ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

УДК 004.032.26:004.415.6 DOI: 10.20998/2411-0558.2026.02.10

М. А. МІРОШНИК, д-р. техн. наук, професор, Харківський національний університет ім. В. Н. Каразіна

С. І. ШМАТКОВ д-р. техн. наук, професор, Харківський національний університет ім. В. Н. Каразіна

Ю. В. ГАЛАЙЧУК, аспірант, Харківський національний університет ім. В.Н. Каразіна

О. Д. ЗАЦ, аспірант, Харківський національний університет ім. В.Н. Каразіна

ПРОБЛЕМИ ПРИЙМАЛЬНОГО ОЦІНЮВАННЯ ПРЕДМЕТНО-ОРІЄНТОВАНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З МОДУЛЯМИ ГЛИБОКОГО НАВЧАННЯ

У статті розглянуто особливості приймального тестування програмних систем, що містять модулі нейронних мереж глибокого навчання. Проаналізовано обмеження традиційних підходів до оцінювання якості програмного забезпечення у випадку використання таких інтелектуальних моделей. Проведено систематизацію проблем приймального тестування таких систем та виконано їх класифікацію. Запропоновано концептуальний підхід до інтеграції моделі передбачення у процедуру приймального тестування для автоматизованого оцінювання якості ML-модуля використовуючи підхід "чорної скриньки".

Ключові слова: нейронні мережі, глибоке навчання, приймальне тестування оцінювання якості, підхід "чорної скриньки".

Постановка проблеми. Інтеграція методів глибокого навчання у прикладні інформаційні системи значно ускладнює структуру сучасного програмного забезпечення. Модулі з нейронними мережами дедалі частіше інтегруються у предметно-орієнтовані системи для автоматизації аналізу даних, класифікації, прогнозування та підтримки прийняття рішень.

Водночас процеси тестування програмного забезпечення, зокрема приймальне тестування користувачами, залишаються орієнтованими переважно на тестування звичайної, детермінованої бізнес-логіки або ШІ

попередніх поколінь, як, наприклад, експертних або скорінгових систем.

Особливо актуальною ця проблема є на етапі приймального тестування користувачами (User Acceptance Testing, UAT), який є завершальною стадією перевірки програмного продукту перед введенням в експлуатацію [1]. У рамках UAT оцінювання системи здійснюється представниками замовника або доменними експертами зазвичай з використанням підходу «чорної скриньки», коли оцінюється відповідність результатів роботи системи очікуванням користувачів без доступу до внутрішньої реалізації програмних компонентів.

У випадку програмних систем, що містять модулі глибокого навчання, така методологія має низку обмежень. Зокрема, складність інтерпретації рішень моделей, залежність їхньої поведінки від навчальних даних, а також можливість подальшого донавчання після введення системи в експлуатацію значно ускладнюють процес формування обґрунтованого рішення щодо прийняття програмного продукту [2, 3].

Таким чином, виникає необхідність у більш системному аналізі проблем приймального тестування програмних систем з модулями глибокого навчання та розробці підходів, які дозволяють підвищити ефективність оцінювання таких інтелектуальних компонентів у рамках процесу UAT.

Аналіз літератури. Зважаючи на стрімкий ріст використання нейронних мереж у складі програмних систем та важливість етапу приймального тестування, існують різні методи оцінки ефективності роботи таких мереж. Так, провідним підходом тестування нейронних мереж глибокого навчання на сьогодні є DeepXplore [5], що дозволяє автоматизувати процес виявлення помилок без потреби в ручному маркуванні даних. Завдяки концепції покриття нейронів (neuron coverage), цей метод забезпечує систематичний аналіз внутрішньої логіки моделі, порівнюючи результати кількох функціонально еквівалентних нейронних мереж для ідентифікації розбіжностей у їхніх прогнозах. DeepXplore ліг в основу інших методів, таких як DLFuzz [6] та DeepGauge [7]. Jianmin Guo та співавтори у своєму дослідженні пропонують замість того, щоб порівнювати нейронну мережу з іншими

системами, використовувати їх DLFuzz метод, що працює з однією моделлю, використовуючи такі принципи мутації вхідних даних та пошуку відхилень в прогнозах. Lei Ma та співавтори методу DeepGauge вводять поняття "багаторівневих критеріїв тестування" (Multi-Granularity Testing Criteria). Замість того, щоб дивитися тільки на результат (правильно/неправильно), система оцінює нейронну мережу на рівнях нейронів та шарів. Проте, такі методи є методами "білої скриньки" та зазвичай використовуються під час unit-тестування в процессі розробки нейронних мереж.

Іншим підходом є використання методів інтерпретації, таких як SHAP (SHapley Additive exPlanations) [8] та LIME (Local Interpretable Model-agnostic Explanations) [9], заснованих відповідно на теорії ігор та принципах лінійної регресії. Ці методи використовують підхід "чорної скриньки", що важливо під час UAT та дозволяють визначати внутрішню логіку роботи мережі, проте потребують значних знань у машинному навчанні та також частіше використовуються розробниками під час налаштування вагів.

Метод експертного оцінювання, при якому доменні спеціалісти аналізують результати роботи моделі та порівнюють їх із власними рішеннями [10].Такі методи в рамках UAT дозволяють залучити доменних спеціалістів для валідації рішень моделі. Їх недоліками є неможливість повноцінного тестування на всіх епохах навчання мережі, а також практична неможливість повного покриття моделі тестами через занадто велику кількість вхідних даних та вихідних результатів, що дозволяє проводити лише так зване вибіркове тестування, тобто тестування окремих випадків роботи.

Також, враховуючи відносну складність тестування нейронних мереж у порівнянні з тестуванням загальної бізнес-логіки ПЗ, під час процесу UAT часто використовуються типові тестові сценарії, що відтворюються із використанням вже розробленої супутньої бізнес-логіки. Zohreh Aghababaeyan та співавтори, наприклад, ставлять під сумнів корисність white-box метрик (таких як neuron coverage), на яких базуються DeepXplore та DeepGauge. Вони стверджують, що показники внутрішньої активності нейронів не завжди корелюють із реальною здатністю тесту виявляти помилки [11]. Замість того, щоб аналізувати

нейрони мережі, автори пропонують вимірювати різноманітність самих вхідних даних (наприклад, зображень). Якщо тестовий набір даних охоплює більш широкий спектр візуальних ознак, він краще навантажує модель і знаходить більше помилок при аналізі вихідних даних. Проте, такий підхід працює лише з моделлю на її поточному етапі навчання та є дуже залежним від наявності вичерпної кількості вхідних тестових даних, що є проблемою для UAT команди, яка не здатна використати такі методи генерації зашумлених даних, як методи змагальних атак.

Мета статті. Таким чином, метою статті є систематизація проблем приймального тестування (UAT) програмного забезпечення з модулями нейронних мереж глибокого навчання та розробка концептуальної моделі автоматизованої діагностики стану таких модулів, що дозволяє прогнозувати їх ефективність та спрощує інтерпретацію результатів для доменних експертів.

Особливості приймального тестування у життєвому циклі ПЗ. Приймальне тестування є фінальним етапом перевірки програмного продукту перед його введенням в експлуатацію. На цьому етапі система вже реалізована, інтегрована та попередньо протестована розробниками.

За типовою структурою життєвого циклу розробки програмного забезпечення етап UAT займає приблизно 5 – 10% від життєвого циклу програмного продукту [12]. Це означає, що:

– тестування проводиться в умовах обмеженого часу;

– кількість сценаріїв перевірки є обмеженою;

– основна увага приділяється ключовим бізнес-функціям.

Тестування виконується доменними експертами, які є фахівцями з предметної області, які оцінюють систему з точки зору її практичної придатності, але не володіють спеціалізованими знаннями з машинного навчання.

На відміну від традиційних програмних компонентів, моделі глибокого навчання:

– мають статистичну природу;

– залежать від навчальних даних;

– можуть демонструвати різну поведінку на нових прикладах;

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

– не мають прозорої логіки прийняття рішень.

Таким чином, можна сформувати більш детальну порівняльну характеристику алгоритмів, що використовують статичну логіку та алгоритмів, що використовують нейронні мережі глибокого навчання (табл. 1).

Таблиця 1

Порівняння традиційних алгоритмів та ML-модулів

Характеристика	**Традиційний алгоритм**	**ML-модуль**
Тип поведінки	Детермінований	Ймовірнісний
Повторюваність результату	Гарантована	Статистична
Залежність від даних	Мінімальна	Висока
Інтерпретованість	Висока	Обмежена
Стабільність на нових даних	Прогнозована	Може змінюватися

Джерело: узагальнено автором на основі [2 – 4].

Хоча сучасні стандарти тестування (серія ISO/IEC/IEEE 29119) пропонують систематичний підхід до документації та процесів, вони залишаються орієнтованими на детерміновану бізнес-логіку, що створює методологічний розрив при тестуванні інтелектуальних модулів, адже їх властивості значно ускладнюють використання стандартних процедур приймального тестування.

2.1 Аналітична класифікація проблем приймального тестування ML-модулів.

Проведений аналіз дозволяє систематизувати проблеми приймального тестування інтелектуальних модулів за чотирма основними групами.

2.1.1 Методологічні проблеми

– Невідповідність детермінованої логіки статистичній природі моделей.

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

– Орієнтація на сценарії (test cases), а не на поведінкові характеристики моделі глибокого навчання.

– Відсутність формалізованих критеріїв прийняття для ймовірнісних результатів.

2.1.2 Статистичні проблеми

– Неможливість перевірки всього простору можливих вхідних даних.

– Чутливість моделей до розподілу даних.

– Потенційна нестабільність результатів на нових прикладах.

– Складність знаходження причин помилок та їх інтерпретації.

2.1.3 Організаційні проблеми

– Обмежені часові ресурси UAT.

– Відсутність у доменних експертів з тестування кваліфікації та знань у машинному навчанні.

– Часта відсутність прямого доступу до внутрішніх метрик оцінювання моделі в UAT-середовищі.

2.1.4 Експлуатаційні проблеми

– Дрейф даних і деградація якості після релізу.

– Складність гарантувати стабільність у live-середовищі при зміні вхідних даних.

– Потенційне донавчання моделі у live-середовищі.

Окремо варто виділити останні пункти, коли частина системи, що використовує модуль нейронної мережі, після проходження процесу UAT продовжує змінюватися: донавчається на нових даних, змінює внутрішні ваги та будує нові зв’язки. У такому випадку приймальне тестування має значний недолік: воно проводиться на обмеженому тестовому наборі, який не відображає майбутні дані live-середовища. У результаті тестувальник, що використовує підхід "чорної скриньки", не може передбачити, як зміниться якість при зміні вхідних даних, що використовуються при подальшому навчанні нейронної мережі.

Порівняльна характеристика особливостей оцінювання якості роботи традиційних алгоритмів та ML-модулів наведена в табл. 2.

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

Таблиця 2

Порівняння особливостей оцінювання традиційних алгоритмів та ML-модулів

Критерій	**Традиційний алгоритм**	**ML-модуль**
Повне покриття тестами	Методологічно можливе	Неможливе
Прогнозованість результатів	Висока	Обмежена
Аналіз причин помилки	Простий	Складний
Стабільність після релізу	Передбачувана	Може змінюватися
Простота приймального рішення	Висока	Нижча
Узагальнювальна здатність	Відсутня	Ключова характеристика
Чутливість до зміни даних	Низька	Висока

Джерело: узагальнено автором на основі [3, 13, 14].

Таким чином, описані проблеми мають комплексний характер: частина з них пов’язана з властивостями нейронних мереж глибокого навчання, частина – з організацією процесів UAT та компетенціями інженерів з тестування предметно-орієнтованого ПЗ. Існуючі рішення допомагають закрити потреби тестування протягом розробки та навчання моделей нейронних мереж, проте не дають простого механізму прийняття рішення у рамках UAT без спеціальної ML-експертизи, тому на практиці виникає потреба у розробці додаткових інструментів, що можуть бути використані в умовах обмежених ресурсів доменними спеціалістами.

Концептуальна модель інтеграції діагностики ML-модуля у процедуру UAT. Одним із можливих рішень у рамках приймального тестування модулів, що використовують нейронні мережі глибокого навчання є застосування окремої моделі передбачення для

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

автоматизованої діагностики стану моделі. Такий підхід дозволяє сформувати прогноз ефективності роботи ML-модуля на змодельованих даних та відобразити внутрішні метрики ефективності нейронної мережі у вигляді, зрозумілому для доменного експерта, який проводить тестування.

Концептуально процес може бути організований таким чином (рис. 1): тестувальник обирає бізнес-сценарій та параметри контрольованого впливу на вхідні дані. Після цього система отримує змінені вхідні дані, на яких нейронна мережа формує вихідні дані. Паралельно автоматично фіксуються показники ефективності модуля, зокрема узагальнені характеристики якості, стабільності та чутливості до збурень.

Далі формується узагальнений вектор ознак, що відображає стан моделі на поточному етапі тестування та прогноз її ефективності після певного часу донавчання на даних із заданим рівнем шуму. Цей вектор передається до окремого діагностичного компонента, який виконує класифікацію стану модуля та формує інтерпретований результат для тестувальника. Таким результатом може бути, наприклад, оцінка стабільності роботи, рівня деградації якості або наявності ознак нестійкої поведінки.

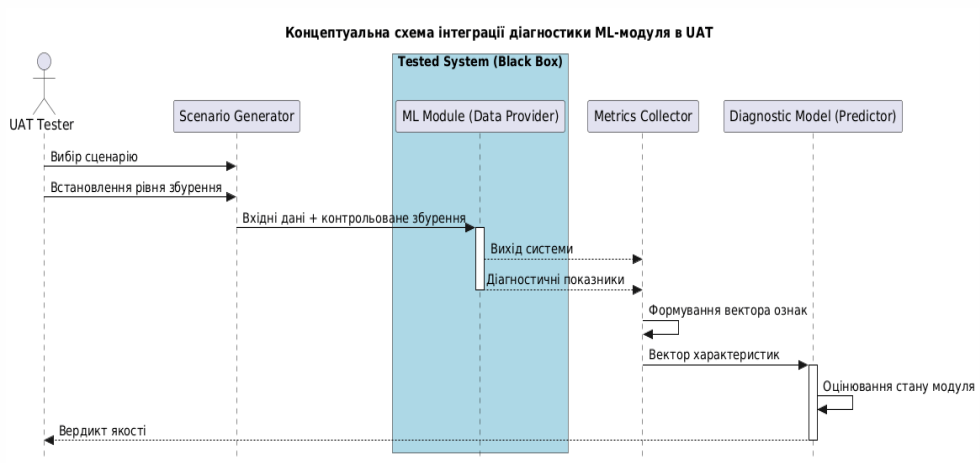

Рис. 1. Загальна схема автоматизованої перевірки стану інтелектуальної моделі

ISSN 2079-0031 (Print) ISSN 2411-0558 (Online)

3.1 Математична формалізація концепції

Для переходу від емпіричного оцінювання до автоматизованої діагностики необхідно формалізувати процес тестування інтелектуального модуля. Нехай об'єкт тестування представлений як модель глибокого навчання $f_\theta: X \to Y$ з параметрами θ, де X – простір вхідних даних, а Y – множина результатів класифікації. Враховуючи ймовірнісну природу таких систем, пропонується формування вектора діагностичних ознак Z для оцінки поточного стану модуля: $Z = \{Acc, Loss, Conf, Ent, \epsilon\}$, де:

– Acc та Loss – функціональні метрики ефективності на тестовому наборі даних;

– Conf – середній рівень впевненості моделі на виході шару Softmax;

– Ent – інформаційна ентропія розподілу ймовірностей, що характеризує невизначеність прийняття рішення;

– ϵ – параметр інтенсивності контрольованого збурення вхідних даних (рівень шуму), заданий тестувальником [2].

Центральним елементом запропонованого підходу є побудова відображення Ф, що реалізується моделлю передбачення (Prediction Model).

$\Phi : Z \to S$ де $S = \{s_1, s_2, \dots, s_n\}$ - множина станів системи, що включає "норму", "деградацію якості" або "аномальну вразливість". Це дозволяє трансформувати складні внутрішні метрики нейронної мережі у зрозумілий для доменного експерта результат оцінки якості ПЗ.

Наведена схема та модель мають концептуальний характер і використовуються для ілюстрації загального принципу використання моделі передбачення при проведенні приймального тестування програмних продуктів з ML-модулями. Реалізація такого підходу може здійснюватися різними методами залежно від вимог системи. Побудова простору ознак для навчання моделі та розробка моделі будуть розглянуті у подальших дослідженнях.

Висновки. У роботі були розглянуті особливості приймального тестування програмних систем, що містять модулі з нейронними мережами глибокого навчання. Наведений стислий огляд існуючих рішень, зокрема низку методів, що використовують підхід "білої

скриньки", методи інтерпретації, застосування тестових сценаріїв та використання існуючої бізнес-логіки ПЗ. Показано, що традиційні підходи приймального тестування користувачами не повністю враховують специфіку роботи моделей машинного навчання.

Проведено аналітичну систематизацію проблем приймального тестування інтелектуальних модулів та запропоновано їх класифікацію. Визначено методологічні, статистичні, організаційні та експлуатаційні проблеми існуючих процедур UAT. Зазначено, що складність проведення UAT для таких систем пов'язана з неможливістю повного покриття простору вхідних даних, обмеженою інтерпретованістю моделей та потенційною зміною їхньої поведінки після введення в експлуатацію.

Наведено концептуальну схему застосування моделі передбачення, спрямованої на надання можливості робити прогноз ефективності нейронної мережі після її донавчання на реальних даних та можливості спрощеної інтерпретації показників ефективності доменним експертом у процесі приймального тестування.

Отримані результати можуть бути використані при вдосконаленні процедур тестування інтелектуальних компонентів програмних систем. Проаналізовано проблеми приймального тестування предметно-орієнтованого програмного забезпечення з модулями глибокого навчання.

Подальші дослідження планується спрямувати на формування простору діагностичних ознак для навчання моделі передбачення та експериментальну перевірку запропонованого підходу на реальних програмних системах.

Список літератури:

1. Software testing — Part 2: Test processes / ISO/IEC/IEEE 29119-2:2021. Женева: International Organization for Standardization, 2021. 76 с.

2. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 800 с.

3. Hidden technical debt in machine learning systems / D. Sculley та ін. Advances in Neural Information Processing Systems. С. 2503–2511.

4. Software engineering for machine learning: A case study / S. Amershi та ін. Proceedings of the 41st International Conference on Software Engineering (ICSE), IEEE. С. 291–300. Doi: https://doi.org/10.1109/ICSE-SEIP.2019.00042

5. DeepXplore: Automated whitebox testing of deep learning systems / K. Pei та ін. Proceedings of the 26th Symposium on Operating Systems Principles. ACM. С. 1–18.

Doi: https://doi.org/10.1145/3132747.3132785

6. DLFuzz: Differential fuzzing testing of deep learning systems / J. Guo та ін. Proceedings of the ESEC/FSE Conference. С. 739–749. Doi: https://doi.org/10.1145/3236024.3264835

7. DeepGauge: Multi-granularity testing criteria for deep learning systems / L. Ma та ін. Proceedings of the 33rd IEEE/ACM International Conference on Automated Software Engineering. С. 120–131. Doi: https://doi.org/10.1145/3238147.3238202

8. A unified approach to interpreting model predictions / S. Lundberg, S. Lee // Advances in Neural Information Processing Systems. 2017. Vol. 30. С. 4765–4774. Doi: https://doi.org/10.48550/arXiv.1705.07874

9. Why should I trust you? Explaining the predictions of any classifier / M. Ribeiro, S. Singh, C. Guestrin // ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2016. С. 1135–1144. Doi: https://doi.org/10.1145/2939672.2939778

10. Goh Y. C., Qian B., Fan T. H. Evaluating human versus machine learning performance in classifying research abstracts. Scientometrics. Т. 125, вип. 2. С. 1197–1212. Doi: https://doi.org/10.1007/s11192-020-03614-2

11. Black-Box Testing of Deep Neural Networks Through Test Case Diversity / Z. Aghababaeyan, T. Ghaleb, L. Briand, S. Sen // arXiv preprint arXiv:2112.12591. 2021. URL: https://arxiv.org/abs/2112.12591 (дата звернення: 07.03.2026). Doi: https://doi.org/10.48550/arXiv.2112.12591.

12. Giang D. Best practices for user acceptance testing. HDWebsoft Blog. URL: https://www.hdwebsoft.com/blog/best-practices-for-user-acceptance-testing.html (дата звернення: 08.03.2026).

13. Machine Learning Testing: Survey, Landscapes and Horizons / J. Zhang та ін. IEEE Transactions on Software Engineering. 2020. Т. 48, вип. 1. С. 1–36. Doi: https://doi.org/10.1109/TSE.2019.2962027.

14. Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks? / L. Wang та ін. Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). С. 851–862.

References:

1. ISO (2021) ISO/IEC/IEEE 29119-2:2021 *Software testing — Part 2: Test processes*. Geneva: International Organization for Standardization.

2. Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, 800 p.

3. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F. and Dennison, D. (2015), 'Hidden technical debt in machine learning systems', paper presented at Advances in Neural Information Processing Systems (NeurIPS 28), available at: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems (accessed 9 March 2026).

4. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B. and Zimmermann, T. (2019), 'Software engineering for machine learning: A case

study', in *Proceedings of the 41st International Conference on Software Engineering (ICSE)*, pp. 291–300.

5. Pei, K., Cao, Y., Yang, J. and Jana, S. (2017), 'DeepXplore: Automated whitebox testing of deep learning systems', in *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP)*, pp. 1–18.

6. Guo, J., Jiang, Y., Zhao, Y., Chen, Q. and Sun, J. (2019), 'DLFuzz: Differential fuzzing testing of deep learning systems', in *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE),* pp. 739–743.

7. Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M. and Li, B. (2018), 'DeepGauge: Multi-granularity testing criteria for deep learning systems', in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pp. 120–131.

8. Lundberg, S. and Lee, S. (2017), 'A unified approach to interpreting model predictions', *Advances in Neural Information Processing Systems 30 (NIPS)*.

9. Ribeiro, M., Singh, S. and Guestrin, C. (2016), 'Why should I trust you? Explaining the predictions of any classifier', in *Proceedings of the ACM SIGKDD Conference*, pp. 1135–1144.

10. Goh, Y.C., Cai, X.Q., Theseira, W. et al. (2020), 'Evaluating human versus machine learning performance in classifying research abstracts', *Scientometrics Vol. 125*, pp. 1197–1212.

11. Aghababaeyan, Z., Abdellatif, M., Briand, L., Ramesh, S. and Bagherzadeh, M. (2023), 'Black-Box Testing of Deep Neural Networks Through Test Case Diversity', *IEEE Transactions on Software Engineering (TSE)*, pp. 1-26.

12. Giang, D. (2025), "Best practices for user acceptance testing", HDWebsoft Blog, available at: https://www.hdwebsoft.com/blog/best-practices-for-user-acceptance-testing.html (accessed 7 March 2026).

13. Zhang, J., Harman, M., Ma, L. and Liu, Y. (2020), "Machine Learning Testing: Survey, Landscapes and Horizons", IEEE Transactions on Software Engineering, Vol. 48, No. 1, pp. 1-36.

14. Wang L., Gulzar M., Gu Q. and Kim M. (2020), 'Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks?' in *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pp. 851–862.

Надійшла (received) 11.01.2026
Стаття прийнята до друку 03.02.2026
Опублікована 27.03.2026

Представив д-р техн. наук, проф., професор кафедри автоматизації проектування обчислювальної техники Харківський Національний Університет Радіоелектроніки Кривуля Генадій Федорович.

УДК004.032.26:004.415.6

Проблеми приймального оцінювання предметно-орієнтованого програмного забезпечення з модулями глибокого навчання / Мірошник М.А., Шматков С.І., Галайчук Ю.В. , Зац О.Д. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2026. – № 2. – С. 164 – 176.

У статті розглянуто особливості приймального тестування програмних систем, що містять модулі нейронних мереж глибокого навчання. Проаналізовано обмеження традиційних підходів до оцінювання якості програмного забезпечення у випадку використання таких інтелектуальних моделей. Проведено систематизацію проблем приймального тестування таких систем та виконано їх класифікацію Запропоновано концептуальний підхід до інтеграції моделі передбачення у процедуру приймального тестування для автоматизованого оцінювання стану ML-модуля використовуючи підхід "чорної скриньки". Іл.: 1. Табл.: 2. Бібліогр.: 14 назв.

Ключові слова: нейронні мережі, глибоке навчання, приймальне тестування оцінювання якості, підхід "чорної скриньки".

UDC 004.032.26:004.415.6

Problems of user acceptance testing of domain-oriented softwarewith deep learning modules / Miroshnyk M., Shmatkov S., Halaichuk Y., Zats O. // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkov: NTU "KhPI". – 2026. – № 2. – P. 164 – 176.

The article examines the features of user acceptance testing of software systems that include deep learning neural networks modules. The limitations of traditional approaches to software quality assessment when such models are used were analyzed. A systematization of the main problems of acceptance testing of such systems is described and their classification is proposed. A concept of a prediction model integration into the user acceptance testing process for automated assessment of the ML module state using the black-box approach is proposed. Fig.: 1. Table: 2. Bibliography: 14 titles

Keywords: neural networks, deep learning, acceptance testing, quality assessment, black box approach.