

<https://doi.org/10.26565/2519-2310-2025-2-02>

УДК 004.89:004.4

## КОНЦЕПЦІЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПРОВЕДЕННЯ ПРИЙМАЛЬНОГО ТЕСТУВАННЯ НЕЙРОННИХ МЕРЕЖ ГЛИБОКОГО НАВЧАННЯ

**Юрій Галайчук<sup>1</sup>**, аспірант кафедри комп'ютерних систем та робототехніки ННІ комп'ютерних наук та штучного інтелекту; e-mail: [yurii.halaichuk@student.karazin.ua](mailto:yurii.halaichuk@student.karazin.ua),  
ORCID: <https://orcid.org/0000-0004-1048-9425>

**Марина Мірошник<sup>1</sup>**, доктор технічних наук, професор кафедри комп'ютерних систем та робототехніки ННІ комп'ютерних наук та штучного інтелекту; e-mail: [m.miroshnyk@karazin.ua](mailto:m.miroshnyk@karazin.ua),  
ORCID: <https://orcid.org/0000-0002-2231-2529>

**Ельвіра Кулак<sup>2</sup>**, доктор філософії, доцент кафедри автоматизації проектування; e-mail: [elvira.kulak@nure.ua](mailto:elvira.kulak@nure.ua), ORCID: <https://orcid.org/0000-0002-8441-5187>

<sup>1</sup>*Харківський національний університет імені В.Н. Каразіна,  
майдан Свободи, 4, Харків, 61022, Україна*

<sup>2</sup>*Харківський національний університет радіоелектроніки, пр. Науки, 14, Харків, 61166, Україна*

Рукопис надійшов 2 вересня 2025 р. Отримано після рецензування 1 жовтня 2025 р.

Прийнято 2 листопада 2025 р. Опубліковано 30 грудня 2025 р.

**Анотація:** У сучасному світі все більше критичних інфраструктур та комерційних систем покладаються у своїй роботі на результати обчислень алгоритмів штучного інтелекту, зокрема нейронних мереж. Паралельно з цим набуває великого значення проведення процесу оцінювання якості роботи таких алгоритмів та належного виконання всіх етапів їх тестування для усунення можливих недоліків та забезпечення здатності отримувати очікувані результати. Дана стаття присвячена проблемі вдосконалення процесу приймального тестування користувачами (User Acceptance Testing, UAT) для предметно-орієнтованого програмного забезпечення, що використовує нейронні мережі глибокого навчання. У роботі розглядаються виклики, пов'язані з обмеженими ресурсами, недостатньою кваліфікацією команд UAT у сфері машинного навчання та складністю тестування систем, які продовжують навчатися після початкової розробки та проведений загальний огляд наявних шляхів вирішення зазначених проблем з вказанням їх переваг та недоліків. Запропоновано концепт інтелектуальної інформаційної системи, що базується на моделі прогнозування для оцінки показників якості нейронних мереж, зокрема точності та функції втрат та дозволяє провести процес оцінювання якості таких мереж використовуючи набір навчальних і валідаційних даних. Описано експериментальну методологію, яка включає алгоритм розробки моделі прогнозування для аналізу трендів якості нейронної мережі та розробку інтелектуальної інформаційної системи для прискорення та спрощення процесу UAT. Представлено схему розгортання компонентів системи, що охоплює взаємодію між клієнтськими додатками, веб-сервером, сервером виконання та базою даних із застосуванням сучасних мережевих протоколів та технологій. Результати дослідження спрямовані на підвищення ефективності проведення процесу UAT шляхом його автоматизації та застосування моделі передбачення для отримання динамічних показників якості алгоритмів нейронних мереж глибокого навчання.

**Ключові слова:** *глибоке навчання, інтелектуальна інформаційна система, модель передбачення, оцінювання якості, приймальне тестування користувачами*

**Як цитувати:** Галайчук Ю., Мірошник М., Кулак Е. Концепція інтелектуальної інформаційної системи для проведення приймального тестування нейронних мереж глибокого навчання. *Комп'ютерні науки та кібербезпека*. 2025; № 2(28): С. 21–31. <https://doi.org/10.26565/2519-2310-2025-2-02>

**In cites:** Halaichuk Y., Miroshnyk M., Kulak E. (2025). The concept of an intelligent information system for conducting acceptance testing of deep learning neural networks. *Computer Science and Cybersecurity*. 2(28): 21–31. <https://doi.org/10.26565/2519-2310-2025-2-02> (in Ukrainian)

## 1. Вступ

Завдяки стрімкому розвитку галузі машинного навчання та зростанню обчислювальних потужностей у сучасному світі ми спостерігаємо стрімке проникнення технологій штучного інтелекту, зокрема нейронних мереж глибокого навчання (Deep Learning), у різноманітні сфери людської діяльності [1]. Від автоматизованого аналізу медичних зображень для ранньої діагностики захворювань та розпізнавання мови в голосових помічниках до виявлення шахрайства та обробки фінансових транзакцій у фінансовому секторі, алгоритми глибокого навчання здатні обробляти складні дані та приймати інтелектуальні рішення. Часто таке програмне забезпечення є предметно-орієнтованим та розробляється по замовленню користувачів, проходячи життєвий цикл розробки від аналізу вимог до приймального тестування (User Acceptance Testing, UAT).

Ключовою відмінністю нейронних мереж глибокого навчання є те, що такі мережі продовжують навчатись після закінчення процесу їх розробки та початкового навчання, що значно ускладнює їх тестування. При цьому у предметно-орієнтованому програмному забезпеченні алгоритм нейронної мережі глибокого навчання може бути лише невеликою частиною відносно загального його обсягу, а UAT проводиться командами, учасники яких є спеціалістами у предметній галузі, але не є спеціалістами у галузі машинного навчання [11].

Через сукупність цих факторів, а також обмеженні ресурси на проведення UAT у порівнянні із загальним процесом розробки та тестування програмного забезпечення, виникає загальна потреба у розробці та впровадженні інноваційних підходів до UAT, які б могли забезпечити більш глибоку та автоматизовану перевірку функціональності складного програмного забезпечення, що використовує алгоритми нейронних мереж глибокого навчання.

## 2. Постановка проблеми

Приймальне тестування користувачами є фінальним етапом тестування програмного забезпечення перед його введенням в експлуатацію, спрямованим на підтвердження готовності системи до використання реальними користувачами в їхньому наближеному до реального середовищі [2]. Зазвичай процес UAT проводиться окремою незалежною командою з боку замовника (користувача) та включає в себе виконання інших типів тестування програмного забезпечення з метою верифікації на відповідність заявленим вимогам. У контексті програмного забезпечення, що використовує алгоритми нейронних мереж глибокого навчання, проведення ефективного UAT особливо важливо, оскільки неправильна робота або неочікувана поведінка таких систем може призвести не лише до фінансових втрат, але й до серйозних наслідків у сферах, де приймаються критично важливі рішення на основі їх висновків.

Незважаючи на те, що не існує точно зазначених правил, скільки часу відносно загального терміну життєвого циклу програмного забезпечення (SDLC) мають займати різні типи тестування, звернувшись до теорії тестування можна сформулювати наступний розподіл [3, 4, 5]:

Таблиця 1 – Порівняння приблизного часу на виконання типів тестування відносно SDLC

*Table 1 – Comparison of estimated execution time of test types relative to SDLC*

Тип тестування	Час на тестування відносно SDLC	Чи входить до UAT
Модульне тестування	5-10%	Ні
Інтеграційне тестування	10-15%	Так
Функціональне тестування	20-30%	Так
<b>Приймальне тестування користувача</b>	<b>10-15%</b>	-
Регресійне тестування	5-10%	Ні
Тестування продуктивності	10-15%	Так
Тестування безпеки	10-15%	Так
Тестування зручності та інтерфейсу користувача	5-10%	Так

Значення можуть значно відрізнятись залежно від специфіки та архітектури проекту, проте з Таблиці 1 можна побачити, що на UAT виконується за малий термін відносно загального життєвого циклу програмного забезпечення та включає в себе великий обсяг робіт. Також, додатково процес ускладнюють:

а) відсутність у команд тестування, що проводять UAT з боку замовника, кваліфікацій у машинному навчанні - знайти людей, які одночасно є спеціалістами у предметній галузі та машинному навчанні важко та не завжди фінансово доцільно;

б) відсутність доступу у UAT команди до процесу розробки та навчання нейронної мережі.

Зазвичай, для проведення UAT у таких випадках застосовуються наступні способи [6]:

1. Використання існуючої бізнес-логіки для передачі вхідних даних що використовуються для подальшого навчання алгоритму та використання наявного набору валідаційних даних для верифікації вихідних даних [7]. Цей підхід має певні недоліки, такі як високі витрати часу на введення даних та необхідність мати великий об'єм навчальних даних, що не використовувались при початковому навчанні алгоритму. Частково проблема з введенням даних вирішується застосуванням засобів автоматизації, імпортом даних у базу даних або за допомогою імпорту через файловий формат, проте, такі можливості не завжди наявні у системі без додаткових доопрацювань, а доопрацювання можуть бути не обґрунтовані економічно.

2. Застосування методів пояснення роботи нейронних мереж, таких як SHAP (SHapley Additive exPlanations) [8] та LIME (Local Interpretable Model-agnostic Explanations) [9], заснованих відповідно на теорії ігор та лінійній регресії, для інтерпретації зв'язків у моделі нейронної мережі. Ці методи є класичними методами тестування чорної скрині (black box

testing) нейронних мереж, проте їх недоліками в контексті проведення UAT є відносна математична складність та нездатність моделювати саме процес донавчання моделі.

3. Використання методів імітаційного моделювання (Synthetic Data Generation, Environment Simulation, Adversarial Attack Simulation, Monte Carlo Simulation). Такі методи дозволяють провести тестування із покриттям багатьох нестандартних випадків, проте, потребують створення нових наборів даних для тестування та є більш доцільними до використання протягом розробки та модульного тестування нейронних мереж.

Таким чином, існує потреба у впровадженні методології незалежного тестування нейронних мереж глибокого навчання з урахуванням часових обмежень та кваліфікації команд, що проводять тестування.

### **3. Концепція інтелектуальної інформаційної системи для проведення UAT нейронних мереж глибокого навчання**

У вирішенні зазначених проблем могла б допомогти розробка інтелектуальної інформаційної системи з алгоритмом моделі передбачення, що надасть змогу передбачити тренд показників якості нейронних мереж, зокрема мереж глибокого навчання та провести незалежне тестування використовуючи наявний набір навчальних та валідаційних даних.

Короткий опис дослідження, спрямованого на розробку моделі передбачення для тестування мереж глибокого навчання, може виглядати наступним чином:

1. Програмна реалізація нейронної мережі для збору даних з відповідності показників якості ступеням навчання (Мережа А);
2. Розподіл навчальних даних Мережі А на три частини - навчальну, валідаційну, дані для донавчання.
3. Використання навчального набору даних для навчання Мережі А.
4. Використання валідаційного набору даних для зняття показників точності, значення функції втрат на  $n$  епохах (ітераціях) навчання Мережі А при різних конфігураціях навчальних даних.
5. Застосування програмних та математичних методів для розробки та реалізації моделі прогнозування (умовно Мережа В).
6. Навчання Мережі В на отриманих при тестуванні мережі а показниках.
7. Застосування Мережі В для прогнозування показників точності, значення функції втрат при подальшому навчанні Мережі А.
8. Тестування моделі із застосуванням набору даних для донавчання, що емулює подальше навчання Мережі А та порівняння отриманих реальних даних з результатами прогнозування Мережі В.

#### **3.1. Вибір моделі для проведення збору показників якості роботи нейронної мережі.**

Для збору показників якості було вирішено використовувати модель класифікації зображень, оскільки класифікація зображень є відносно простою задачею та при цьому широко застосовується у предметно-орієнтованих системах. Оскільки кінцевою метою роботи є розробка системи для широкої аудиторії користувачів, були розглянуті найбільш популярні типи нейронних мереж що використовуються при навчанні та тестуванні для задач класифікації зображень, такі як залишкові (Residual network, ResNet), згорткові (Convolutional networks, зокрема, MobileNet та VGG), зорові трансформери (Vision Transformers, ViT), нейронні мережі прямого поширення (Feedforward neural network, FNN).

Також при виборі типу нейронної мережі крім ступеня її розповсюдженості були враховані наступні критерії:

1. складність та місткість архітектури (Capacity): модель не повинна бути занадто простою або занадто складною задля запобігання швидкого перенавчання або недонавчання;
2. динаміка збіжності (Convergence Dynamics): для отримання достатньої вибірки даних для навчання моделі прогнозування необхідна тривала фаза навчання Мережі А;
3. чутливість до гіперпараметрів: при формуванні набору даних з показників навчання Мережі А необхідні невдалі епохи навчання, яких можна добитись зміною налаштувань мережі;
4. ефективність обчислення: одна епоха навчання не має займати вкрай довгий час.

Стислий аналіз розглянутих типів нейронних мереж наведено у таблиці 2 [10][15][16]. Згідно з проведеним аналізом було прийняте рішення використовувати для збору показників ефективності згортокву модель глибинного навчання MobileNet.

Таблиця 2 – Порівняння архітектур-класифікаторів для генерації метрик  
*Table 2 – Comparison of classifier architectures for metric generation*

Критерій	ResNet	MobileNet	Vision Transformer	VGG	FNN
Складність та місткість	Середня / Висока	Низька	Дуже висока	Висока	Залежить від кількості шарів нейронів
Динаміка збіжності	Стабільна (плавна крива втрат)	Швидка, але нестабільна (є коливання метрик)	Дуже повільна	Повільна	Дуже швидка лише на початку
Чутливість до гіперпараметрів	Середня	Низька	Дуже висока	Середня / Висока	Дуже висока
Ефективність обчислення	Висока	Найвища	Низька	Низька	Низька для зображень

### 3.2. Вибір моделі прогнозування.

Вибір моделі для прогнозування результатів навчання базується на аналізі метрик як часових рядів. Оскільки значення точності та втрат на кожній епосі прямо залежать від попередніх кроків, головним критерієм вибору архітектури є її здатність працювати з послідовними даними та «запам'ятовувати» тренди.

Для цієї задачі було розглянуто архітектури, що здатні ефективно виявляти закономірності у зміні градієнта. Рекурентні мережі (наприклад, LSTM) є пріоритетними, оскільки вони спроможні відрізнити випадкові коливання метрик від реального плато. Водночас модель має бути достатньо компактною, щоб не перенавчатися на обмеженій кількості кривих навчання, зібраних під час тестування мережі класифікатора. Таким чином, пріоритетом є баланс між глибиною пам'яті моделі та швидкістю її роботи для оперативного прийняття рішень у процесі UAT.

При виборі моделі було враховано наступні критерії:

1. обробка послідовностей (Sequential Dependency): здатність мережі пам'ятати, що відбувалося на 1-й епосі, коли вона аналізує 10-ту;
2. робота з вхідними даними змінної довжини: можливість дати прогноз як після 5-ї епохи, так і після 50-ї;
3. ризик перенавчання на малих вибірках: оскільки набір даних «кривих навчання» відносно невеликий, модель не повинна бути надто складною;
4. швидкість виведення (Inference Speed): прогноз має бути миттєвим, оскільки система передбачає отримання даних у реальному часі;
5. здатність до екстраполяції: наскільки точно мережа може передбачити значення на 100-й епосі, бачачи лише перші 10.

Стислий аналіз розглянутих типів нейронних мереж наведено у таблиці 3 [17] [18] [19] [20]:

Таблиця 3 – Стислий аналіз розглянутих типів нейронних мереж  
Table 3 – Brief analysis of the considered types of neural networks

Критерій	LSTM / GRU (Рекурентні)	1D-CNN (Згорткові 1D)	Трансформери	Multilayered perceptron (MLP)
Обробка послідовностей	Має «внутрішню пам'ять» про минулі епохи	Обмежена розміром вікна згортки	Теоретично необмежена	Відсутня
Тип даних	Спадкові послідовності	Локальні патерни в часі	Глобальні залежності	Статичні вектори
Стійкість до малих наборів даних	Середня	Висока	Низька	Дуже висока
Швидкість	Низька	Висока	Середня / Низька	Найвища
Здатність до екстраполяції	Висока	Висока	Висока	Середня

Після проведеного аналізу найкращим вибором визнана Довга короткочасна пам'ять (Long Short-Term Memory, LSTM) або її спрощена версія Вентильний рекурентний вузол (Gated recurrent units, GRU). Вибір цих моделей зумовлений їх здатністю ефективно опрацьовувати часові послідовності метрик, зберігаючи інформацію про динаміку навчання на довгих інтервалах епох.

Основні переваги застосування LSTM / GRU для поставленої задачі:

1. Точність прогнозування: завдяки механізму «пам'яті», модель здатна екстраполювати значення Accuracy та Loss, ідентифікуючи передчасний вихід на плато або деградацію градієнта.
2. Універсальність: модель демонструє стабільні результати при роботі з вхідними даними від різних класифікаторів (ResNet, MobileNet), що робить її надійним інструментом для етапу UAT.

### 3.3. Проектування інтелектуальної інформаційної системи.

Для подальшої автоматизації може бути впроваджена інформаційна система, яка дозволяє передавати на вхід дані про залежність показників якості роботи нейронної мережі від конфігурації навчальних даних (наприклад, у файловому вигляді), задавати очікувану конфігурацію реальних даних, які буде обробляти алгоритм нейронної мережі після його впровадження у роботу та у реальному часі отримувати прогноз очікуваних показників якості роботи мережі, що тестується, у процесі її подальшого навчання. Можлива архітектура такої системи представлена на Схемі 1.

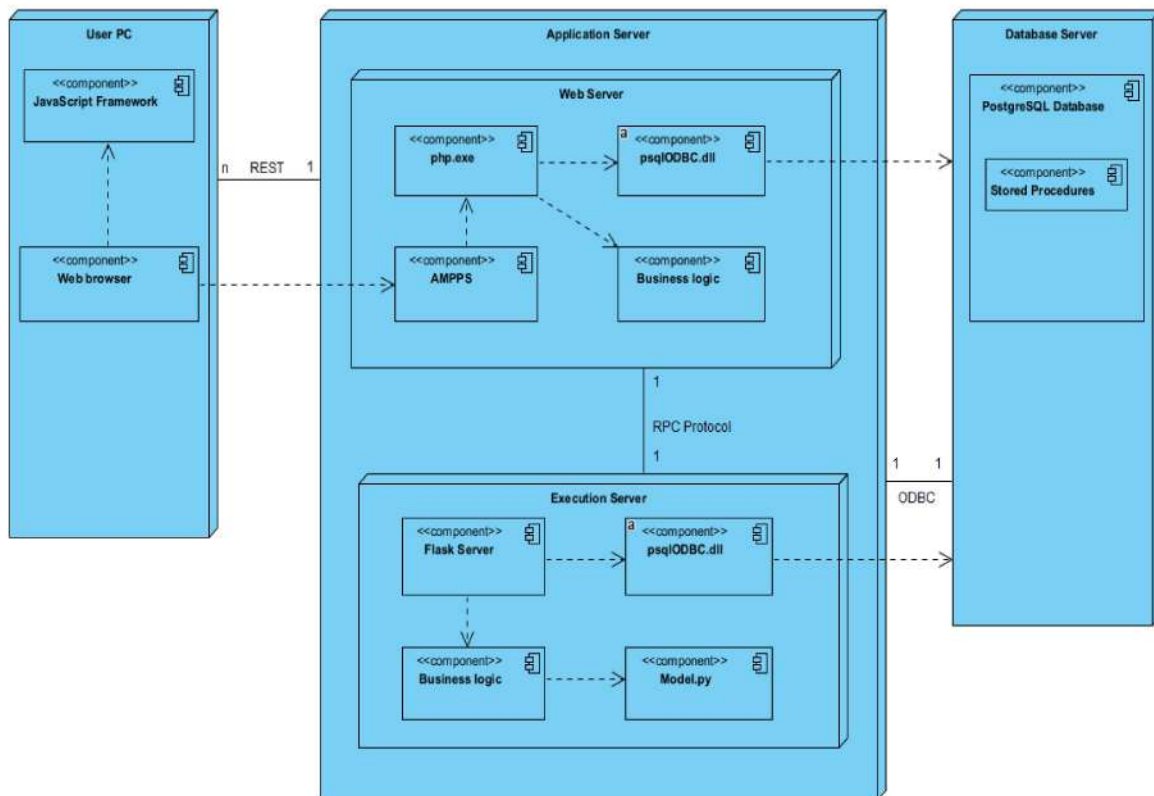


Схема 1 – Діаграма розгортання компонентів [12]

Figure 1 - Component Deployment Diagram [12]

Де:

Вузол User PC:

- 1) Web Browser - програма веб-браузер, що надає користувачеві доступ до системи;
- 2) JavaScript Framework - це каркас коду JS, що є необхідним для програмної реалізації певної бізнес-логіки на стороні клієнта.

Вузол Web Server [13]:

- 1) AMPPS - це каркас для виконання php коду веб-сторінок;
- 2) php.exe - php інтерпретатор;
- 3) psqODBC.dll - PostgreSQL ODBC драйвер;
- 4) Business logic - програмний код на мові php, що виконує певну частину бізнес логіки (парсинг отриманих від користувача файлів з даними для подальшої обробки)

Вузол Execution Server:

- 1) Flask Server - це каркас для виконання back end python коду;

- 2) Business logic - програмний код на мові python, що виконує певну частину бізнес логіки (обробку отриманих від користувача даних);
- 3) Model.py - це модель передбачення на основі нейронної мережі, що використовується для отримання вихідних даних;
- 4) psycopg2.dll - PostgreSQL ODBC драйвер;  
Вузол Database Server[14]:
- 1) PostgreSQL - це БД системи, з якою Web-сервер взаємодіє за допомогою протоколу драйверів ODBC (Open Database Connectivity), а компонент StoredProcedure - це збережені процедури, що містяться у БД системи.

Обмін статичними (збереженими) даними при цьому відбувається через БД, до якої мають доступ компоненти як Web Server, так і Execution Server. Обмін даними між серверами у реальному часі (наприклад, даними про початок / закінчення обробки отриманих даних) відбувається з використанням RPC Protocol (A Remote Procedure Call). Для комунікації вузлів User PC та Web Server використовується REST архітектура мережеских протоколів, оскільки майже не має обмежень щодо формату передачі наборів даних.

Запропонована система має певні переваги та недоліки. Із переваг можна відмітити:

1. універсальність: модель може працювати із різними нейронними мережами спільної архітектури, що усуває необхідність у запровадженні інструментів з автоматизації протягом розробки кожного окремого проекту;

2. швидкість застосування: завдяки автоматизації процесу система дозволяє значно заощадити час на тестування в умовах UAT;

3. низькі вимоги до кваліфікації користувачів у галузі машинного навчання, що робить використання системи доступним для будь-якої команди спеціалістів у предметній галузі;

4. відсутність необхідності у великому обсязі реальних даних для тестування;

5. рішення може бути інтегровано у процес тестування доповнюючи використання інших наявних методів, наприклад, SHAP і LIME для подальшої інтерпретації незадовільних по показникам якості вихідних даних.

Недоліки:

1. модель передбачення не здатна надавати точні значення очікуваних показників якості, а лише значення їх трендів, що можуть бути використані для прийняття рішення про проходження процесу UAT або повернення на доопрацювання;

2. система потребує передачі даних про навчання нейронної мережі, що тестується, від команди розробки до UAT команди, що часто є звичайним процесом передачі тестової документації;

3. модель передбачення вимагає формування очікуваної (estimated) конфігурації даних, що будуть у подальшому використані для здійснення навчання після процесу тестування; ця вимога є типічною вимогою при проведенні процесу тестування та збір очікуваних даних про майбутню роботу програмного забезпечення є невід'ємною частиною таких типів тестування, як тестування продуктивності та тестування сумісності.

#### 4. Висновки

У результаті проведеного аналізу встановлено, що сучасні підходи до приймального тестування користувачами (UAT) програмного забезпечення, яке використовує нейронні мережі глибокого навчання, стикаються з низкою викликів, зокрема через обмежені часові ресурси, недостатню кваліфікацію команд у сфері машинного навчання та складність тестування систем, що продовжують навчатися після початкового етапу розробки. Ці фактори ускладнюють забезпечення надійності та відповідності таких систем заявленим вимогам, особливо в

предметно-орієнтованому програмному забезпеченні, де нейронні мережі є лише частиною загальної архітектури.

Запропонована концепція інтелектуальної інформаційної системи для автоматизації UAT нейронних мереж глибокого навчання базується на використанні моделі передбачення, яка дозволяє прогнозувати показники якості, такі як точність і значення функції втрат, на основі даних, отриманих із тестування базової нейронної мережі. Такий підхід дає змогу моделювати процес донавчання мережі та проводити незалежне тестування з використанням наявних навчальних і валідаційних даних, що зменшує залежність від великих обсягів нових даних і знижує витрати часу на введення даних.

Проведений стислий аналіз існуючих архітектур нейронних мереж, які підходять для задач збору показників навчання для прогнозування з формуванням ключових критеріїв вибору та обґрунтований вибір моделі для кожної із задач.

Розроблена схема розгортання компонентів системи, яка включає вузли User PC, Web Server, Execution Server і Database Server, забезпечує ефективний обмін даними через REST-архітектуру, RPC-протокол і ODBC-драйвери. Це дозволяє реалізувати гнучку та масштабовану систему, здатну обробляти дані в реальному часі та зберігати статичні дані для подальшої обробки.

Запропонована методологія має потенціал для спрощення UAT шляхом автоматизації ключових етапів тестування, зменшення залежності від висококваліфікованих спеціалістів у галузі машинного навчання та підвищення точності оцінки якості нейронних мереж. Подальші дослідження можуть бути спрямовані на вдосконалення моделей прогнозування, розширення їхньої застосовності до складніших архітектур нейронних мереж, а також інтеграцію з іншими методами пояснення роботи нейронних мереж, такими як SHAP і LIME, для забезпечення більшої інтерпретованості результатів.

### Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

### References

1. Goodfellow I., Bengio Y., Courville A. (2016) *Deep Learning*. Cambridge: MIT Press, 800 c. Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville [https://books.google.com.ua/books/about/Deep\\_Learning.html?id=Np9SDQAAQBAJ&redir\\_esc=y](https://books.google.com.ua/books/about/Deep_Learning.html?id=Np9SDQAAQBAJ&redir_esc=y)
2. Sommerville I. (2015) *Software Engineering*. 10th ed. Boston: Pearson, 816 c. Software Engineering, Global Edition - Ian Sommerville [https://books.google.com.ua/books/about/Software\\_Engineering\\_Global\\_Edition.html?id=W\\_LjCwAAQBAJ&redir\\_esc=y](https://books.google.com.ua/books/about/Software_Engineering_Global_Edition.html?id=W_LjCwAAQBAJ&redir_esc=y)
3. Myers G.J., Sandler C., Badgett T. (2011) *The Art of Software Testing*, 3rd ed, New York: Wiley, 256 c. The Art of Software Testing - Glenford J. Myers, Corey Sandler, Tom Badgett [https://books.google.com.ua/books/about/The\\_Art\\_of\\_Software\\_Testing.html?id=GjyEFPkMCwC&redir\\_esc=y](https://books.google.com.ua/books/about/The_Art_of_Software_Testing.html?id=GjyEFPkMCwC&redir_esc=y)
4. ISO/IEC/IEEE 29119-1:2013. (2013) *Software and Systems Engineering. Software Testing Part 1: Concepts and Definitions*. Geneva: International Organization for Standardization. ISO/IEC/IEEE 29119-1:2013 | IEC <https://webstore.iec.ch/en/publication/11972>
5. International Software Testing Qualifications Board. (2023). *ISTQB Certified Tester Foundation Level Syllabus (Version 4.0.1)*. URL: <https://www.istqb.org/certifications/certified-tester-foundation-level>
6. Russell S., Norvig P. (2020) *Artificial Intelligence: A Modern Approach*, 4th ed, Boston: Pearson, 1152 c. Artificial Intelligence: A Modern Approach, 4th US ed. <https://aima.cs.berkeley.edu/>
7. Tian, Y., Pei, K., Jana, S., & Ray, B. (2018). *DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars*. Proceedings of the 40th International Conference on Software Engineering.

- DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars <https://arxiv.org/pdf/1708.08559v1>
8. Lundberg S.M., Lee S.I. (2017) *A Unified Approach to Interpreting Model Predictions* // *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30, с. 4765-4774. DOI: <https://doi.org/10.48550/arXiv.1705.07874>
  9. Ribeiro M.T., Singh S., Guestrin C. (2016) *"Why Should I Trust You?": Explaining the Predictions of Any Classifier* // *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, с. 1135-1144. DOI: <https://doi.org/10.1145/2939672.2939778>
  10. He K. (2016) *Deep Residual Learning for Image Recognition* / K. He, X. Zhang, S. Ren, J. Sun // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. - P. 770-778. <https://doi.org/10.48550/arXiv.1512.03385>
  11. Zhang J. M., Harman M., Ma L., Liu Y. (2020) *Machine Learning Testing: Survey, Landscapes and Horizons* // *IEEE Transactions on Software Engineering*. Vol. 48 No. 1, с. 1-36. DOI: <https://doi.org/10.1109/TSE.2019.2962027>
  12. Axel K. (1998) *Using UML for Business Object Based Systems Modeling* / Korthaus Axel // *ResearchGate*. [https://doi.org/10.1007/978-3-642-48673-9\\_15](https://doi.org/10.1007/978-3-642-48673-9_15)
  13. Nawroze I. (2023) *A Comparative Analysis of PHP and Python Programming Languages for Optimal Software Development* / I. Nawroze, C. Rubel // *International Journal of Information Technology* 8(1): 1-13, URL: <http://dx.doi.org/10.6084/m9.figshare.24885846.v1>
  14. Faisal Qureshi, Haida Rasheed. (2022) *Comparative Analysis of Modern Database Technologies for Scalable Data Storage in AI-Driven Ecommerce Applications* / Faisal Qureshi, Haida Rasheed // *ResearchGate*, URL: <http://dx.doi.org/10.13140/RG.2.2.14668.83848>
  15. Howard A.G. (2017) *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* / A. G. Howard, M. Zhu, B. Chen [et al.]. - URL: <https://arxiv.org/abs/1704.04861>
  16. Tan M. (2019) *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* / M. Tan, Q. V. Le // *International Conference on Machine Learning (ICML)*. <https://doi.org/10.48550/arXiv.1905.11946>
  17. Hochreiter S. (1997) *Long Short-Term Memory* / S. Hochreiter, J. Schmidhuber // *Neural Computation*. - Vol. 9, Iss. 8. - P. 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
  18. Vaswani A. (2017) *Attention is All You Need* / A. Vaswani, N. Shazeer, N. Parmar [et al.] // *Advances in Neural Information Processing Systems (NIPS)*. - P. 5998-6008. <https://doi.org/10.48550/arXiv.1706.03762>
  19. Ismail Fawaz H. (2019) *Deep learning for time series classification: a review* / H. Ismail Fawaz, G. Forestier, J. Weber [et al.] // *Data Mining and Knowledge Discovery*. - Vol. 33. - P. 917-963. <https://link.springer.com/article/10.1007/s10618-019-00619-1>
  20. Domhan T. (2015) *Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolating Learning Curves* / T. Domhan, J. T. Springenberg, F. Hutter // *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.- P. 3460-3468. <https://dl.acm.org/doi/10.5555/2832581.2832731>

## THE CONCEPT OF AN INTELLIGENT INFORMATION SYSTEM FOR CONDUCTING ACCEPTANCE TESTING OF DEEP LEARNING NEURAL NETWORKS

**Yurii Halaichuk**<sup>1</sup>, PhD student of Computer Systems and Robotics Department Institute of Computer Science and Artificial Intelligence; e-mail: [yurii.halaichuk@student.karazin.ua](mailto:yurii.halaichuk@student.karazin.ua); ORCID:

<https://orcid.org/0009-0004-1048-9425>

**Maryna Miroshnyk**<sup>1</sup>, Doctor of technical sciences, Professor of Computer Systems and Robotics Department Institute of Computer Science and Artificial Intelligence; e-mail:

[m.miroshnyk@karazin.ua](mailto:m.miroshnyk@karazin.ua); ORCID: <https://orcid.org/0000-0002-2231-2529>

**Elvira Kulak**<sup>2</sup>, PhD, Associate Professor, Associate Professor of Design Automation Department; e-mail: [elvira.kulak@nure.ua](mailto:elvira.kulak@nure.ua); ORCID: <https://orcid.org/0000-0002-8441-5187>

<sup>1</sup> V. N. Karazin Kharkiv National University, Ukraine

<sup>2</sup>Kharkiv National University of Radio Electronics, Nauka Avenue, 14, Kharkiv, Ukraine, 61166

Manuscript was received September 2, 2025; Received after review October 1, 2025;

Accepted November 2, 2025; Published December 30, 2025

**Abstract.** In the modern world, an increasing number of critical infrastructures and commercial systems rely on the results of computations by artificial intelligence algorithms, particularly neural networks. In parallel, the process of evaluating the quality of these algorithms and ensuring proper execution of all stages of their testing has become highly significant to eliminate potential flaws and ensure their ability to deliver expected results. The article addresses the issue of improving the User Acceptance Testing (UAT) process for domain-specific software utilizing deep learning neural networks. It examines challenges related to limited resources, insufficient UAT team expertise in machine learning, and the complexity of testing systems that continue learning post-initial development. A general overview of existing solutions is provided, highlighting their advantages and drawbacks. A concept of an intelligent information system based on a predictive model for evaluating neural network quality metrics, specifically accuracy and loss function is proposed, enabling the quality assessment process of such networks using a set of training and validation data. An experimental methodology is described, including the algorithm of development of a predictive model for analyzing network quality trends and the creation of an intelligent information system to streamline and accelerate the UAT process. The system's component deployment architecture is presented, covering interactions between client applications, a web server, an execution server, and a database, leveraging modern network protocols and technologies. The research results aim to enhance UAT efficiency through automation and the application of a predictive model to obtain dynamic quality metrics for deep learning neural network algorithms.

**Keywords:** *deep learning, intelligent information system, quality assurance, prediction model, user acceptance testing*

**Conflicts of Interest:** the authors declare no conflict of interest.