

Д. В. БРЕСЛАВСЬКИЙ, О. О. БРОВАРНИК, В. О. МЕТЕЛЬОВ, О. А. ТАТАРИНОВА

АНАЛІЗ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЧНОГО МОНІТОРИНГУ БАЗ ДАНИХ З РУХУ НАСЕЛЕННЯ

Представлено підхід та результати аналізу надійності програми збирання даних про рух населення у формі утворення міст. Надано стислий опис даної програми, що працює з моніторингом відкритих баз даних Української Вікіпедії та програмного застосунку OpenStreetMap. Програму створено на мові програмування Python з використанням бібліотек Requests, BeautifulSoup, Pandas, що забезпечує ефективну роботу з HTML-контентом веб-сторінок, обробку геоданих та структуроване зберігання інформації. Валідація отриманих результатів проводиться шляхом перевірки коректності форматів назв, координат, історичних чисельних даних. Розробку, тестування та візуалізацію процесу виконано за допомогою Jupyter Notebook. Розглянуто дані з виявлення числа дефектів, отриманих при розробці програмного засобу, в залежності від часового інтервалу тестування. Для розрахункового аналізу використано модель Желінські-Моранди, надано її математичний опис. Параметри моделі визначено за допомогою розв'язання системи нелінійних рівнянь ітераційним методом. Представлено отримані розрахункові залежності від часу тестування інтенсивності відмов та функції надійності програми збирання даних. Виконано порівняння отриманих даних попереднього розрахункового моделювання надійності з даними, отриманими при тестуванні програми, що розглядається. Визначено час забезпечення необхідної надійності програмного засобу. Показано, що використання описаного підходу та моделі Желінські-Моранди надає задовільний ступінь опису поведінки програмного засобу при тестуванні, що дозволяє визначити час забезпечення необхідної надійності його роботи. Зроблено висновок, що використання представленого підходу та отримані при цьому результати дозволяють вважати можливим його застосування й при проектуванні подібних програмних засобів, присвячених аналізу полів баз даних.

Ключові слова: надійність, інтенсивність відмов, програма, вебзастосунок, тестування, бази даних.

D. BRESLAVSKY, O. BROVARNYK, V. MIETIELOV, O. TATARINOVA

RELIABILITY ANALYSIS OF SOFTWARE FOR AUTOMATIC MONITORING OF POPULATION MOVEMENT DATABASES

The approach and results of the reliability analysis of the program for collecting data on population movement in the form of city formation are presented. A brief description of this program, which operates with monitoring of open databases of Ukrainian Wikipedia and the OpenStreetMap software application, is provided. The program was developed by use of the Python programming language as well as the Requests, BeautifulSoup, Pandas libraries, which provides effective work with HTML content of web pages, geodata processing, and structured information storage. Validation of the received results is carried out by checking the correctness of the formats of names, coordinates, and historical numerical data. The development, testing, and visualization of the process were performed using Jupyter Notebook. The data on the detection of the number of defects obtained during the development of the software tool, depending on the testing time interval, were considered. The Jelinski-Moranda model was used for computational analysis, and its mathematical description was provided. The model parameters were determined by solving a system of nonlinear equations by the iterative method. The obtained calculated dependences on the testing time of the failure intensity and the reliability function of the data collection program are presented. The obtained data of the preliminary computational modeling of reliability were compared with the data obtained during testing of the program under consideration. The time for ensuring the required reliability of the software tool was determined. It was shown that the use of the described approach and the Jelinski-Moranda model provides a satisfactory degree of description of the behavior of the software tool during testing, which allows determining the time for ensuring the required reliability of its operation. It was concluded that the use of the presented approach and the results obtained at the same time allow us to consider its application possible in the design of similar software tools dedicated to the analysis of database fields.

Key words: reliability, failure rate, program, web application, testing, databases.

Вступ. Визначення надійності та безвідмовної роботи програмного забезпечення (ПЗ) є важливим завданням для будь-якого колективу його розробників. Надійне ПЗ забезпечує подальшу його безпеку, конкурентоспроможність та переваги на ринку. Тому крім класичних засобів аналізу якості ПЗ останнім часом все більш інтенсивним є використання спеціалізованих методів оцінювання надійності, заснованих на поєднанні методів теорії ймовірності, математичної статистики та теорії випадкових процесів [1-3]. Безпосередньо теорія надійності програмних застосунків та систем розробляється починаючи з класичних робіт Желінські та Моранди [4-6]. На теперішній час основні розроблені підходи, математичні методи та моделі для оцінювання надійності ПЗ підсумовано роботами Лю [7], Ямади [8] та інших дослідників [1; 2].

Класична модель Желінські та Моранди ефективно використовується й у теперішній час [3]. Наприклад, у роботі [9] представлено узагальнену її форму, отриману шляхом додавання додаткового

параметра до функції коефіцієнта ризику моделі. Автори наголошують, що застосування такого параметру може призводити до більшої гнучкості моделі та збільшувати можливість отримання найкращого прогнозу залишкових помилок для даних, отриманих з різних проєктів. В роботі [10] представлено модель надійності ПЗ, яка враховує виявлення, залежне від несправностей, недосконале усунення несправностей та максимальну кількість несправностей, які можуть бути присутніми в системі.

Одним з важливих видів ПЗ є застосунки, що безпосередньо працюють з базами даних, в тому числі й розташованими онлайн. Вимоги до надійності такого ПЗ можуть бути підвищеними завдяки комбінації впливів внутрішніх дефектів та зовнішніх акцій [11]. Це питання стало до центру уваги дослідників відносно нещодавно, та дослідження у даному напрямі починають розвиватись. У роботі [12] розглядаються методи забезпечення якості програмних застосунків та онлайн-сервісів у вищих навчальних закладах (ВНЗ). Аналізуються такі підходи до тестування: аналізи

продуктивності, сумісності, безпеки та зручності використання. Запропоновано рішення для зменшення вразливостей та забезпечення надійності та функціональності програмного забезпечення. Наголошується, що безперервне тестування та інтегрована безпека мають вирішальне значення для забезпечення надійності ПЗ університетських систем. Автором роботи [13] відмічається, що при постійному змінюванні характеру управління базами даних відмовостійкість та надійність є першорядними для забезпечення роботи. У статті досліджується застосування методів штучного інтелекту (ШІ) для покращення механізмів виявлення та відновлення помилок у базах даних високої доступності. Отримані результати демонструють значне скорочення часу простою. Відмічається, що адаптивність структури дозволяє їй навчатися на історичних моделях помилок, покращуючи свої прогностичні можливості з часом. У монографії [14] підкреслюється, що питання забезпечення надійності програмних систем, які оперують з базами даних, на теперішній момент продовжує залишатись одним з першочергових.

Дану статтю присвячено викладенню результатів проведеного аналізу надійності ПЗ для автоматичного моніторингу баз даних з руху населення. Надано стислий опис розробленого ПЗ.

Опис програми, що аналізується. Як приклад, буде розглянуто алгоритм та програму для автоматичного збирання даних про рік утворення населених пунктів України [15]. Використовується інформація з баз даних Української Вікіпедії [16] та програмного застосунку OpenStreetMap [17]. Основними даними для збирання (вихідними даними) є назва населеного пункту, географічні координати та рік його заснування. Вхідними даними є списки посилань на необхідні розділи тематичних збірників наведених вище баз даних, які представляються у вигляді csv-файлу з інформацією про населені пункти.

Для реалізації завдання використано мову програмування Python, з використанням бібліотек Requests, BeautifulSoup, Pandas [18]. Цей набір інструментів забезпечує ефективну роботу з HTML-контентом веб-сторінок, обробку геоданих та структуроване зберігання інформації. Для зручності розробки, тестування та візуалізації процесу використовувався Jupyter Notebook.

Алгоритм роботи програми може бути представленим наступними ключовими етапами:

- програма збирає посилання на необхідні сторінки Української Вікіпедії, з тематичних сторінок-збірників та бази OpenStreetMap;

- використовуючи бібліотеку BeautifulSoup, здійснюється пошук та вилучення даних (назва, координати, рік заснування) із HTML-коду записів;

- проводиться валідація отриманих даних, перевіряється коректність форматів назв, координат, року заснування. За необхідності здійснюється коригування або відсіювання помилкових та некоректних записів. Валідація включає перевірку на відповідність типу даних (рядок, ціле або дійсне число), діапазон значень (наприклад, координати повинні відповідати допустимим географічним межах, рік заснування не може бути у майбутньому) та формат запису;

- отримані дані зберігаються у вигляді структурованих таблиць за допомогою бібліотеки Pandas в форматі csv.

Програма забезпечує гнучкість у налаштуванні параметрів пошуку і легко адаптується до нових джерел даних та форматів сторінок.

HTML-сторінки відрізняються за структурою, тому для пошуку необхідної інформації додатково використовуються регулярні вирази, які істотно збільшують кількість зібраної інформації.

Приклад зібраних даних зображено на рисунку 1.

city	founded	url	translite	region	region_ids	coordinate	latitude	longitude
Підгородці	1397	https://uk.wikipedia	Pidhorodtsi	Кількість наявного та постійного	49°09'29"N 23°2'49.158055555555	23.424166666666668		
Скопе	1397	https://uk.wikipedia	Skole	Львівська облас	12.0	49°02'16"N 23°3'49.037777777777	23.513888888888889	
Сокільники (Льв)	1397	https://uk.wikipedia	Sokilnyky (Lviv)	Львівська облас	12.0	49°46'37"N 23°5'49.776944444444	23.961388888888887	
Тухля (Україна)	1397	https://uk.wikipedia	Tukhlya (Ukraine)	Львівська облас	12.0	48°55'05"N 23°2'48.918055555555	23.471944444444443	
Бистрий	1398	https://uk.wikipedia	Bystryi	Коди автоматичного мікроміс	49°14'36"N 23°3'49.243333333333	23.593888888888888		
Джурків	1398	https://uk.wikipedia	Dzhurkiv	Івано-Франківс	8.0	48°38'N 25°12'E 48.633333333333	25.2	
Мацошин	1398	https://uk.wikipedia	Matsoshyn	Львівська облас	12.0	50°01'29"N 23°5'50.024722222222	23.9925	
Мокротин	1398	https://uk.wikipedia	Mokrotyyn	Львівська облас	12.0	50°00'47"N 23°5'50.013055555555	23.898333333333333	
Пищаче	1398	https://uk.wikipedia	Pyshchache	Івано-Франківс	8.0	48°36'44"N 25°1'48.612222222222	25.188888888888889	
Стара Скварява	1398	https://uk.wikipedia	Stara Skvariava	Львівська облас	12.0	50°02'19"N 23°5'50.038611111111	23.944722222222222	
Верб (Новгоро)	1399	https://uk.wikipedia	Verba (Novhorod)	Чернігівська об	24.0	51°42'02"N 32°4'51.700555555555	32.7925	
Воля-Добростаї	1399	https://uk.wikipedia	Voila-Dobrostan	Львівська облас	12.0	49°53'32"N 23°3'49.892222222222	23.636944444444445	

Рис. 1 – Приклад зібраних даних щодо міст України

Більш детальний опис програми надано у роботі [15].

Опис процесу тестування. Під час розробки будь-якого програмного продукту важливим є етап тестування, він дозволяє виявити та усунути помилки до моменту початку використання програми. Особливістю такого тестування є поступове зменшення кількості виявлених помилок, що пов'язано

з постійним вдосконаленням та відлагодженням програмного коду. Такий процес й обумовлює процес постійного зростання надійності ПЗ [8].

На початковому етапі розробки кількість помилок є найбільшою, оскільки значна частина функціоналу тільки створюється та вперше перевіряється. Через деякий час їх кількість суттєво знижується завдяки поступовому виправленню виявлених помилок.

Процес тестування та виправлення помилок у

даному конкретному випадку аналізу надійності програми для збирання даних щодо часу утворення міст України виявився таким, що відповідає класичним уявленням [8]. Отриману при тестуванні залежність від часу (годин) числа визначених та виправлених помилок представлено на рис. 2 (точки), тобто має місце зменшення кількості помилок залежно від тривалості розробки та тестування.

Постановка задачі та визначення надійності програми збирання даних. Аналізується надійність роботи описаної вище програми отримання даних про рік утворення населених пунктів України. Для аналізу надійності залучимо класичний підхід з використанням моделі Джелінські-Моранди [4] (ДМ), згідно якому можливо провести моделювання числа інтенсивності відмов (числа дефектів в одиницю часу), густини ймовірності та інших статистичних характеристик процесу, що розглядається.

У моделях, що базуються на марковських процесах [19], передбачається, що кількість дефектів, виявлених у ПЗ, у певний момент часу залежить від поведінки системи й подається у вигляді стаціонарного ланцюга Маркова. При цьому кількість дефектів скінченна, але є невідомою величиною, що задається для моделі у вигляді сталої. Інтенсивність відмов у ПЗ або швидкість проходження ланцюгом залежить лише від кількості дефектів, які залишилися в ПС.

Згідно з представленим у роботі [4] теоретичним підходом, постійна частота відмов програмного забезпечення моделі ДМ на i -му інтервалі відмов визначається як

$$\lambda(t_i) = \varphi[N - (i - 1)], \quad (1)$$

де φ - постійна інтенсивність відмов; N - параметр, що потрібно визначити. t_i - час між двома послідовними $i - 1$ та i відмовами.

Середнє значення функції інтенсивності відмов для цієї моделі визначається як

$$\mu(t_i) = N[1 - \exp(-\varphi t_i)], \quad (2)$$

а сама функція інтенсивності відмов визначається наступним чином

$$\xi(t_i) = N\varphi \exp(-\varphi t_i). \quad (3)$$

Густина ймовірності відмов визначається наступним виразом:

$$f(t_i) = \varphi(N - (i - 1)) \exp(-\varphi[N - (i - 1)t_i]). \quad (4)$$

Функція надійності у моделі ДМ має експоненційний характер:

$$R(t_i) = \exp(-\varphi[N - (i - 1)t_i]). \quad (5)$$

Для визначення значень параметрів N та φ , що визначають процес опису надійності ПЗ, є необхідним розв'язати наступну систему нелінійних алгебраїчних рівнянь:

$$\begin{cases} \sum_{i=1}^n \frac{1}{N - (i - 1)} = \frac{n}{N - \frac{1}{T} \sum_{i=1}^n (i - 1)\Delta t_i} \\ \varphi = \frac{n}{NT - \sum_{i=1}^n (i - 1)\Delta t_i} \end{cases}, \quad (6)$$

де n - проміжків часу тестування, Δt_i - величина проміжку часу вимірювання числа помилок,

$T = \sum_{i=1}^n \Delta t_i$. Система (6) розв'язується за допомогою

методу простої ітерації. Після визначення значень параметрів N та φ є можливим провести аналіз показників надійності з використанням залежностей (1) - (3).

Викладений підхід з застосуванням моделі ДМ було використано для аналізу надійності описаної програми збирання даних. Як видно з рис. 2, загальний час тестування програми T склав 11 год, використовуємо значення $\Delta t_i = 1$ год.

За результатами розв'язання отриманої системи типу (6) визначено значення параметрів моделі ДМ: $N=10$, $\varphi=2.44$. На рис. 2 суцільною кривою представлено результати розрахунку кількості помилок в залежності від часу (інтенсивності відмов). Для порівняння з розрахунковими результатами було обрано дані тестування описаної програми збирання даних, отриманих на етапі її розробки [15]. Порівняння розрахункових та результатів тестування, наданих точками, свідчить про цілком задовільний рівень відмінностей, який не перевищує 15%.

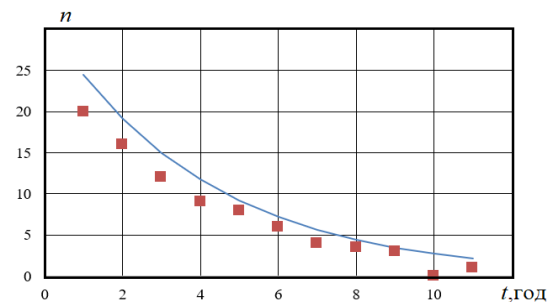


Рис. 2 – Залежність кількості помилок від -часу (інтенсивність відмов). Точки – дані тестування, суцільна крива – розрахунок

Як приклад, представимо також функцію надійності (рис.3):

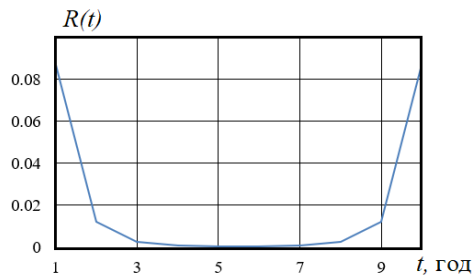


Рис. 3 – Функція надійності

За отриманими результатами розрахунків з використанням моделі ДМ видно, що заздалегідь виконане моделювання надійності підтверджується даними тестування програми збирання даних про рух населення у формі утворення міст та загальний час забезпечення надійності може бути оціненим в 11 год.

Висновки. Стаття містить результати аналізу надійності програми збирання даних про рух населення у формі утворення міст. Надано стислий опис даної програми, що працює з моніторингом відкритих баз даних Української Вікіпедії та програмного застосунку OpenStreetMap. Для розрахункового аналізу використано модель Джелінські-Моранди. Виконано порівняння отриманих даних моделювання з даними, отриманими при тестуванні програми, що розглядається.

Показано, що використання описаного підходу та моделі Джелінські-Моранди надає задовільний ступінь опису поведінки програми при тестуванні, що дозволяє визначити час забезпечення необхідної надійності її роботи. Використання представленого підходу та отримані при цьому результати дозволяють вважати можливим його застосування й при проектуванні подібних програмних засобів, присвячених аналізу полів баз даних та в яких отримання даних про програмні збої та помилки відповідає експоненційному розподілу [8].

Список літератури

1. Atoum I. Challenges of software requirements quality assurance and validation: A systematic literature review / I. Atoum et al. // IEEE Access. – 2021. – Vol. 9. – P. 137613-137634. <https://doi.org/10.1109/ACCESS.2021.3117989>
2. Dohi T. Mathematics in Software Reliability and Quality Assurance / T. Dohi, S. Liu // MDPI-Multidisciplinary Digital Publishing Institute. – 2022. – 218 p. <https://doi.org/10.3390/books978-3-0365-3800-6>
3. Shukla A. System security assurance: A systematic literature review / A. Shukla et al. // Computer Science Review. – 2022. – Vol. 45. – 100496 p. <https://doi.org/10.1016/j.cosrev.2022.100496>
4. Jelinski Z. Software reliability research //Statistical computer performance evaluation / Z. Jelinski, P. Moranda // Academic Press. – 1972. – P. 465-484. <https://doi.org/10.1016/B978-0-12-266950-7.50028-1>
5. Moranda P. B. A Comparison of Software Error-Rate Models / P.B. Moranda // Texas Conference on Computing. – 1975. – P. 2A-6.1-2A-6.9.
6. Van Driel W. D. Prediction of software reliability / W. D. Van Driel, J. W. Bikker, M. Tijink //Microelectronics Reliability. – 2021. – Vol. 119. – 114074 p. <https://doi.org/10.1016/j.microrel.2021.114074>
7. Lyu M. R. T. Software reliability theory / M. R. T. Lyu. // Encyclopedia of Software engineering. – 2002. – Vol. 2. – P. 1611-1630. <https://doi.org/10.1002/0471028959.sof329>
8. Yamada S. Software reliability modeling: fundamentals and applications / S. Yamada // Tokyo : Springer, 2014. – Vol. 5. – 90 p. <https://doi.org/10.1007/978-4-431-54565-1>
9. Al Turk L. I. Jelinski-Moranda software reliability growth model: a brief literature and modification / L. I. Al Turk, E. G. Alsolami // International Journal of Software Engineering & Applications (IJSEA). – 2016. – Vol. 7. – No. 2. <https://doi.org/10.5121/ijsea.2016.7204>
10. Samal U. Redefining software reliability modeling: embracing fault-dependency, imperfect removal, and maximum fault considerations / U. Samal, A. Kumar // Quality Engineering. – 2024. – Vol. 36. – No. 3. – P. 500-509. <https://doi.org/10.1080/08982112.2023.2241067>
11. Wimalasooriya C. A systematic mapping study addressing the reliability of mobile applications: The need to move beyond testing reliability / C. Wimalasooriya et al. // Journal of Systems and Software. – 2022. – Vol. 186. – 111166 p. <https://doi.org/10.1016/j.jss.2021.111166>
12. Pavlova E. Quality Assurance Methods of Software Applications and Online Services in Institutions of Higher Education / E. Pavlova // Yearbook of UNWE. – 2024. – Vol. 2. – No. 2. – P. 83-92.
13. Gadde H. AI-Powered Fault Detection and Recovery in High-Availability Databases / H. Gadde // International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence. – 2024. – Vol. 15. – No. 1. – P. 500-529.
14. Campbell L. Database reliability engineering: designing and operating resilient database systems / L. Campbell, C. Majors // O'Reilly Media, Inc. – 2017.
15. Бреславський Д. В. Використання методу зважених відхилів у формі МСЕ для апроксимації двовимірних розподілів / Д. В. Бреславський, О. О. Броварник // Вісник Національного технічного університету «ХПІ». Серія: Динаміка та міцність машин. – 2024. – No. 1. – С. 36-41.
16. Wikipedia – вільна енциклопедія. – Режим доступу: https://en.wikipedia.org/wiki/Main_Page. – Дата звернення: 1 червня 2025.
17. OpenStreetMap – вільна картографічна платформа. – Режим доступу: <https://www.openstreetmap.org>. – Дата звернення: 1 червня 2025.
18. McClain B. P. Python for Geospatial Data Analysis. / B. P. McClain // O'Reilly Media, Inc., 2022.
19. Capasso V. Introduction to continuous-time stochastic processes. / V. Capasso, D. Bakstein // Springer International Publishing, 2021. – 560 p. <https://doi.org/10.1007/978-3-030-69653-5>

References (transliterated)

1. Atoum I. et al. Challenges of software requirements quality assurance and validation: A systematic literature review. IEEE Access. – 2021. – Vol. 9. – P. 137613-137634. <https://doi.org/10.1109/ACCESS.2021.3117989>
2. Dohi T., Liu S. (ed.). Mathematics in Software Reliability and Quality Assurance. MDPI-Multidisciplinary Digital Publishing Institute, 2022. – 218 p. <https://doi.org/10.3390/books978-3-0365-3800-6>
3. Shukla A. et al. System security assurance: A systematic literature review. Computer Science Review. – 2022. – Vol. 45. – 100496 p. <https://doi.org/10.1016/j.cosrev.2022.100496>
4. Jelinski Z., Moranda P. Software reliability research. //Statistical computer performance evaluation. – Academic Press, 1972. – P. 465-484. <https://doi.org/10.1016/B978-0-12-266950-7.50028-1>
5. Moranda P. B. A Comparison of Software Error-Rate Models. Texas Conference on Computing. – 1975. – P. 2A-6.1-2A-6.9.
6. Van Driel W. D., Bikker J. W., Tijink M. Prediction of software reliability. Microelectronics Reliability. – 2021. – Vol. 119. – 114074 p. <https://doi.org/10.1016/j.microrel.2021.114074>
7. Lyu M. R. T. Software reliability theory. Encyclopedia of Software engineering. – 2002. – Vol. 2. – P. 1611-1630. <https://doi.org/10.1002/0471028959.sof329>
8. Yamada S. Software reliability modeling: fundamentals and applications. Tokyo : Springer, 2014. – Vol. 5. – 90 p. <https://doi.org/10.1007/978-4-431-54565-1>
9. Al Turk L. I., Alsolami E. G. Jelinski-Moranda software reliability growth model: a brief literature and modification. International Journal of Software Engineering & Applications (IJSEA). – 2016. – Vol. 7. – No. 2. <https://doi.org/10.5121/ijsea.2016.7204>

10. Samal U., Kumar A. *Redefining software reliability modeling: embracing fault-dependency, imperfect removal, and maximum fault considerations*. Quality Engineering. – 2024. – Vol. 36. – No. 3. – P. 500-509. <https://doi.org/10.1080/08982112.2023.2241067>
11. Wimalasooriya C. et al. *A systematic mapping study addressing the reliability of mobile applications: The need to move beyond testing reliability* // Journal of Systems and Software. – 2022. – Vol. 186. – 111166 p. <https://doi.org/10.1016/j.jss.2021.111166>
12. Pavlova E. *Quality Assurance Methods of Software Applications and Online Services in Institutions of Higher Education*. Yearbook of UNWE. – 2024. – Vol. 2. – No. 2. – P. 83-92.
13. Gadde H. *AI-Powered Fault Detection and Recovery in High-Availability Databases*. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence. – 2024. – Vol. 15. – No. 1. – P. 500-529.
14. Campbell L., Majors C. *Database reliability engineering: designing and operating resilient database systems*. O'Reilly Media, Inc. – 2017.
15. Breslavsky D. V., Brovarnyk O. O. *Vykorystannia metodu zvaženykh vidkhylyv u formi MSE dlia aproksymatsii dvovymirnykh rozpodiliv*. Bulletin of the National Technical University «KhPI» Series: Dynamics and Strength of Machines. – 2024. – No. 1. – P. 36-41.
16. Wikipedia. Available at: https://en.wikipedia.org/wiki/Main_Page. (accessed 01.06.2025)
17. OpenStreetMap – free cartographic platform. Available at: <https://www.openstreetmap.org>. (accessed 01.06.2025)
18. McClain B. P. *Python for Geospatial Data Analysis*. O'Reilly Media, Inc., 2022.
19. Capasso V., Bakstein D. *Introduction to continuous-time stochastic processes*. Springer International Publishing, 2021. – 560 p. <https://doi.org/10.1007/978-3-030-69653-5>

Надійшла (received) 05.06.2025

Прийнята до друку (accepted) 01.09.2025

Опублікована (published) 03.09.2025

Відомості про авторів/ About the Authors

Бреславський Дмитро Васильович (Breslavsky Dmytro) – доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», професор кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; тел.: (057)7076454; ORCID: <https://orcid.org/0000-0002-3792-5504>; email: Dmytro.Breslavsky@khp.edu.ua

Броварник Олексій Олексійович (Brovarnyk Oleksii) – Національний технічний університет «Харківський політехнічний інститут», аспірант кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; тел.: (057)-707-64-54; ORCID: <https://orcid.org/0009-0001-9689-1850>; e-mail: Oleksii.Brovarnyk@infiz.khpi.edu.ua

Метельов Володимир Олександрович (Mietilov Volodymyr) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», доцент кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; тел.: (057)-707-64-54; ORCID: <https://orcid.org/0000-0002-2633-6296>; e-mail: Volodymyr.Mietielov@khp.edu.ua

Татарінова Оксана Андріївна (Tatarinova Oksana) – кандидат технічних наук, доцент, Національний технічний університет «Харківський політехнічний інститут», завідувачка кафедри комп'ютерного моделювання процесів та систем; м. Харків, Україна; тел.: (057)-707-64-54; ORCID: <https://orcid.org/0000-0003-3090-8469>; e-mail: Oksana.Tatarinova@khp.edu.ua