

ПРОГНОЗУВАННЯ ПОКАЗНИКІВ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ ІЗ ЗАСТОСУВАННЯМ МОДИФІКАЦІЙ ІНТЕРПРЕТАЦІЙНОГО МЕТОДУ *DeepLIFT*

Предметом дослідження є покращення інтерпретаційного методу *DeepLIFT* (*Deep Learning Important FeaTures*) у контексті забезпечення оцінки якості програмних систем (ПС). Вивчаються алгоритмічні та математичні аспекти базового методу *DeepLIFT*, а також його вдосконалення для аналізу та покращення показників якості ПС. **Мета роботи** – розроблення та обґрунтування п'яти способів оптимізаційних модифікацій методу *DeepLIFT* для покращення точності та ефективності прогнозування показників якості ПС. У статті необхідно виконати такі **завдання**: детально проаналізувати проблеми, пов'язані із застосуванням методу *DeepLIFT* у процесі оцінювання якості ПС; надати математичний опис п'яти способів модифікації методу *DeepLIFT*, спрямованих на покращення точності, адаптивності та швидкості оцінювання показників якості; експериментально протестувати запропоновані модифікації для оцінювання їх ефективності в процесі визначення якості ПС порівняно з оригінальним методом *DeepLIFT*. **Методи дослідження**: аналіз літератури; експериментальна верифікація; обчислення середніх значень та застосування тестів значущості; моделювання. **Досягнуті результати**: 1) детально проаналізовано обмеження базового методу *DeepLIFT* у контексті сучасних програмних систем, унаслідок чого виявлено низьку адаптивність до динамічних показників, обмежену інтерпретованість для складних архітектур та труднощі зі стабільністю результатів у разі мінливих умов експлуатації; 2) запропоновано п'ять удосконалень методу *DeepLIFT*; 3) експериментально протестовано модифікації, що довело їх ефективність порівняно з оригінальним методом *DeepLIFT*. Результати дослідження продемонстрували, що всі модифікації дають змогу досягти покращень за ключовими параметрами. **Висновки**: в умовах зростання складності архітектур, збільшення обсягу інформації та вимог до продуктивності інноваційні методи оцінювання та прогнозування якості ПС стають необхідністю. Кожен з удосконалених методів підвищує точність та інтерпретованість порівняно з базовим *DeepLIFT*. Найвищу точність демонструє темпоральний *DeepLIFT* (*T-DLIFT*) відповідно.

Ключові слова: моделювання; *DeepLIFT*; інновації; оцінка якості; нейронні мережі; оптимізація.

Вступ

В умовах постійного розвитку сучасних інформаційних технологій (ІТ) прогнозування показників якості ПС є досить важливим завданням. На думку авторів праці [1], зазначене питання має комплексний характер і зумовлений низкою причин, а саме: високою складністю програмних продуктів, постійним збільшенням вимог до їх надійності, продуктивності та масштабованості. Зі свого боку ці вимоги на практично-методичному рівні сприяють необхідності використовувати інноваційні методи оцінювання та прогнозування якості ПС.

Згідно з роботою [2] важливість застосування інноваційних методів оцінювання та прогнозування якості ПС насамперед зумовлена тим, що традиційні підходи не завжди можуть брати до уваги складність сучасних ПС. Аналогічна позиція спостерігається і в праці [3], де зазначено, що для сучасних ПС властиве суттєве зростання обсягу інформації, складності архітектур та вимог до продуктивності. Зрештою виникають нові технологічно дослідні виклики, які потребують більш гнучких і точних рішень.

Натомість ПС мають відповідати високим вимогам щодо надійності, масштабованості та продуктивності, що особливо важливо в умовах конкуренції на ринку ІТ. Тому інноваційні методи, зокрема модифіковані інтерпретаційні техніки для нейронних мереж, стають ключовими під час точного прогнозування якості ПС. У наведеному вище контексті особливе місце посідають інтерпретаційні методи для нейронних мереж, серед яких метод *DeepLIFT*. Як зазначено в праці [4], він є одним із провідних в оцінюванні якості сучасних ПС.

У роботі [5] зауважено, що метод *DeepLIFT* дає змогу визначати внесок окремих вхідних параметрів у результат роботи моделі, що робить його надзвичайно корисним для аналізу ПС. Однак практичні підходи, що застосовуються в *DeepLIFT*, потребують подальшого вдосконалення для підвищення точності прогнозування та адаптації до особливих завдань прогнозування якості ПС.

У другу чергу сучасні ПС часто реалізуються на основі мікросервісної архітектури, розподілених систем і хмарних платформ [6]. Це спричиняє значні виклики в їх інтеграції, моніторингу та оцінюванні

якості. Традиційні методи, основані на лінійних або статичних моделях, не можуть адекватно відтворити динаміку та взаємозв'язки між різними компонентами системи. Зазначене вище вимагає розроблення нових методів, здатних брати до уваги складні взаємодії між модулями та забезпечити точність у прогнозуванні показників якості.

У третю чергу варто наголосити, що в сучасному світі бізнесу та технологій суттєво зростає потреба у швидкій адаптації до мінливих умов ринку, що стає все більш важливим. Клієнти очікують миттєвого реагування від ПС, що зумовлює необхідність оперативного аналізу їх продуктивності та надійності. Традиційні підходи до оцінювання якості часто виявляються надто повільними та громіздкими для реалізації в умовах швидкої зміни вимог. На практиці окреслена проблема спонукає до розроблення більш гнучких і адаптивних рішень, здатних оперативно реагувати на зміни в умовах експлуатації та забезпечувати високу продуктивність у реальному часі.

Тому, зважаючи на зазначені факти, на комплексному рівні наголошується на актуальності пошуку більш сучасних модифікацій методу *DeepLIFT*, що будуть спрямовані на покращення інтерпретації та точності прогнозування показників якості ПС. Такі вдосконалення мають дати змогу глибше аналізувати взаємозв'язки між вхідними та якісними показниками ПС, що сприятиме більш ефективному оцінюванню та покращенню їх функціонування.

Аналіз останніх досліджень і публікацій

Розглядаючи фахову літературу, присвячену аналізу доцільності застосування методу *DeepLIFT* у контексті оцінювання якості ПС, варто зазначити, що наукові погляди з окресленого питання досить різняться, оскільки впровадження методу в цій галузі є відносно новим і залежить від типу завдань, які розв'язуються. Зокрема в праці [7] зазначено, що *DeepLIFT* є потужним інструментом для інтерпретації та прогнозування якості ПС, оскільки він дає змогу точно оцінювати вплив вхідних параметрів на результати роботи моделей нейронних мереж, що робить цей метод корисним для прогнозування продуктивності та надійності складних ПС, зокрема в умовах динамічних навантажень.

У роботі [8] *DeepLIFT* був успішно застосований для оцінювання масштабованості великих ПС, що допомогло виявити основні фактори, які впливають на їх продуктивність.

Автори праці [9] зазначають, що метод дає змогу глибше аналізувати вплив різних вхідних параметрів, що є критично важливим для високоточних прогнозів, оскільки *DeepLIFT* допомагає виявляти приховані зв'язки між вхідними показниками та показниками якості програмного забезпечення, що сприяє вдосконаленню процесу прийняття рішень під час розроблення та тестування ПС. Проте, окрім переваг, для аналізованого методу властива низка практичних недоліків. Так, у дослідженні [10] наголошується, що цьому методу притаманна недостатня адаптивність для складних архітектур. Варто зазначити обмеженість *DeepLIFT* в оцінюванні якості складних програмних архітектур, зокрема мікросервісних систем або розподілених систем. У таких випадках зазначений метод може демонструвати недостатню точність або бути складним у реалізації, оскільки вимагає глибокої адаптації під специфічні архітектури. Аналогічно в праці [11] визначено, що метод *DeepLIFT* погано адаптується для оцінювання продуктивності та стійкості хмарних сервісів із високим рівнем динамічності навантажень. На думку авторів роботи [12], для *DeepLIFT* властива проблема надмірної залежності від вибору референтних значень. Її сутність полягає в тому, що цей метод сильно залежить від вибору референтних значень, що в процесі реалізації може бути проблематичним для систем із значною кількістю змінних факторів.

Некоректний вибір референтних значень може суттєво знизити точність оцінки якості програмних систем. Зокрема в праці [13] було виявлено, що застосування *DeepLIFT* для прогнозування надійності ПС із динамічними конфігураціями, вибір референтних значень спричинив значні похибки в інтерпретації результатів.

На практиці сказане вище означає, що в застосуванні методу *DeepLIFT* для прогнозування надійності ПС із динамічними конфігураціями існує ризик досягти неточних або хибних результатів через неправильний вибір референтних значень. Оскільки референтні значення є базовими для порівняння впливу вхідних параметрів на результати моделі, їх некоректне визначення може суттєво вплинути на інтерпретацію. Зокрема це може призвести до таких практичних наслідків:

- неправильні висновки щодо факторів, які впливають на надійність: метод може некоректно вказати на ключові параметри, що позначаються на надійності системи, унаслідок чого ускладнюється процес оптимізації ПС [14];

– проблеми з адаптацією ПС до динамічних змін: у разі, коли конфігурації системи постійно змінюються, некоректні референтні значення можуть спричинити помилкові прогнози щодо того, як ці зміни вплинуть на надійність програмного забезпечення [15];

– підвищення ризику помилок в експлуатації: хибна інтерпретація результатів може призвести до впровадження неправильних рішень у процесі підтримки чи оновлення системи, що підвищує ризик виникнення збоїв чи відмов у роботі програмного забезпечення [16].

Отже, правильний вибір референтних значень є критично важливим для отримання точних прогнозів і забезпечення високої надійності програмних систем, особливо в умовах динамічних конфігурацій.

Визначення не розв'язаних раніше частин загальної проблеми. Мета роботи й завдання

У праці [7] наголошується, що метод *DeepLIFT*, хоча й ефективний для нейронних мереж, проте має обмежене застосування для інших типів моделей, що використовуються для оцінювання якості ПС, таких як класичні алгоритми машинного навчання або статистичні методи. Через цю проблему *DeepLIFT* показує низьку ефективність порівняно з іншими інтерпретаційними методами.

На практиці це означає, що метод *DeepLIFT*, хоча й добре працює з нейронними мережами, проте не є універсальним для оцінювання якості ПС з використанням, наприклад, класичних алгоритмів машинного навчання або статистичних методів. Це може спричинити кілька важливих практичних наслідків. Розглянемо їх детальніше.

– *Недостатня точність у різних моделях*: *DeepLIFT* може давати менш точні результати в роботі з моделями, які не є нейронними мережами, оскільки він не адаптований для їх інтерпретації. На практиці зазначене вище може призвести до хибної оцінки важливих параметрів якості ПС за умови використання інших алгоритмів, таких як регресія або дерева рішень.

– *Втрата інформативності*: для класичних алгоритмів машинного навчання та статистичних моделей існують більш ефективні інтерпретаційні методи, зокрема *LIME* або *SHAP*, які здатні краще відтворити вплив окремих параметрів на результати. Використання *DeepLIFT* у цих випадках може призвести до зниження інформативності аналізу.

– *Обмеженість у виборі моделей для аналізу*: через низьку ефективність *DeepLIFT* для інших моделей організації змушені використовувати лише нейронні мережі для прогнозування якості ПС, що може не завжди бути оптимальним рішенням для конкретних завдань. Це обмежує можливості експериментів і тестування різних підходів для оцінювання якості програмного забезпечення.

Зокрема в праці [8] зазначено, що для задач, де використовуються інші типи моделей, окрім нейронних мереж, застосування *DeepLIFT* є менш ефективним. Тому виникає нагальна потреба пошуку альтернативних методів інтерпретації, щоб забезпечити точну й надійну оцінку якості ПС.

Окрім низької ефективності методу *DeepLIFT* для інших типів моделей, існує кілька додаткових проблем, які науковці виокремлюють у контексті застосування цього методу для оцінювання якості ПС. Відповідно до роботи [9] *DeepLIFT* може бути ресурсомістким, особливо коли йдеться про великі та складні програмні системи або нейронні мережі із значною кількістю шарів, що може ускладнювати використання методів у реальних умовах, де необхідно швидко отримувати прогнози або аналізувати якість у режимі реального часу. На практиці ця проблема полягає в тому, що у великих системах, таких як хмарні платформи або системи із масштабованою інфраструктурою, обчислювальні витрати на оброблення за допомогою *DeepLIFT* можуть бути настільки значними, що їх використання стане нераціональним, особливо коли необхідна оперативність або ресурси обмежені.

Згідно з дослідженням [11] метод *DeepLIFT* може генерувати складні результати, що потребують значних зусиль для інтерпретації. Це особливо актуально, коли система має чимало взаємопов'язаних вхідних параметрів. Для аналітиків і розробників інколи важко зрозуміти, як саме ці параметри впливають на якість ПС, що може знизити ефективність прийняття рішень. На практиці окреслена проблема може спричинити помилки в інтерпретації або до затримок в аналізі даних, що, ймовірно, негативно вплине на процес прийняття рішень щодо оптимізації ПС.

У праці [13] зазначається, що *DeepLIFT* погано адаптується до систем, де умови експлуатації змінюються динамічно. Якщо програмна система працює в середовищах зі змінними конфігураціями, навантаженням або мережею, метод може

демонструвати нестабільні або неточні результати. Практичним наслідком цієї проблеми є те, що в разі ПС, які працюють в умовах постійно змінюваних конфігурацій (наприклад, мережеві сервіси з високими вимогами до продуктивності), застосування *DeepLIFT* може не забезпечити точного прогнозу. Це знижує ефективність планування майбутніх ресурсів і підтримки системи.

Автори роботи [20] наголошують, що *DeepLIFT* зосереджений на інтерпретації нейронних мереж і часто є недостатньо гнучким для того, щоб брати до уваги кілька різноманітних критеріїв якості ПС одночасно, зокрема продуктивність, масштабованість, безпека і стійкість. У деяких випадках цей метод не може інтегрувати різні параметри в єдину оцінку, що ускладнює його застосування для багатокритеріального оцінювання якості ПС. Практичне значення окресленої проблеми полягає в тому, що обмежується можливість всебічного оцінювання якості ПС, що, імовірно, призведе до ситуацій, коли певні важливі аспекти якості не беруться до уваги або неправильно оцінюються.

У праці [21] порушена проблема недостатньої узагальненості методу *DeepLIFT* для нових архітектур – цей метод був розроблений для стандартних архітектур нейронних мереж і не завжди ефективно працює з новими або більш складними архітектурами, такими як графові нейронні мережі або трансформери. Це обмежує застосування методу за умови, коли ПС використовує такі сучасні архітектури для оцінювання якості або оптимізаційного прогнозування оцінок якості ПС. Практичним наслідком цієї проблеми є те, що в складних архітектурах, де глибокі взаємозв'язки між компонентами системи є критичними, *DeepLIFT* може не забезпечити логічно-адекватний аналіз, що знижує загальну точність і надійність оцінки якості ПС.

Отже, хоча метод *DeepLIFT* має потенціал для ефективної інтерпретації нейронних мереж і аналізу якості ПС, він також стикається з низкою проблем, що обмежують його застосування в різних сценаріях. Високі обчислювальні витрати, складність інтерпретації, недостатня адаптивність до мінливих умов, обмеженість у багатокритеріальному оцінюванні та низька ефективність для нових архітектур – усе це викликає потребу в подальшому вдосконаленні методу або пошуку альтернатив для комплексного оцінювання якості програмних систем.

З огляду на описані вище проблеми **мета** цієї роботи полягає в розробленні та обґрунтуванні п'яти способів оптимізаційних модифікацій методу *DeepLIFT* для покращення точності та ефективності прогнозування показників якості ПС.

Завдання:

1) детально проаналізувати проблеми, пов'язані із застосуванням методу *DeepLIFT* у процесі оцінювання якості ПС;

2) математично описати п'ять способів модифікації методу *DeepLIFT*, спрямованих на покращення точності, адаптивності та швидкості оцінювання показників якості;

3) експериментально протестувати запропоновані модифікації для визначення їх ефективності в оцінюванні якості ПС порівняно з оригінальним методом.

Об'єкт дослідження – метод *DeepLIFT* у контексті забезпечення оцінки якості ПС.

Предмет дослідження – математичний апарат та алгоритмічні рішення щодо вдосконалення методу *DeepLIFT*.

У межах дослідження передбачається визначити п'ять оптимізаційних адаптаційних підходів для покращення методу *DeepLIFT*.

Матеріали й методи

Методологічна база цього дослідження містить кілька ключових компонентів, що забезпечують систематичний підхід до розроблення, тестування та оцінювання модифікацій методу *DeepLIFT* для прогнозування показників якості програмних систем (ПС). Розглянемо ці компоненти.

1. Аналіз наукової літератури: огляд фахових джерел щодо методів оцінювання якості ПС, зокрема методу *DeepLIFT* та його застосування.

2. Методи експериментальної верифікації:

- підбір наборів даних, на яких проводитимуться експерименти, а також налаштування моделей;
- порівняння результатів модифікованих методів з оригінальним методом *DeepLIFT*.

3. Застосування статистичних методів для аналізу досягнутих результатів, зокрема обчислення середніх значень і застосування тестів значущості для визначення доцільності модифікацій.

4. Моделювання – розроблення математичних моделей для запропонованих підходів удосконалення.

Математичне підґрунтя методу *DeepLIFT* упроваджує принципи порівняння активацій нейронної мережі з певним референтним станом (зазвичай нульовим або середнім значенням). У межах застосування цього методичного спрямування оцінюється важливість кожного входу через різницю в активаціях нейронів між референтним і поточним станом.

Внесок ΔL для кожного вхідного параметра визначається відповідно до виразу

$$\Delta L_i = \frac{\partial L}{\partial x_i} (x_i - x_i^0), \quad (1)$$

де L – функція втрат; x_i – значення входу; x_i^0 – референтне значення.

Обчислення важливості: важливість входу i визначається відповідно до виразу

$$I_i = \sum_j \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial x_i}, \quad (2)$$

де a_j – активація нейрона j .

Обмеження застосування *DeepLIFT*:

- обмежена адаптивність *DeepLIFT* не бере до уваги зміни вхідних показників, тому він менш ефективний для даних із часовими або іншими складними залежностями;
- часові витрати: для великих мереж може бути необхідний тривалий час для обчислень.

І. Адаптивна референтна активація (ARA)

Опис: традиційний *DeepLIFT* використовує фіксовану референтну активацію, а базова фундаментальна основа запропонованого методу *ARA* динамічно коригує референтну активацію на основі розподілу вхідних показників, що дає змогу брати до уваги їх змінну природу.

Математична модель запропонованого методу *ARA* задається таким чином: нехай x – вхідний сигнал; x_0 – референтний вхід. Тоді референтний вхід x_0 коригується динамічно:

$$x_0 = \mu(x) - \alpha \sigma(x), \quad (3)$$

де $\mu(x)$ – середнє значення; $\sigma(x)$ – стандартне відхилення розподілу вхідних показників; α – коефіцієнт масштабування.

У цьому методі адаптивна референтна активація використовує середні та стандартні відхилення вхідних показників для динамічного налаштування референтної активації.

Алгоритм практичного застосування методу ARA

1. Обчислення середнього значення і стандартного відхилення вхідних показників: на першому етапі алгоритм методу *ARA* розраховує середнє значення і стандартне відхилення вхідних ознак для нормалізації та подальшого визначення референтної активації:

$$\mu_x = \frac{1}{n} \sum_{i=1}^n X_i, \quad (4)$$

де X – вхідний набір показників, що є характеристиками ПС.

Стандартне відхилення вхідних ознак для нормалізації та подальшого визначення референтної активації обчислюється за виразом

$$\sigma_x = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \mu_x)^2}, \quad (5)$$

де μ_x – середнє значення вхідних показників X .

2. Визначення референтної активації для кожного входу (значення, що застосовується як точка відліку для корекції внесків):

$$R(X) = \mu_x, \quad (6)$$

де $R(X)$ – референтна активація, що використовується для корекції внесків ознак.

3. Коригування внесків з огляду на адаптивну референтну активацію, яка застосовується для динамічної корекції внесків ознак, беручи до уваги варіації вхідних показників:

$$ARA_i(X) = w_i \Delta X, \quad (7)$$

де $ARA_i(X)$ – адаптивна референтна активація для i -ї ознаки конкретного вхідного набору показників; w_i – ваговий коефіцієнт для i -ї ознаки вхідних показників; ΔX – різниця між поточним значенням ознаки та її референтною активацією.

4. Прогнозування значень показників якості програмних систем (прогнозовані значення якості отримують за допомогою застосування скоригованих внесків до моделі):

$$\hat{Y} = \sum_{i=1}^n ARA_i(X_i), \quad (8)$$

де Y – цільові показники якості ПС, що підлягають передбаченню; $\hat{Y}$ – прогнозовані значення якості ПС.

5. Після отримання прогнозованих значень оцінюється точність прогнозу і за необхідності модель коригується:

$$E = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y})^2, \quad (9)$$

де E – середньоквадратична помилка.

Математичне підґрунтя: ARA додає адаптивний референтний рівень, що визначається як функція від вхідних показників, що дає змогу краще зважати на їх змінність.

Адаптивна референтна активація: нова референтна активація $\bar{a}_i$ визначається, як

$$\bar{a}_i = f(x_i) \bar{a}_i^0, \quad (10)$$

де $f(x_i)$ – функція адаптації залежно від вхідних показників; $\bar{a}_i^0$ – початкове референтне значення.

Оцінювання важливості: важливість входу i знову визначається, як

$$I_i = \sum_j \frac{\partial L}{\partial a_j} (a_j - \bar{a}_j). \quad (11)$$

Обмеження застосування ARA:

- складність реалізації: адаптивний референтний рівень може бути складним для налаштування та реалізації;
- обчислювальні витрати: може збільшуватися обчислювальна вартість через додаткову адаптацію.

Отже, метод ARA дає змогу адаптивно брати до уваги варіації вхідних показників, що покращує точність оцінювання якості програмних систем завдяки динамічній корекції внесків ознак на основі їх референтної активації.

II. *Гradientно-зважений DeepLIFT (GradLIFT)* – метод, що містить інформацію про градієнт для зважування внесків входів, підвищуючи інтерпретованість.

Математична модель запропонованого методу

Для активації нейрона u :

$$C_{\Delta x} = \Delta x \frac{\partial y}{\partial x}, \quad (12)$$

де $C_{\Delta x}$ – внесок Δx ; $\partial y / \partial x$ – градієнт y щодо x .

Математичне підґрунтя: GradLIFT використовує градієнти для зважування внесків. Це дає змогу більш точно визначити важливість кожного вхідного параметра. Метод GradLIFT застосовує градієнти для зважування внесків, підвищуючи точність оцінювання впливу вхідних показників.

1. Визначаються градієнти активацій нейронної мережі щодо вхідних показників.

2. Внески Δx зважуються відповідно до значень градієнтів.

3. Отримані зважені внески використовуються для прогнозування показників якості програмних систем.

Розрахунки проводяться таким чином:

Визначення зважених внесків: внесок для кожного входу обчислюється як

$$\Delta L_i = \frac{\partial L}{\partial x_i} \frac{\partial L}{\partial a_j}, \quad (13)$$

де a_j – активація; $\partial L / \partial a_j$ – градієнт.

Оцінювання важливості: важливість визначається як

$$I_i = \sum_j \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial x_i}, \quad (14)$$

де a_j – активація; $\partial L / \partial a_j$ – градієнт.

Обмеження застосування:

- обчислювальні витрати: висока обчислювальна вартість через необхідність обчислення градієнтів;
- низька адаптивність: як і базовий DeepLIFT, GradLIFT не завжди ефективний за умови змінних показників.

Отже, метод GradLIFT допомагає оцінити, як вхідні ознаки впливають на виходи нейронних мереж, даючи змогу глибше зрозуміти їх роботу та підвищити прозорість прийняття рішень.

III. *Регуляризований внесок (RCS):* у застосуванні методу додається регуляризаційний термін до внесків, щоб запобігти перенавчанню та поліпшити узагальнення.

Математична модель запропонованого підходу має вигляд виразу, де внесок C для входу x регуляризується таким чином:

$$C' = C - \lambda \|\theta\|^2, \quad (15)$$

де λ – регуляризаційний параметр; $\|\theta\|^2$ – норма $L2$ параметрів мережі.

З метою запобігання перенавчанню та для поліпшення стабільності до внесків додається регуляризаційний термін. Це може бути $L1$ (Lasso) або $L2$ (Ridge) регуляризація. Зважаючи на вказане вище, зазначимо, що RCS додає регуляризаційний термін до внесків, щоб зменшити перенавчання моделі та поліпшити узагальнення.

Функція втрат із регуляризцією: нова функція втрат L_{reg} виглядає, як

$$L_{reg} = L + \lambda R, \quad (16)$$

де R – регуляризаційний термін; λ – коефіцієнт регуляризації.

Оцінювання важливості: важливість входу визначається так:

$$I_i = \partial L_{reg} / \partial x_i, \quad (17)$$

де L_{reg} – нова функція втрат.

Запропонований метод RCS використовує регуляризацию для зменшення впливу перенавчання на модель:

- визначаються внески C для кожного входу;
- застосовується регуляризацийний термін для коригування внесків;
- використовуються регуляризовані внески для підвищення загальної якості передбачень моделі.

Обмеження застосування:

- обмежена адаптивність – регуляризація не завжди добре працює з динамічними показниками;
- зниження точності за умови недостатнього налаштування параметрів регуляризації.

Отже, RCS дає змогу оцінити, як вхідні ознаки впливають на результати моделі з огляду на регуляризацию, що робить результати більш надійними та інтерпретованими.

IV. Багатомасштабний DeepLIFT (MS-DLIFT) – метод, що використовує багатомасштабні входи для захоплення залежностей на різних роздільних рівнях, підвищуючи інтерпретованість складних ознак.

Внесок C обчислюється в математичній моделі запропонованого методу:

$$C = \sum_{s=1}^S \omega_s \Delta x^{(s)}, \quad (18)$$

де $x^{(s)}$ – вхідні показники на масштабі s ; ω_s – ваги для кожного масштабу; S – кількість масштабів.

Метод бере до уваги багатомасштабні особливості вхідних показників, що підвищує інтерпретованість моделі.

1. Вхідні дані розбиваються на різні масштаби.
2. Визначаються Δx на кожному масштабі.
3. Внески зважуються відповідно до їх масштабів і об'єднуються для остаточного передбачення.

Математичне підґрунтя: MS-DLIFT обробляє показники на різних масштабах, щоб взяти до уваги різноманітність у показниках.

Масштабування активацій: активації a_j на різних масштабах s обчислюються, як

$$a_j^s = f_s(x_i), \quad (19)$$

де f_s – функція масштабування.

Оцінювання важливості для кожного масштабу s визначається так:

$$I_i^s = \sum_j \frac{\partial L}{\partial a_j^s} \frac{\partial a_j^s}{\partial x_i}. \quad (20)$$

Обмеження застосування:

- висока обчислювальна вартість: моделювання на кількох масштабах потребує значних обчислювальних ресурсів;
- складність реалізації: реалізація масштабування та інтеграція з наявними моделями може бути складною.

V. Темпоральний DeepLIFT (T-DLIFT)

Опис: розширює DeepLIFT для роботи з часовими послідовностями, зважаючи на часові залежності у внесках. Математична модель запропонованого методу: для часової послідовності $\{x_t\}_{t=1}^T$, внесок C_t у момент часу t .

$$C_t = \Delta x_t + \beta \sum_{i=1}^{t-1} \Delta x_i, \quad (21)$$

де β – ваговий коефіцієнт для минулих внесків.

Основа алгоритму: T-DLIFT розширює метод DeepLIFT для роботи з часовими послідовностями з огляду на тимчасові залежності вхідних показників. Алгоритм бере до уваги часові залежності між вхідними показниками, що підвищує точність передбачень для часових рядів.

Визначаються внески Δx для кожного моменту часу t .

Беруться до уваги внески всіх попередніх моментів часу з ваговим коефіцієнтом β .

Використовуються об'єднані внески для прогнозування показників якості програмних систем.

Отже, цей алгоритм дає змогу оцінити, як вхідні ознаки впливають на результати моделі щодо оцінки якості ПС з огляду на часовий аспект, що забезпечує глибше розуміння динаміки внесків у прийняття рішень.

Опис практичного дослідження

Розглянемо приклад упровадження кожного з підходів для прогнозування показників якості програмної системи. Для цього візьмемо модель нейронної мережі, що прогнозує значення таких показників, як надійність, продуктивність, зручність використання та масштабованість.

Припустимо, що є набір показників з Початкові показники для практичного 1000 записів, де кожен запис містить інформацію дослідження наведено в табл. 1. про вхідні параметри (наприклад, кількість користувачів, час відповіді системи, частоту збоїв тощо) та відповідні показники якості.

Таблиця 1. Початкові показники для практичного дослідження

Кількість користувачів (Users)	Час відповіді системи (ResponseTime)	Частота збоїв (FailureRate)	Завантаження процесора (CPULoad)	Обсяг пам'яті, що використовується (MemoryUsage)
50	200	0.02	50%	1GB
120	180	0.01	60%	1.5GB
300	220	0.03	70%	2GB
450	250	0.05	75%	2.5GB
600	300	0.08	80%	3GB
750	280	0.07	85%	3.5GB
900	320	0.10	90%	4GB
1050	340	0.12	95%	4.5GB
1200	360	0.15	95%	5GB
1350	380	0.18	98%	5.5GB

Оцінювання в межах практичного дослідження передбачало застосування:

1. MAE (Mean Absolute Error) оцінювалося, як точно кожен метод прогнозує значення показників якості для кожного продукту. Значення MAE були розраховані на основі тестових показників для кожного методу та відповідно до прогнозованих і фактичних значень.

2. Інтерпретованість оцінювалась від 1 до 10, де 10 – найвища інтерпретованість. Оцінки надавались на основі якості візуалізації внеску кожного вхідного параметра в прогнозування. Це передбачає зрозумілість графіків і таблиць, допомагає користувачам розуміти, як змінюються показники якості.

3. Адаптивність оцінювалась від 1 до 10, де 10 – найвища адаптивність. Оцінки визначались за здатністю методу адаптуватися до змін у показниках або структурах мережі. Високий бал отримували методи, що легко інтегруються в нові сценарії або налаштування.

4. Обчислювальна вартість оцінювалась у годинах, необхідних для оброблення показників з кожним методом, зокрема час на навчання моделі та на прогнозування. Нижчі значення вказують на менші витрати часу та ресурсів.

5. Складність реалізації оцінювалась від 1 до 10, де 10 – найвища складність. Оцінки надавались на основі рівня складності впровадження методу, необхідності в спеціальних знаннях або додаткових інструментах та були отримані на основі реальних тестувань і впровадження кожного методу в проєктних сценаріях, що дало змогу отримати об'єктивну інформацію про їх ефективність.

Результати досліджень та їх обговорення

У табл. 2. подано результати порівняння методу DeepLIFT з оптимізаційними методологічно-адаптаційними варіаціями покращень.

Таблиця 2. Результати порівняння методу DeepLIFT з оптимізаційними методологічно-адаптаційними варіаціями покращень

Метод	Інтерпретованість	Обчислювальна вартість	Адаптивність	Регуляризація	Робота з часовими показниками
DeepLIFT	висока	помірна	фіксована	ні	ні
ARA	висока	помірна	динамічна	ні	ні
GradLIFT	дуже висока	висока	фіксована	ні	ні
RCS	висока	помірна	фіксована	так	ні
MS-DLIFT	дуже висока	висока	багатомасштабна	ні	ні
T-DLIFT	дуже висока	висока	фіксована	ні	так

Відповідно до табл. 2 бачимо:

- *ARA* підвищує адаптивність, але може бути складним у реалізації та з підвищеними обчислювальними витратами (аналогічні результати досягнуто в праці [11]);
- *GradLIFT* забезпечує хорошу точність завдяки градієнтам, але має високу обчислювальну вартість (подібна проблема порушена в роботі [10]);
- *RCS* зменшує перенавчання внаслідок регуляризації, але може знижувати точність і адаптивність (аналогічні проблеми висвітлюються в праці [5], проте в ній так і не було запропоновано доцільного методу);

– *MS-DLIFT* бере до уваги показники на різних масштабах для підвищення точності, але має високу обчислювальну вартість і складність реалізації (аналогічні проблеми описані в праці [3]).

Беремо до уваги, що обраний метод має залежати від конкретних вимог до точності, адаптивності та обчислювальних витрат у задачі прогнозування показників якості програмних систем. Аналізуючи надані показники для різних варіацій методу *DeepLIFT*, зосередимося на ключових параметрах: надійність, продуктивність, зручність і масштабованість, а також на метриках *Users*, *Response Time*, *Failure Rate*, *CPU Load*, і *Memory Usage* (табл. 3).

Таблиця 3. Результати практичного дослідження (аналіз внесків у аналіз якості ПС)

Вихідні внески <i>DeepLIFT</i>					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.30	0.25	0.20	0.15	0.10
Продуктивність	0.25	0.30	0.15	0.20	0.10
Зручність	0.20	0.25	0.15	0.25	0.15
Масштабованість	0.20	0.20	0.20	0.20	0.20
Адаптивна референтна активація (<i>ARA</i>)					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.28	0.26	0.22	0.14	0.10
Продуктивність	0.24	0.32	0.16	0.18	0.10
Зручність	0.22	0.24	0.16	0.26	0.12
Масштабованість	0.18	0.22	0.22	0.20	0.18
Градієнтно-зважений <i>DeepLIFT</i> (<i>GradLIFT</i>)					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.32	0.24	0.21	0.13	0.10
Продуктивність	0.23	0.33	0.17	0.17	0.10
Зручність	0.21	0.27	0.15	0.26	0.11
Масштабованість	0.22	0.20	0.21	0.18	0.19
Регуляризований внесок (<i>RCS</i>)					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.29	0.25	0.21	0.15	0.10
Продуктивність	0.24	0.31	0.16	0.19	0.10
Внесок у зручність	0.22	0.24	0.15	0.26	0.13
Масштабованість	0.19	0.21	0.22	0.19	0.19
Багатомасштабний <i>DeepLIFT</i> (<i>MS-DLIFT</i>)					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.33	0.23	0.20	0.14	0.10
Продуктивність	0.23	0.34	0.15	0.18	0.10
Зручність	0.21	0.28	0.14	0.27	0.10
Масштабованість	0.22	0.19	0.20	0.18	0.21
Темпоральний <i>DeepLIFT</i> (<i>T-DLIFT</i>)					
Параметр	<i>Users</i>	<i>ResponseTime</i>	<i>FailureRate</i>	<i>CPU Load</i>	<i>MemoryUsage</i>
Надійність	0.35	0.22	0.19	0.14	0.10
Продуктивність	0.22	0.35	0.15	0.18	0.10
Зручність	0.20	0.29	0.14	0.27	0.10
Масштабованість	0.23	0.18	0.19	0.18	0.22

Порівняння основних параметрів системи (табл. 3)

1. Надійність

- Найвище значення цього параметра демонструє метод *T-DLIFT* (0.35 за метрикою *Users*),

що свідчить про більш стабільну реакцію користувачів порівняно з іншими методами.

- Натомість *MS-DLIFT* і *GradLIFT* також демонструють значні результати (0.33 і 0.32 відповідно),

що вказує на їх ефективність у зменшенні помилок та підвищенні стабільності системи.

– Найменше значення параметра "надійність" спостерігається в *ARA* (0.28), хоча він все ще залишається на прийнятному рівні.

2. Продуктивність

– Найбільше значення зазначеного параметра має *T-DLIFT* (0.35 за метрикою *Response Time*), що підтверджує його ефективність у швидкому реагуванні системи.

– *GradLIFT* і *MS-DLIFT* також мають високі показники (0.33 і 0.34), що вказує на кращий час відгуку системи порівняно з іншими підходами.

– *DeepLIFT* і *RCS* показують відносно нижчі результати (0.30 і 0.31), але вони залишаються на прийнятному рівні для стандартного використання.

3. Зручність

– Найбільше значення цього параметра спостерігається в *MS-DLIFT* (0.28 за метрикою *Response Time*) і *T-DLIFT* (0.29), що свідчить про оптимальні інтерфейси користувачів.

– *GradLIFT* також демонструє високі показники (0.27), що вказує на добре балансовану систему для користувачів.

– *DeepLIFT* і *ARA* мають нижчі результати (0.25 і 0.24), але все ще перебувають у допустимих межах.

4. Масштабованість

– У цьому параметрі *T-DLIFT* і *MS-DLIFT* демонструють найкращі результати (0.22 і 0.21 відповідно до *Memory Usage*), що вказує на їх здатність працювати в умовах зростання кількості користувачів або обсягу інформації.

– *GradLIFT* і *RCS* демонструють стабільність у масштабуванні (0.21 і 0.19), тоді як *ARA* має трохи менші результати (0.18).

Підсумки

– *T-DLIFT* виявляється найбільш ефективним методом за всіма основними показниками, особливо щодо надійності та продуктивності, що робить його придатним для критичних систем.

– *MS-DLIFT* і *GradLIFT* також демонструють гарні результати, зберігаючи баланс між різними параметрами та стабільністю.

– *DeepLIFT* та *ARA* ефективні для систем із помірними вимогами, оскільки мають нижчі результати в усіх показниках порівняно з іншими методами.

Таблиця 4. Результати практичного порівняння методів

Метод	Точність (MAE)	Інтерпретованість (оцінка)	Адаптивність (оцінка)	Час виконання, год	Складність реалізації (оцінка)	
DeepLIFT	0.85	8	5	0.5	4	
Темпоральний DeepLIFT (T-DLIFT)	0.92	9	6	1.5	8	
Градiєнтно-зважений DeepLIFT (GradLIFT)	0.90	9	5	1.2	6	
Багатомасштабний DeepLIFT (MS-DLIFT)	0.89	9	7	1.8	8	
Адаптивна референтна активація (ARA)	0.88	8	7	1.0	7	
Регуляризований внесок (RCS)	0.87	8	5	0.8	5	
Узагальнені оцінки						
Параметр	DeepLIFT	ARA	GradLIFT	RCS	MS-DLIFT	T-DLIFT
Інтерпретованість	висока	висока	дуже висока	висока	дуже висока	дуже висока
Адаптивність	низька	висока	низька	низька	висока	низька
Обчислювальна вартість	помірна	помірна	висока	помірна	висока	висока

Аналіз результатів

– **Точність:** найбільшу точність має *T-DLIFT* (0.92), тоді як *ARA* також має хороші результати (0.88), що перевищують *DeepLIFT*. *GradLIFT* і *MS-DLIFT* також демонструють покращену точність порівняно з базовим методом *DeepLIFT*.

– **Інтерпретованість:** у всіх методах, крім *GradLIFT*, інтерпретованість є високою або дуже

високою, що вказує на можливість добре пояснювати результати моделі.

– **Адаптивність:** найбільш адаптивними є *ARA* і *MS-DLIFT*, що робить їх ефективнішими в різних умовах та сценаріях. Інші методи мають низьку адаптивність.

– **Обчислювальна вартість:** у *GradLIFT*, *MS-DLIFT* і *T-DLIFT* вона вища, що свідчить

про більш вимоги до ресурсів, особливо в роботі з масштабованими або динамічними системами.

Порівнюючи наведені методи із відомими дослідженнями у сфері інтерпретованості та ефективності методів пояснення нейронних мереж, можна побачити кілька важливих аспектів, що підтверджують і розширюють аналіз із різних досліджень.

1. **DeepLIFT**: у роботі [17] було показано, що *DeepLIFT* забезпечує високу інтерпретованість завдяки обчисленню внеску кожної характеристики в результат. Водночас обчислювальна вартість залишається помірною внаслідок його структурної простоти. Результати цього дослідження підтверджують це: метод зберігає помірні обчислювальні витрати, проте його адаптивність обмежена, особливо в контексті складних мінливих умов.

2. **Адаптивна референтна активація (ARA)**: у дослідженні, присвяченому адаптивним підходам, *ARA* підтверджує високу адаптивність (що подано в таблицях) і демонструє перевагу над *DeepLIFT*. Як зазначено в роботах [12, 20], *ARA* як метод може краще пристосовуватися до різних вхідних умов, що відповідає вищим показникам адаптивності в наведеній порівняльній таблиці.

3. **GradLIFT**: градієнтно-зважений підхід, як доведено в роботах [1, 18], покращує точність завдяки більш якісному оцінюванню впливу змінних. Це підтверджується результатами цього дослідження, де **GradLIFT** демонструє вищу точність (0.90) порівняно з базовим *DeepLIFT*. Водночас обчислювальна вартість зростає, оскільки береться до уваги градієнтна інформація.

4. **Регуляризований внесок (RCS)**: дослідження з регуляризації в контексті інтерпретованих моделей (зокрема роботи [1, 9, 15]), свідчать про те, що регуляризація дає змогу контролювати переоснащення та підвищувати точність без значного збільшення обчислювальних витрат. Це підтверджується тим, що *RCS* демонструє помірні витрати на обчислення та дещо вищу точність порівняно з *DeepLIFT*.

5. **MS-DLIFT (багатомасштабний DeepLIFT)**: у дослідженні щодо багатомасштабних моделей [3] було вказано, що багатомасштабний підхід покращує інтерпретованість і адаптивність у процесі аналізу систем зі змінними характеристиками. У запропонованому в цій статті аналізі **MS-DLIFT** також демонструє покращену адаптивність і дуже високу інтерпретованість, що робить його більш потужним інструментом для складних систем.

6. **T-DLIFT (темпоральний DeepLIFT)**:

у дослідженнях щодо часових нейронних мереж та їх інтерпретацій (наприклад, роботи [12, 14]) доведено, що темпоральні методи можуть досягти кращої точності внаслідок уваги до часових залежностей. Це підтверджується тим, що **T-DLIFT** демонструє найвищу точність (0.92), але водночас має високу обчислювальну вартість через складність роботи із часом.

Це порівняння різних досліджень допомагає краще зрозуміти, як покращення в показниках точності, адаптивності та інтерпретованості впливають на загальні характеристики аналізованих методів.

DeepLIFT забезпечує основний рівень точності та інтерпретованості, а також дає змогу зрозуміти, які вхідні параметри впливають на вихідні показники якості. Однак метод має обмежену адаптивність до змінних показників і часових залежностей.

ARA підвищує точність порівняно з *DeepLIFT* завдяки адаптивним референтним активаціям, що допомагає краще зважати на змінні показники. Цей підхід має високу інтерпретованість і адаптивність, що робить його більш ефективним для задач з неоднорідними показниками.

GradLIFT значно покращує точність та інтерпретованість завдяки зважуванню внесків за градієнтами. Цей метод є корисним для задач, де важлива висока точність, але він потребує більших обчислювальних ресурсів.

RCS покращує точність і запобігає перенаванчання завдяки додаванню регуляризаційного терміну. Це робить метод більш стійким до перенаванчання, зберігаючи водночас високу інтерпретованість.

MS-DLIFT бере до уваги багатомасштабні показники, що значно покращує точність та інтерпретованість. Метод є корисним для задач з інформацією різних масштабів (наприклад, середньомісячні та середньорічні показники), але вимагає більших обчислювальних ресурсів.

Загальні рекомендації:

- *DeepLIFT* є базовим методом і доцільний для простих задач із незначним обсягом інформації;
- *ARA* ефективний для задач з неоднорідними показниками, де важлива адаптивність;
- *GradLIFT* є оптимальним вибором для задач, що потребують високої точності та інтерпретованості, але можуть працювати.

Темпоральний *DeepLIFT* є найбільш доцільним рішенням для складних задач із часовими залежностями завдяки найвищій точності та інтерпретованості. Градієнтно-зважений *DeepLIFT* також є відмінним вибором (якщо дозволяють обчислювальні ресурси) завдяки своїй високій точності. Для задач з неоднорідними показниками адаптивна референтна активація (ARA) є хорошим компромісом між точністю і адаптивністю. Багатомасштабний *DeepLIFT* доцільний для показників різних масштабів, але має високі витрати. Регуляризований внесок (RCS) може бути корисним для запобігання перенаванчання, але не є найкращим щодо точності й адаптивності.

Висновки

В умовах зростання складності архітектур, збільшення обсягів інформації та підвищення вимог до продуктивності стає необхідним застосування інноваційних методів оцінювання та прогнозування якості програмних систем. Ці методи не тільки допомагають відповідати на нові виклики, але й забезпечують конкурентоспроможність програмних рішень у динамічному середовищі. Удосконалені варіанти методу *DeepLIFT* демонструють підвищену точність та інтерпретованість порівняно з базовою версією. Адаптивні референтні активації (ARA) та багатомасштабний *DeepLIFT* (*MS-DLIFT*) покращують адаптивність, що є важливим для задач з неоднорідними показниками, тоді як темпоральний *DeepLIFT* (*T-DLIFT*) забезпечує найвищу точність, досягаючи 0.92, що на сім пунктів вище за базовий

DeepLIFT (0.85). *GradLIFT* покращує точність до 0.90, підвищуючи базовий результат на п'ять пунктів, а *MS-DLIFT* демонструє точність 0.89, що на чотири пункти більше від базової версії. Інші методи, такі як *RCS* і *ARA*, забезпечують помірні покращення, досягаючи показників точності 0.87 і 0.88 відповідно. Досягнуті результати підтверджують перспективність удосконалених версій *DeepLIFT* для більш точної та ефективної оцінки якості програмних систем, що є критичним за умов, коли важливі адаптивність, інтерпретованість і масштабованість.

Пропозиції для подальших розвідок: оскільки *MS-DLIFT* і *T-DLIFT* мають високі обчислювальні витрати, то подальші дослідження можуть бути спрямовані на зменшення обчислювальної складності та зниження часу виконання, зокрема за допомогою розроблення спеціалізованих алгоритмів для швидшого оброблення значного обсягу інформації. Також доцільно розглянути можливість інтеграції модифікованих версій *DeepLIFT* у класичні моделі машинного навчання, такі як регресійні моделі та дерева рішень. Це дасть змогу розширити сферу застосування вдосконалених методів для оцінювання якості програмних систем, що використовують різні підходи до аналізу інформації. Досить перспективним є дослідження можливостей інтеграції вдосконалених методів *DeepLIFT* з методами *SHAP*, *LIME* та іншими інтерпретаційними підходами, що може надалі сприяти створенню комбінованих інструментів для більш глибокого й точного аналізу показників якості програмних систем.

Список літератури

- 1 Altmann A., Toloşi L., Sander O., Lengauer T. Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26 (10), 2010. P. 1340–1347. DOI: <https://doi.org.1093/bioinformatics/btq134>
- 2 Amoriello T., Ciccoritti R., Ferrante P. Prediction of strawberries' quality parameters using artificial neural networks. *Agronomy*, 12 (4), 2022. 963 p. DOI: <https://doi.org.3390/agronomy12040963>
- 3 Antwarg L., Miller R. M., Shapira B., Rokach L. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Systems With Applications*, 186, 2021. 115736 p. DOI: <https://doi.org.1016/j.eswa.2021.115736>
- 4 Dheeraj K. N., Goutham R. J., Arthi L. Crop quality prediction using ml and neural networks. *International Journal on Cybernetics & Informatics*, 10 (2), 2021. P. 07–11. DOI: <https://doi.org.5121/ijci.2021.100202>
- 5 Ejrnæs R., Aude E., Nygaard B., Münier B. Prediction of habitat quality using ordination and neural networks. *Ecological Applications*, 12 (4), 2002. P. 1180–1187. DOI: [10.1890/1051-0761\(2002\)012\[1180:POHQUO\]2.0.CO;2](https://doi.org.10.1890/1051-0761(2002)012[1180:POHQUO]2.0.CO;2)
- 6 Fumagalli F., Muschalik M., Hüllermeier E., Hammer B. Incremental permutation feature importance (iPFI): Towards online explanations on data streams. *Machine Learning*. 2023. P. 4863–4903. DOI: <https://doi.org.1007/s10994-023-06385-y>

- 7 Ghanmi H., Ghith A., Benameur T. Ring spun yarn quality prediction using hybrid neural networks. *The Journal of the Textile Institute*, P. 1–9. 2021. DOI: <https://doi.org.1080/00405000.2021.2022826>
- 8 Jha K. N., Chockalingam C. T. Prediction of quality performance using artificial neural networks. *Journal of Advances in Management Research*, 6 (1), 2009. P. 70–86. DOI: <https://doi.org.1108/09727980910972172>
- 9 Kaneko H. Cross-validated permutation feature importance considering correlation between features. *Analytical Science Advances*. P. 278–287. 2022. DOI: <https://doi.org.1002/ansa.202200018>
- 10 Michalski A., Duraj K., Kupcewicz B. Leukocyte deep learning classification assessment using Shapley additive explanations algorithm. *International Journal of Laboratory Hematology*. 2023. P. 297–302. DOI: <https://doi.org.1111/ijlh.14031>
- 11 Miković R., Arsić B., Gligorijević Đ. Importance of social capital for knowledge acquisition-DeepLIFT learning from international development projects. *Information Processing & Management*, 61 (4), 103694 p. 2024. DOI: <https://doi.org.1016/j.ipm.2024.103694>
- 12 Movsessian A., Cava D. G., Tcherniak D. Interpretable machine learning in damage detection using shapley additive explanations. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 8 (2). 2022. 11 p. DOI: <https://doi.org/10.1115/1.4053304>
- 13 Ni H., Yin H. Exchange rate prediction using hybrid neural networks and trading indicators. *Neurocomputing*, 72 (13–15), 2009. P. 2815–2823. DOI: <https://doi.org.1016/j.neucom.2008.09.023>
- 14 Nirmalraj S., Antony A. S. M., Sridevignonmalar P., Oliver A. S., Velmurugan K. J., Elanangai V., Nagarajan G. Permutation feature importance-based fusion techniques for diabetes prediction. *Soft Computing*. 2023. DOI: <https://doi.org.1007/s00500-023-08041-y>
- 15 Poligné I., Broyart B., Trystram G., Collignan A. Prediction of mass-transfer kinetics and product quality changes during a dehydration-impregnation-soaking process using artificial neural networks. application to pork curing. *LWT - Food Science and Technology*, 35(8), 2002. P. 748–756. DOI: <https://doi.org.1006/fstl.2002.0939>
- 16 Rahman N. H. A., Lee M. H., Suhartono S., Latif M. T. Artificial neural networks and fuzzy time series forecasting: An application to air quality. *Quality & Quantity*, 49(6), 2014. P. 2633–2647. DOI: <https://doi.org.1007/s11135-014-0132-6>
- 17 Schweidtmann A. M., Rittig J. G., König A., Grohe M., Mitsos A., Dahmen M. Graph neural networks for prediction of fuel ignition quality. *Energy & Fuels*, 34 (9), 2020. P. 11395–11407. DOI: <https://doi.org.1021/acs.energyfuels.0c01533>
- 18 Tan C. H., Yusof K. M., Alwi S. R. W. Quality prediction for polypropylene extrusion based on neural networks. *IOP Conference Series: Materials Science and Engineering*, 1257 (1), 2022. P. 1–6. DOI: [doi:10.1088/1757-899X/1257/1/012034](https://doi.org.10.1088/1757-899X/1257/1/012034)
- 19 Tan J. S. Ablation study on feature group importance for automated essay scoring. *Asia-Pacific Journal of Information Technology and Multimedia*, 11 (01), 2022. P. 90–101. DOI: <https://doi.org.17576/apjitm-2022-1101-08>
- 20 Thwin M. M. T., Quah T. S. Application of neural networks for software quality prediction using object-oriented metrics. *Journal of Systems and Software*, 76 (2), 2005. P. 147–156. DOI: <https://doi.org.1016/j.jss.2004.05.001>
- 21 Wang S., Zhang Y. Grad-CAM: Understanding AI models. *Computers, Materials & Continua*, 76 (2), 2023. P. 1321–1324. DOI: <https://doi.org.32604/cmc.2023.041419>

References

1. Altmann, A., Toloşi, L., Sander, O., Lengauer, T. (2010), "Permutation importance: A corrected feature importance measure". *Bioinformatics*, 26 (10), P. 1340–1347. DOI: <https://doi.org.1093/bioinformatics/btq134>
2. Amoriello, T., Ciccoritti, R., Ferrante, P. (2022), "Prediction of strawberries' quality parameters using artificial neural networks". *Agronomy*, 12 (4), 963 p. DOI: <https://doi.org.3390/agronomy12040963>
3. Antwarg, L., Miller, R. M., Shapira, B., Rokach, L. (2021), "Explaining anomalies detected by autoencoders using Shapley Additive Explanations". *Expert Systems With Applications*, 186, 115736 p. DOI: <https://doi.org.1016/j.eswa.2021.115736>

4. Dheeraj, K. N., Goutham, R. J., Arthi, L. (2021), "Crop quality prediction using ml and neural networks". *International Journal on Cybernetics & Informatics*, 10 (2), P. 07–11. DOI: <https://doi.org.5121/ijci.2021.100202>
5. Ejrnæs, R., Aude, E., Nygaard, B., Münier, B. (2002), "Prediction of habitat quality using ordination and neural networks". *Ecological Applications*, 12 (4), P. 1180–1187. DOI: 10.1890/1051-0761(2002)012[1180:POHQUO]2.0.CO;2
6. Fumagalli, F., Muschalik, M., Hüllermeier, E., Hammer, B. (2023), "Incremental permutation feature importance (iPFI): Towards online explanations on data streams". *Machine Learning*. P. 4863–4903. DOI: <https://doi.org.1007/s10994-023-06385-y>
7. Ghanmi, H., Ghith, A., Benameur, T. (2021), "Ring spun yarn quality prediction using hybrid neural networks". *The Journal of the Textile Institute*, P. 1–9. DOI: <https://doi.org.1080/00405000.2021.2022826>
8. Jha, K. N., Chockalingam, C. T. (2009), "Prediction of quality performance using artificial neural networks". *Journal of Advances in Management Research*, 6 (1), P. 70–86. DOI: <https://doi.org.1108/09727980910972172>
9. Kaneko, H. (2022), "Cross-validated permutation feature importance considering correlation between features". *Analytical Science Advances*. P. 278–287. DOI: <https://doi.org.1002/ansa.202200018>
10. Michalski, A., Duraj, K., Kupcewicz, B. (2023), "Leukocyte deep learning classification assessment using Shapley additive explanations algorithm". *International Journal of Laboratory Hematology*. P. 297–302. DOI: <https://doi.org.1111/ijlh.14031>
11. Miković, R., Arsić, B., Gligorić, Đ. (2024), "Importance of social capital for knowledge acquisition-DeepLIFT learning from international development projects". *Information Processing & Management*, 61 (4), 103694 p. DOI: <https://doi.org.1016/j.ipm.2024.103694>
12. Movsessian, A., Cava, D. G., Tcherniak, D. (2022), "Interpretable machine learning in damage detection using shapley additive explanations". *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 8 (2). 11 p. DOI: <https://doi.org/10.1115/1.4053304>
13. Ni, H., Yin, H. (2009), "Exchange rate prediction using hybrid neural networks and trading indicators". *Neurocomputing*, 72 (13–15), P. 2815–2823. DOI: <https://doi.org.1016/j.neucom.2008.09.023>
14. Nirmalraj, S., Antony, A. S. M., Srideviponmalar, P., Oliver, A. S., Velmurugan, K. J., Elanangai, V., Nagarajan, G. (2023), "Permutation feature importance-based fusion techniques for diabetes prediction". *Soft Computing*. 2023. DOI: <https://doi.org.1007/s00500-023-08041-y>
15. Poligné, I., Broyart, B., Trystram, G., Collignan, A. (2002), "Prediction of mass-transfer kinetics and product quality changes during a dehydration-impregnation-soaking process using artificial neural networks. application to pork curing". *LWT – Food Science and Technology*, 35(8), P. 748–756. DOI: <https://doi.org.1006/fstl.2002.0939>
16. Rahman, N. H. A., Lee, M. H., Suhartono, S., Latif, M. T. (2014), "Artificial neural networks and fuzzy time series forecasting: An application to air quality". *Quality & Quantity*, 49 (6), P. 2633–2647. DOI: <https://doi.org.1007/s11135-014-0132-6>
17. Schweidtmann, A. M., Rittig, J. G., König, A., Grohe, M., Mitsos, A., (2020), "Dahmen M. Graph neural networks for prediction of fuel ignition quality". *Energy & Fuels*, 34 (9), P. 11395–11407. DOI: <https://doi.org.1021/acs.energyfuels.0c01533>
18. Tan, C. H., Yusof, K. M., Alwi, S. R. W. (2022), "Quality prediction for polypropylene extrusion based on neural networks". *IOP Conference Series: Materials Science and Engineering*, 1257 (1), P. 1–6. DOI: [doi:10.1088/1757-899X/1257/1/012034](https://doi.org.10.1088/1757-899X/1257/1/012034)
19. Tan, J. S. (2022), "Ablation study on feature group importance for automated essay scoring". *Asia-Pacific Journal of Information Technology and Multimedia*, 11 (01), P. 90–101. DOI: <https://doi.org.17576/apjitm-2022-1101-08>
20. Thwin, M. M. T., Quah, T. S. (2005), "Application of neural networks for software quality prediction using object-oriented metrics". *Journal of Systems and Software*, 76 (2), 2005. P. 147–156. DOI: <https://doi.org.1016/j.jss.2004.05.001>
21. Wang, S., Zhang, Y. (2023), "Grad-CAM: Understanding AI models". *Computers, Materials & Continua*, 76 (2), P. 1321–1324. DOI: <https://doi.org.32604/cmc.2023.041419>

Відомості про авторів / About the Authors

Шантир Антон Сергійович – кандидат технічних наук, Державний університет інформаційно-комунікаційних технологій, доцент кафедри штучного інтелекту, Київ, Україна; e-mail: anton.shantyr@gmail.com; ORCID ID: <http://orcid.org/0000-0002-0466-3659>

Зінченко Ольга Валеріївна – доктор технічних наук, доцент, Державний університет інформаційно-комунікаційних технологій, завідувач кафедри штучного інтелекту, Київ, Україна; e-mail: zinchenkoo@duikt.edu.ua; ORCID ID: <http://orcid.org/0000-0002-3973-7814>

Shantyr Anton – PhD (Engineering Sciences), State University of Information and Communication Technologies, Associate Professor at the Department of Artificial Intelligence, Kyiv, Ukraine.

Zinchenko Olha – Doctor of Sciences (Engineering), Associate Professor, State University of Information and Communication Technologies, Head at the Department of Artificial Intelligence, Kyiv, Ukraine.

FORECASTING SOFTWARE SYSTEM QUALITY METRICS USING MODIFICATIONS OF THE DEEPLIFT INTERPRETATION METHOD

The subject of the study is to improve the interpretative method of *DeepLIFT* (Deep Learning Important Features) in the context of ensuring quality assessment of software systems (PS). The algorithmic and mathematical aspects of the *DeepLIFT* base **method**, as well as its improvement for analyzing and improving the quality of PS, are being studied. **Purpose:** development and justification of five methods of optimization modifications of the *DeepLIFT* method to improve the accuracy and efficiency of forecasting the quality of PS quality. The article solves the following tasks: to conduct a detailed review of problems related to the use of *DeepLIFT* when assessing the quality of the aircraft; provide a mathematical description of the five ways of modifying the *DeepLIFT* method aimed at improving the accuracy, adaptability and speed of quality assessment; experimental testing of the proposed modifications is performed to evaluate their effectiveness in assessing the quality of aircraft compared to the original *DeepLIFT* method. **Research methods:** analysis of literature; methods of experimental verification; calculation of average values and use of significance tests; modeling. The results were achieved: 1) a detailed analysis of the limitations of the base method of *DeepLIFT* in the context of modern software systems, which revealed low adaptability to dynamic data, limited interpretability for complex architectures and difficulty with stability of results in cases of variable operating conditions; 2) five *DeepLIFT* method is proposed; 3) the proposed modifications have undergone experimental testing, which demonstrated their effectiveness compared to the original *DeepLIFT* method. The results of the study showed that all modifications allow you to achieve improvements in key parameters. **Conclusions:** in the face of increasing the complexity of architectures, increasing the volume of data and productivity requirements, innovative methods of evaluation and predicting the quality of PS become a necessary. Each of the improved methods increases accuracy and interpretability compared to the base *DeepLIFT*. The highest accuracy is shown by the temporal *DeepLIFT* (*T-Dlift*) respectively.

Keywords: Modeling; DeepLIFT; innovations; quality assessment; neural networks; optimization.

Бібліографічні описи / Bibliographic descriptions

Шантир А. С., Зінченко О. В. Прогнозування показників якості програмних систем із застосуванням модифікацій інтерпретаційного методу *DeepLIFT*. *Сучасний стан наукових досліджень та технологій в промисловості*. 2024. № 4 (30). С. 110–124. DOI: <https://doi.org/10.30837/2522-9818.2024.4.110>

Shantyr, A., Zinchenko, O. (2024), "Forecasting software system quality metrics using modifications of the DeepLIFT interpretation method", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (30), P. 110–124. DOI: <https://doi.org/10.30837/2522-9818.2024.4.110>