. E., & Sajeev, S. M. (2001). A vector-based approach to software size measurement and effort estimation. *IEEE Trans. Softw. Eng*., 27 *(4)*, 337–350.

[8] Kaczmarek, J., & Kucharski M. (2004). Size and effort estimation for applications written in Java. *Information and Software Technology, 46 (9)*, 589-601. DOI: 10.1016/j.infsof.2003.11.001.

[9] Lind, K., Heldal, R., Harutyunyan T., & Heimdahl, T. (2011) CompSize: Automated Size Estimation of Embedded Software Components. Proceedings from Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement. (pp. 86-95). Nara, Japan. DOI: 10.1109/IWSM-MENSURA.2011.49.

[10] Zifen, Y. (2012). An improved software size estimation method based on object-oriented approach. Proceedings from EEESYM*'12*: IEEE Symposium on Electrical & Electronics Engineering. (pp. 615-617). Kuala Lumpur, Malaysia. DOI: 10.1109/EEESym.2012.6258733.

[11] Kiewkanya, M., & Surak, S. (2016). Constructing C++ software size estimation model from class diagram. Proceedings from CSSE'16: *Computer Science and Software Engineering: 13th International Joint Conference*. (pp. 1-6). Khon Kaen, Thailand. DOI: 10.1109/JCSSE.2016.7748880.

[12] Cheng, Z., Shensi, T., Wenkai, M., Yang, Z., Yong, X., & Beijun, S. (2016) Esse: An early software size estimation method based on autoextracted requirements features. Proceedings from Internetware'16: *the 8th Asia-Pacific Symposium on Internetware*. (pp. 112–115), New York, NY, USA.

[13] Nassif, A. B., AbuTalib, M., & Capretz, L. F. (2020). Software Effort Estimation from Use Case Diagrams Using Nonlinear Regression Analysis, Proceedings from CCECE'20: *IEEE Canadian Conference on Electrical and Computer Engineering.* (pp. 1-4). London, ON, Canada. DOI: 10.1109/CCECE47787.2020.9255712.

[14] Zaw, T., Hlaing, S. Z., Myint Lwin, M., & Ochimizu, K. (2019) The Measurement of Software Size based on Generation Model using COSMIC FSM. Proceedings from ICSEC'19: *23rd International Computer Science and Engineering Conference*. (pp. 373-378). Phuket, Thailand. DOI: 10.1109/ICSEC47112.2019.8974688.

[15] Neyveli, V.R. N., Sivakumar, S. S., Arunagiri, D., Arumugam, C., & Veeramani, A. M. (2019) An Approach to Estimate the Size of Web Application Using IFML User Interface Model. Proceedings from AICAI'19: *Amity International Conference on Artificial Intelligence.* (pp. 292-295). Dubai, United Arab Emirates. DOI: 10.1109/AICAI.2019.8701268.

[16] Zhang, K., Wang, X., Ren, J., & Liu, C. (2020). Efficiency Improvement of Function Point-Based Software Size Estimation with Deep Learning Model. Proceedings in IEEE access. DOI: 10.1109/ACCESS.2020.2998581.

[17] Prykhodko, S., & Prykhodko, N. (2020). Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations. Proceedings from MODS'2020: *Mathematical Modeling and Simulation of Systems. Advances in Intelligent Systems and Computing, 1265.* (pp. 166-174). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58124-4_16

[18] Mardia. K. V. (1970) Measures of multivariate skewness and kurtosis with applications. *Biometrika. 57*, 519–530. DOI: 10.1093/biomet/57.3.519

[19] Chatterjee, S., & Price, B. (1977) *Regression analysis by example*. New York: John Wiley & Son.

[20] Foss, T., Stensrud, E., Kitchenham, B., & Myrtveit, I. (2003) A simulation study of the model evaluation criterion MMRE. *IEEE Transactions on software engineering*, *11 (29)*, 985–995.

[21] Port, D., & Korte, M. (2008) Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. (pp. 51-60). New York: ACM.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Boehm B.W., Abts, C., & Brown, A. W. et al. (2000). *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall PTR.

[2] Yahya, M. A., Ahmad, R., & Lee, S. P. (2008). Effects of software process maturity on COCOMO II's effort estimation from CMMI perspective. Proceedings from RIVFCCT'08: IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies. (pp. 255-262). Ho Chi Minh City, Vietnam. DOI: 10.1109/RIVF.2008.4586364.

[3] Tan, H. B. K., Zhao, Y., & Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models. Proceedings from ICSE'06: *Software Engineering: the 28th International Conference*. (pp. 321-330). Shanghai, China. DOI: 10.1145/1134285.1134331.

[4] Tan, H. B.K., Zhao, Y., & Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. *Transactions on Software Engineering and Methodology*, *19 (2)*, article no. 4. DOI: 10.1145/1571629.1571630.

[5] Prykhodko, N. V., & Prykhodko, S. B. (2018). Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*, *40 (6).* 101-110. DOI: 10.15407/emodel.40.06.101.

[6] Приходько, С. Б., Приходько, Н. В., Фаріонова, Т. А., Ворона, М. В. (2020). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом. *Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки», 31, (70), no. 1*. 124-131. DOI: 10.32838/2663-5941/2020.1-1/23.

[7] Hastings, T. E., & Sajeev, S. M. (2001). A vector-based approach to software size measurement and effort estimation. *IEEE Trans. Softw. Eng*., 27 *(4)*, 337–350.

[8] Kaczmarek, J., & Kucharski M. (2004). Size and effort estimation for applications written in Java. *Information and Software Technology, 46 (9)*, 589-601. DOI: 10.1016/j.infsof.2003.11.001.

[9] Lind, K., Heldal, R., Harutyunyan T., & Heimdahl, T. (2011) CompSize: Automated Size Estimation of Embedded Software Components. Proceedings from Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement. (pp. 86-95). Nara, Japan. DOI: 10.1109/IWSM-MENSURA.2011.49.

[10] Zifen, Y. (2012). An improved software size estimation method based on object-oriented approach. Proceedings from EEESYM*'12*: IEEE Symposium on Electrical & Electronics Engineering. (pp. 615-617). Kuala Lumpur, Malaysia. DOI: 10.1109/EEESym.2012.6258733.

[11] Kiewkanya, M., & Surak, S. (2016). Constructing C++ software size estimation model from class diagram. Proceedings from CSSE'16: *Computer Science and Software Engineering: 13th International Joint Conference*. (pp. 1-6). Khon Kaen, Thailand. DOI: 10.1109/JCSSE.2016.7748880.

[12] Cheng, Z., Shensi, T., Wenkai, M., Yang, Z., Yong, X., & Beijun, S. (2016) Esse: An early software size estimation method based on autoextracted requirements features. Proceedings from Internetware'16: *the 8th Asia-Pacific Symposium on Internetware*. (pp. 112–115), New York, NY, USA.

[13] Nassif, A. B., AbuTalib, M., & Capretz, L. F. (2020). Software Effort Estimation from Use Case Diagrams Using Nonlinear Regression Analysis, Proceedings from CCECE'20: *IEEE Canadian Conference on Electrical and Computer Engineering.* (pp. 1-4). London, ON, Canada. DOI: 10.1109/CCECE47787.2020.9255712.

[14] Zaw, T., Hlaing, S. Z., Myint Lwin, M., & Ochimizu, K. (2019) The Measurement of Software Size based on Generation Model using COSMIC FSM. Proceedings from ICSEC'19: *23rd International Computer Science and Engineering Conference*. (pp. 373-378). Phuket, Thailand. DOI: 10.1109/ICSEC47112.2019.8974688.

[15] Neyveli, V.R. N., Sivakumar, S. S., Arunagiri, D., Arumugam, C., & Veeramani, A. M. (2019) An Approach to Estimate the Size of Web Application Using IFML User Interface Model. Proceedings from AICAI'19: *Amity International Conference on Artificial Intelligence*. (pp. 292-295). Dubai, United Arab Emirates. DOI: 10.1109/AICAI.2019.8701268.

[16] Zhang, K., Wang, X., Ren, J., & Liu, C. (2020). Efficiency Improvement of Function Point-Based Software Size Estimation with Deep Learning Model. Proceedings in IEEE access. DOI: 10.1109/ACCESS.2020.2998581.

[17] Prykhodko, S., & Prykhodko, N. (2020). Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations. Proceedings from MODS'2020: *Mathematical Modeling and Simulation of Systems. Advances in Intelligent Systems and Computing, 1265.* (pp. 166-174). Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58124-4_16.

[18] Mardia. K. V. (1970) Measures of multivariate skewness and kurtosis with applications. *Biometrika. 57*, 519–530. DOI: 10.1093/biomet/57.3.519.

[19] Chatterjee, S., & Price, B. (1977) *Regression analysis by example*. New York: John Wiley & Son.

[20] Foss, T., Stensrud, E., Kitchenham, B., & Myrtveit, I. (2003) A simulation study of the model evaluation criterion MMRE. *IEEE Transactions on software engineering*, *11 (29)*, 985–995.

[21] Port, D., & Korte, M. (2008) Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. (pp. 51-60). New York: ACM.

Дата надходження статті до редакції: 09.03.2021
Дата затвердження статті до друку: 23.03.2021