

Електронний журнал «Ефективна економіка» включено до переліку наукових фахових видань України з питань економіки (Категорія «Б», Наказ Міністерства освіти і науки України № 975 від 11.07.2019). Спеціальності – 051, 071, 072, 073, 075, 076, 292. Ефективна економіка. 2025. № 1.

DOI: <http://doi.org/10.32702/2307-2105.2025.58>

УДК 658:007

Н. М. Данилевич,

*к. е. н, доцент, доцент кафедри менеджменту економічного факультету,
Львівський національний університет імені Івана Франка*

ORCID ID: <https://orcid.org/0000-0001-9906-1492>

С. М. Кошовський,

аспірант, Національний університет «Львівська політехніка»

ORCID ID: <https://orcid.org/0009-0001-8969-2925>

МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ШТУЧНИМ ІНТЕЛЕКТОМ

N. Danylevych,

*PhD in Economics, Associate Professor, Associate Professor
of the Department of Management, Ivan Franko National University*

S. Kopovskyi,

Postgraduate student, Lviv Polytechnic National University

METHODS OF TESTING SOFTWARE WITH ARTIFICIAL INTELLIGENCE

У статті розглянуто методи тестування програмного забезпечення із застосуванням штучного інтелекту (ШІ), зосереджуючи увагу на

потенціалі та перевагах інтеграції ІІІ у процеси забезпечення якості. Методи тестування, що базуються на ІІІ, представляють трансформаційний підхід до автоматизації перевірки програмного забезпечення, забезпечуючи виявлення та усунення дефектів із більшою точністю та ефективністю. У роботі досліджено техніки машинного навчання, включаючи навчання з учителем, без учителя та навчання з підкріпленням, які є ключовими для динамічної генерації тестових сценаріїв і проведення прогностичного аналізу. Нейронні мережі та генетичні алгоритми висвітлено як потужні інструменти для моделювання взаємодії користувачів, оптимізації охоплення тестів і виявлення рідкісних сценаріїв, які традиційні методи можуть пропустити.

Особливу увагу приділено ролі ІІІ в регресійному тестуванні, де автоматизовані інструменти аналізують зміни в коді та прогнозують потенційні побічні ефекти, що значно скорочує обсяг ручної роботи. Також обговорюється застосування обробки природної мови (NLP) для аналізу тестової документації та створення тестових сценаріїв, демонструючи, як ІІІ може інтерпретувати та впроваджувати складні вимоги користувачів. Крім того, ІІІ підвищує ефективність навантажувального та стресового тестування, моделюючи різноманітні операційні середовища та визначаючи вузькі місця в системі за різних умов.

У статті проаналізовано реальні приклади впровадження ІІІ у процеси тестування, демонструючи успішні кейси з таких галузей, як фінансові технології, охорона здоров'я та електронна комерція. Ці приклади ілюструють, як тестування на основі ІІІ дозволяє скоротити час виходу на ринок, підвищити надійність програмного забезпечення та знизити витрати завдяки автоматизації рутинних завдань і виявленню зон високого ризику на ранніх етапах життєвого циклу розробки. Окрім цього, у статті розглянуто такі виклики, як упередженість алгоритмів, залежність від якості даних і складність інтеграції, що забезпечує збалансований погляд на

сучасні обмеження й можливості застосування ІІІ у тестуванні програмного забезпечення.

Результати підкреслюють зростаюче значення технологій ІІІ у трансформації процесів тестування. Упроваджуючи рішення на основі ІІІ, організації можуть підвищити свою здатність керувати складними системами, адаптуватися до швидких технологічних змін і відповідати зростаючим очікуванням користувачів. У статті наведено рекомендації щодо подальших досліджень і розробок, спрямованих на максимізацію потенціалу ІІІ у створенні інноваційних, надійних та масштабованих програмних рішень.

The article examines methods of software testing using artificial intelligence (AI), focusing on the potential and benefits of integrating AI into quality assurance processes. AI-driven testing methodologies represent a transformative approach to automating software verification, enabling the identification and resolution of defects with greater accuracy and efficiency. The paper explores machine learning techniques, including supervised, unsupervised, and reinforcement learning, which are instrumental in dynamic test case generation and predictive analysis. Neural networks and genetic algorithms are highlighted as powerful tools for simulating user interactions, optimizing test coverage, and detecting edge cases that traditional methods might overlook.

Particular attention is paid to the role of AI in regression testing, where automated tools analyze code changes and predict potential side effects, significantly reducing manual labor. The application of natural language processing (NLP) in testing documentation and test case synthesis is also discussed, demonstrating how AI can interpret and implement complex user requirements. Furthermore, AI enhances the efficiency of load and stress testing by simulating diverse operational environments and identifying system bottlenecks under varying conditions.

The article reviews real-world implementations of AI in testing processes, showcasing successful case studies from industries such as fintech, healthcare, and

e-commerce. These examples illustrate how AI-based testing has reduced time-to-market, increased software reliability, and minimized costs by automating repetitive tasks and identifying high-risk areas early in the development lifecycle. Additionally, challenges such as algorithm biases, dependency on data quality, and integration complexities are addressed, providing a balanced perspective on the current limitations and opportunities of AI in software testing.

The findings underline the growing significance of AI technologies in reshaping the software testing landscape. By adopting AI-powered solutions, organizations can enhance their ability to manage complex systems, adapt to rapid technological changes, and meet evolving user expectations. The paper concludes with recommendations for further research and development to maximize the potential of AI in delivering innovative, robust, and scalable software solutions.

Ключові слова: *штучний інтелект, ефективність, стратегія, зміни, гнучкість, машинне навчання.*

Keywords: *artificial intelligence, efficiency, strategy, change, flexibility, machine learning.*

Постановка проблеми. Штучний інтелект, зокрема методи машинного навчання та нейронних мереж, достатньо широко використовуються для автоматизації процесу тестування та покращення ефективності виявлення помилок. Наприклад, інтелектуальні алгоритми можуть застосовуватись для визначення оптимальних шляхів проведення тестування програми, а також для прогнозування можливих проблеми в роботі програми, що дозволяє зменшити кількість помилок та скоротити час, необхідний для тестування програмного забезпечення. У цьому контексті дослідження методів штучного інтелекту для тестування програмного забезпечення може допомогти у пошуку можливих шляхів покращення якості тестування. Інтелектуальні алгоритми машинного

навчання та нейронні мережі можуть допомогти автоматизувати процес тестування, зменшити час, необхідний для його проведення та збільшити ефективність виявлення помилок. Також використання штучного інтелекту може підвищити точність прогнозування поведінки програмного забезпечення в різних умовах його функціонування.

Аналіз останніх досліджень та публікацій. Одним з найважливіших аспектів управління в сучасних умовах є гнучкість до змін, одним з ефективних способів для організацій, що прагнуть досягати високої результативності в умовах конкуренції є впровадження активного використання штучного інтелекту в наявних системах тестування функціоналу програмного забезпечення. Кількість проблемних аспектів значним чином скорочується шляхом впровадження ефективних сучасних методів тестування програмного забезпечення. Теоретичні основи методів впровадження штучного інтелекту в тестування функціоналу програмного забезпечення, зокрема, машинного навчання на процес управління в організації висвітлювались в роботах зарубіжних науковців, в тому числі таких як Гартнер [3], Т. Юмото [12] та інші.

У своїх публікаціях науковці висвітлили проблематику та теоретичні засади тлумачення поняття «методи тестування функціоналу програмного забезпечення за допомогою штучного інтелекту» та розробленні практичних рекомендацій щодо оцінювання рівня впливу цієї інноваційної складової менеджменту для формування чіткого інформаційного базису прийняття й провадження управлінських рішень. Водночас з цим, в недостатній мірі залишається дослідженою проблематика практичних переваг впровадження кожного з методів тестування ПЗ, а також викликів та обмежень, що пов'язані з використанням штучного інтелекту у розрізі тестування функціоналу програмного забезпечення в організації.

Метою статті є встановлення методик тестування програмного забезпечення та машинного навчання штучним інтелектом,

результативність та переваги даних способів тестування штучним інтелектом для процесу управління в організації, створення комплексного огляду можливостей, переваг та викликів впровадження ШІ у процес тестування.

Виклад основного матеріалу дослідження.

Інтеграція штучного інтелекту у тестування програмного забезпечення стає визначальним фактором у зміні підходів до забезпечення якості у світі сучасних технологій. Умови конкуренції вимагають від компаній все більш швидкого розгортання нових продуктів, але при цьому збереження найвищих стандартів якості. ШІ-тестування дозволяє суттєво спростити, прискорити та покращити якість перевірок програмного забезпечення завдяки використанню автоматизованих алгоритмів, які не тільки імітують роботу людини, а й виходять за межі можливостей традиційного ручного тестування.

Застосування ШІ у тестуванні програмного забезпечення стало не просто трендом, а життєвою необхідністю для компаній, які прагнуть залишатися конкурентоспроможними. Замість витратного та часом неефективного ручного тестування, організації дедалі частіше використовують інструменти ШІ, які здатні швидко і точно знаходити проблеми, автоматизувати рутинні завдання та надавати аналітичну інформацію для прийняття рішень.

За даними Gartner (2023), світовий ринок ШІ у тестуванні програмного забезпечення в 2023 році досяг обсягу \$2,9 мільярда, і прогнозується, що до 2028 року ця цифра збільшиться до \$7,8 мільярда зі щорічним темпом зростання (CAGR) у 21,8%. Основними драйверами зростання цього ринку є зростаюча складність програмного забезпечення, вимоги до його надійності та необхідність швидкого розгортання нових функціональних можливостей.

Ключову роль відіграє і статистика, яка свідчить про ефективність використання ШІ у тестуванні. Наприклад, 67% компаній зі списку

Fortune 500 вже впровадили ШІ у свої процеси тестування. Це демонструє не лише популярність цієї технології серед найбільших корпорацій, але й її здатність вирішувати критичні завдання, такі як мінімізація ризиків, пов'язаних із людськими помилками, та значне скорочення часу на виконання тестів.

Методи тестування програмного забезпечення на основі штучного інтелекту забезпечують численні переваги, які суттєво змінюють підхід до розробки та перевірки продуктів. Суттєвими перевагами тестування штучним інтелектом програмного забезпечення є:

1. скорочення часу на тестування на 45%. Штучний інтелект автоматизує повторювані завдання, такі як регресійне тестування, перевірка окремих модулів або автоматичне створення звітів. Це дозволяє командам розробників звільнити час для роботи над більш важливими завданнями, наприклад, для створення нових функцій або усунення критичних багів.

2. зменшення витрат на тестування на 35%. Завдяки автоматизації ШІ мінімізує потребу у великій кількості людських ресурсів. Наприклад, ручні тести, які раніше вимагали тижнів роботи декількох тестувальників, тепер можуть бути виконані за години за допомогою інструментів ШІ.

3. покращення виявлення дефектів на 78%. ШІ здатен аналізувати значно більші обсяги даних, ніж це може зробити людина, і виявляти дефекти, які могли б бути пропущені. Особливо це стосується складних систем, де проблеми можуть виникати лише за певних умов, які важко відтворити вручну.

4. скорочення ручних зусиль на 62%. Штучний інтелект бере на себе виконання рутинних завдань, таких як перевірка сумісності на різних пристроях, тестування багатомовного інтерфейсу або аналіз візуальних елементів. Це дозволяє тестувальникам зосередитися на стратегічних задачах, таких як планування тестових сценаріїв або аналіз поведінки користувачів.

Крім цього, технології ШІ дозволяють виявляти дефекти не лише під час тестування, але й на ранніх етапах розробки, що суттєво знижує загальні витрати на виправлення помилок[6].

Основні методи тестування на основі ШІ передбачають :

- генерація тестів на основі машинного навчання . Машинне навчання (ML) є важливим інструментом автоматизації процесу створення тестів. Алгоритми ML аналізують історичні дані про помилки, взаємодію користувачів і результати попередніх тестів, щоб передбачити потенційні проблеми та створити сценарії тестування, які забезпечують максимально можливе покриття.

Перевагами ML є:

- Підвищення точності виявлення дефектів на 87%.
- Розширення охоплення тестами на 42%.
- 3,2-кратне зростання ефективності у порівнянні з ручним тестуванням.
- Автоматизоване візуальне тестування. Ця технологія зосереджується на аналізі графічного інтерфейсу користувача, забезпечуючи правильне відображення елементів інтерфейсу в різних середовищах. Інструменти візуального тестування аналізують базові та тестові зображення, знаходячи найменші невідповідності.

Результатами використання є:

- економія до 4,5 годин на цикл тестування;
- підвищення ROI до 312%;
- підвищення точності тестування у 58% компаній, які інтегрували цю методологію.

Згідно з останнім опитуванням від IEEE Software Testing Standards (2024), візуальне тестування на основі ШІ допомогло організаціям скоротити цикли тестування та виявити дефекти, які могли б залишитися непоміченими під час традиційного тестування. Ці інструменти особливо

корисні для тестування кроссплатформних додатків і мобільних програм. Приклад коду для візуального порівняння за допомогою штучного інтелекту наведено на рис.1.

```
from ai_visual_testing import VisualAI

def compare_visual_elements(baseline_image, test_image):
    ai_engine = VisualAI(tolerance=0.98) # High tolerance for minor visual differences

    differences = ai_engine.compare(
        baseline_image,
        test_image,
        ignore_dynamic_elements=True
    )

    return {
        'match_percentage': differences.similarity_score,
        'detected_changes': differences.changes_list,
        'critical_differences': differences.get_critical_changes()
    }
```

Рис. 1. Код генерації тестових випадків для візуального порівняння

Тестування програмного забезпечення за допомогою штучного інтелекту включає різноманітні методики, які охоплюють різні аспекти процесу тестування[8]. Нижче наведено найпоширеніші методи, які використовуються сьогодні.

1. Генерація тестів на основі машинного навчання (Machine Learning-Based Test Generation).

Ця методика ґрунтується на аналізі історичних даних про тестування за допомогою алгоритмів машинного навчання (ML). Використовуючи ці дані, ШІ генерує нові тестові сценарії, які охоплюють потенційні "вузькі місця" в програмному забезпеченні, значно підвищуючи рівень тестового покриття.

Основні перевагами є:

- алгоритми ML прогнозують можливі сценарії використання програмного забезпечення, включаючи ті, які важко врахувати вручну;
- автоматичне створення тестових сценаріїв для різних компонентів додатка;
- постійне вдосконалення алгоритмів із часом завдяки накопиченню нових даних.

Приклад коду Python для генерації тестових випадків на основі машинного навчання зображено на рис. 2.

```
from sklearn.model_selection import train_test_split
from tensorflow import keras
import numpy as np

def generate_test_cases(historical_data, coverage_threshold=0.85):
    model = keras.Sequential([
        keras.layers.Dense(128, activation='relu'),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(64, activation='relu'),
        keras.layers.Dense(32, activation='softmax')
    ])

    x_train, x_test = train_test_split(historical_data, test_size=0.2)

    model.fit(x_train, epochs=50, validation_data=x_test)

    return model.predict(np.random.rand(5, 32)) # Random new test scenarios as an example
```

**Рис. 2. Код генерації тестових випадків на основі
машинного навчання**

Показники ефективності:

- підвищення успішності реалізації на 87%;
- розширення тестового покриття на 42%;
- ефективність у виявленні дефектів у 3,2 рази вища порівняно з традиційним тестуванням.

2. Автоматизоване візуальне тестування (Automated Visual Testing)

Цей метод використовується для перевірки візуальних компонентів програмного забезпечення, таких як графічний інтерфейс користувача (GUI). За допомогою технологій обробки зображень ШІ визначає

розбіжності між очікуваним дизайном та фактичним виглядом додатка на різних пристроях і в різних середовищах[1].

Застосування цього методу практикується для:

- перевірки коректного відображення елементів інтерфейсу на мобільних і десктопних платформах;
- виявлення візуальних помилок, які могли б залишитися непоміченими при ручному тестуванні.

Перевагами стало:

- зменшення часу тестування на 4,5 години за цикл;
- підвищення точності тестування в середньому на 78%;
- значна економія ресурсів, що дозволяє командам зосередитися на функціональних перевірках.

3. Аналіз поведінки користувачів (User Behavior Analysis).

Штучний інтелект моделює реальну поведінку користувачів у додатку, щоб визначити, як вони взаємодіють із системою. Це дозволяє знаходити дефекти, які можуть проявлятися лише в процесі фактичного використання продукту.

Прикладами використання є:

- виявлення проблем з продуктивністю під час одночасної роботи багатьох користувачів;
- оптимізація функціональності на основі поведінкових моделей.

4. Нейронні мережі для аналізу логів (Neural Networks for Log Analysis).

Нейронні мережі аналізують логи додатків, щоб виявити повторювані патерни або аномалії, які можуть свідчити про проблеми у функціонуванні.

Перевагами є :

- здатність обробляти великі обсяги даних у реальному часі;
- виявлення складних проблем, які складно ідентифікувати вручну.

5. Адаптивне тестування (Adaptive Testing)

Цей метод дозволяє ШІ змінювати тестові сценарії в реальному часі залежно від отриманих результатів, щоб зосередитися на найбільш вразливих ділянках програмного забезпечення[4].

Як результат організаціями отримано підвищення ефективності тестування за рахунок оптимізації сценаріїв та мінімізовано витрати на повторні тестові ітерації.

Використання методів тестування ШІ дозволяє суттєво підвищити якість програмного забезпечення, скоротити витрати й оптимізувати процеси розробки. Зі зростанням складності програмних систем і впровадженням нових технологій, таких як квантові обчислення та більш просунуті нейронні мережі, ці методи будуть і надалі розвиватися, забезпечуючи нові можливості для тестувальників.

Розглянемо практичний досвід застосування компаніями тестування програмного забезпечення штучним інтелектом. До прикладу, Netflix є одними з перших, хто застосував впровадження ШІ у процеси тестування програмного забезпечення, використовуючи машинне навчання для вирішення складних завдань. Їхній підхід дозволяє значно скоротити час тестування, мінімізувати витрати та покращити якість сервісу для мільйонів користувачів по всьому світу[7].

Перевагами, досягнутими Netflix є:

- 92% скорочення часу на регресійне тестування;
- 76% покращення охоплення тестами;
- 89% зниження хибних спрацювань;
- \$4,3 млн щорічної економії завдяки автоматизації.

Netflix активно використовує машинне навчання для:

1. імітації реальної поведінки користувачів;
2. оптимізації тестування складних інтерактивних функцій;
3. прогнозування проблем, пов'язаних із взаємодією користувачів з новими функціями;
4. виявлення проблем сумісності на різних пристроях;

5. впровадження безперервного А/В тестування;
6. автоматичного аналізу логів із серверів для визначення потенційних збоїв;
7. розширення сценаріїв тестування для нових мов інтерфейсу.

У майбутньому ШІ у тестуванні ще більше інтегруватиметься з такими технологіями, як квантові обчислення та нейронні мережі. Це дозволить досягати революційних показників ефективності:

- квантові алгоритми підвищать швидкість тестування на 400%;
- нейронні мережі забезпечать покращення точності на 56%.

Штучний інтелект стає центральним елементом тестування програмного забезпечення[10]. Компанії, які впроваджують ШІ, досягають високих стандартів якості, скорочуючи витрати і час. З розвитком нових технологій його значення лише зростатиме.

Висновки. Проблематика використання штучного інтелекту, зокрема, у сфері тестування функціоналу програмного забезпечення в українських організаціях перебуває на стадії активного формування, водночас відбувається якісний перехід від впровадження штучного інтелекту на етапі тестування функціоналу ПЗ та цілісного ефективного покращення управління шляхом використання сучасних гнучких та адаптивних інструментів штучного інтелекту. Ефективне лідерство вимагає уважного розгляду різноманітних чинників для досягнення цілей, уникнення можливих ризиків і загроз, що можуть виникнути.

В подальших дослідженнях увагу доцільно приділити моделювання адаптивним моделям впровадження штучного інтелекту на різних етапах управління та діяльності організації в цілому, що дозволяють ефективно управляти компаніями в кризових умовах провадження діяльності.

Розвиток моделей ефективного використання ресурсів є ключовим у забезпеченні конкурентоспроможності підприємств України на міжнародному та вітчизняному ринку. Шляхом ефективного управління цими ресурсами компанії можуть вирішити низку актуальних проблем та

підвищити свою конкурентоспроможність завдяки різноманітності сучасних інструментів в управлінні.

Також можна зазначити, що штучний інтелект є потужним інструментом, який допомагає значно зменшити час, потрібний для виконання рутинних завдань та обробки великих обсягів інформації. Проте необхідно усвідомлювати недоліки використання штучного інтелекту і дотримуватися розумного балансу між перевагами та ризиками, особливо у роботі науково-освітніх працівників.

Література

1. Artificial Intelligence to Have Dramatic Impact on Business by 2020, - According to Tata Consultancy Services Global Trend Study. URL: <https://www.tcs.com/artificialintelligence-to-have-dramatic-impact-on-business-by-2020> (дата звернення 12.10.2024)
2. Duklan, A., & Kumari, N. Intelligent Software Testing with AI and Machine Learning. Apress. 2021. 14 с.
3. Gao, J., & Xu, B. Automated Test Case Generation for Software Testing Using Machine Learning. Journal of Software Engineering Research and Development. 2021. 9(1), 1-18 с.
4. Gartner. (2023). Market Analysis Report: AI in Software Testing. Gartner. URL: <https://www.gartner.com> (дата звернення 16.10.2024)
5. IEEE. (2024). IEEE Software Testing Standards. IEEE Software Engineering Standards. URL : <https://www.ieee.org> (дата звернення 16.10.2024)
6. AI Testing Implementation Survey. (2023). Survey Results from 1,500 Companies on AI Adoption in Software Testing. URL : <https://www.surveymethods.com> (дата звернення 16.10.2024)
7. Machine Learning in Software Testing: A Comprehensive Review. (2023). Journal of Software Engineering, 45(3), 123-145.

8. Capgemini. (2023). AI Testing Case Study: Transforming Global Enterprises with AI-Driven Testing Solutions. Capgemini Research Institute. URL: <https://www.capgemini.com> (дата звернення 16.10.2024)
9. Quantum Computing for AI Testing. (2025). Future of AI in Software Testing: The Role of Quantum Computing. Journal of Emerging Technologies, 12(1), 33-50.
10. Waze, S., & Vidhate, A. AI-Powered Software Testing: A Practical Approach. Packt Publishing. 2021. 134 с.
11. Yumoto, T. Artificial Intelligence in Software Testing: A Comprehensive Guide. Apress. 2022. 120 с.
12. Zahoor, S., & Kumari, N. Applying Artificial Intelligence to Software Testing. 2021. 58-64 с.

References

1. Tata Consultancy Services Global Trend (2020), “Artificial Intelligence to Have Dramatic Impact on Business by 2020 According to Tata Consultancy Services Global Trend Study”, available at: <https://www.tcs.com/artificialintelligence-to-have-dramatic-impact-on-business-by-2020> (accessed 12.10.2024).
2. Duklan, A., & Kumari, N. (2021), Intelligent Software Testing with AI and Machine Learning, Apress, NY, USA.
3. Gao, J., & Xu, B. (2021), “Automated Test Case Generation for Software Testing Using Machine Learning”,. Journal of Software Engineering Research and Development, vol. 9(1), pp. 1-18 с.
4. Gartner. (2023), “Market Analysis Report: AI in Software Testing”, available at: <https://www.gartner.com> (accessed 16.10.2024).
5. IEEE. (2024), “IEEE Software Testing Standards. IEEE Software Engineering Standards”, available at: <https://www.ieee.org> (accessed 16.10.2024)

6. AI Testing Implementation Survey. (2023), “Survey Results from 1,500 Companies on AI Adoption in Software Testing”, available at: <https://www.surveymethods.com> (accessed 16.10.2024).
7. Journal of Software Engineering. (2023), “Machine Learning in Software Testing: A Comprehensive Review”, vol. 45(3), pp. 123-145.
8. Capgemini. (2023), “AI Testing Case Study: Transforming Global Enterprises with AI-Driven Testing Solutions”, Capgemini Research Institute, available at: <https://www.capgemini.com> (accessed 16.10.2024).
9. Quantum Computing for AI Testing. (2025), “Future of AI in Software Testing: The Role of Quantum Computing”, Journal of Emerging Technologies, vol. 12(1), pp. 33-50.
10. Waze, S., & Vidhate, A. (2021), AI-Powered Software Testing: A Practical Approach, Packt Publishing, Birmingham, UK.
11. Yumoto, T. (2022), Artificial Intelligence in Software Testing: A Comprehensive Guide, Apress, NY, USA.
12. Zahoor, S., & Kumari, N. (2021), “Applying Artificial Intelligence to Software Testing”, pp. 58-64 c.

Стаття надійшла до редакції 19.11.2024 р.