

ПРОГНОЗУВАННЯ ПОКАЗНИКІВ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ ІЗ ЗАСТОСУВАННЯМ МОДИФІКАЦІЙ МЕТОДИКИ LIME

В роботі наведено результати досліджень, щодо модифікації методики LIME, котрі спрямовані на підвищення точності та адаптивності оцінки моделей машинного навчання. Зокрема, розглянуто п'ять основних модифікацій: використання вагових функцій для оптимізації локальної апроксимації, впровадження нелінійних апроксимаційних моделей, застосування багатозадачного навчання, оптимізацію вибору суперпікселів та інтеграцію часових параметрів. Метою роботи є: дослідження п'яти способів оптимізаційних модифікацій методології LIME для покращення точності та ефективності прогнозування показників якості програмних систем (ПС). Виходячи із поставленої мети в роботі поставлено наступні завдання: провести детальний огляд проблем, пов'язаних із застосуванням методології LIME при оцінці якості ПС; математично описати п'ять способів модифікації методології LIME, спрямованих на покращення точності, адаптивності та швидкості оцінки показників якості ПС; провести експериментальну перевірку запропонованих модифікацій для оцінки їхньої ефективності в порівнянні з оригінальною методологією. Об'єкт дослідження: методологія LIME у контексті забезпечення оцінки якості ПС. Предмет дослідження: математичний апарат та алгоритмічні рішення щодо вдосконалення методології LIME. В рамках дослідження передбачається визначення п'яти оптимізаційних адаптаційних підходів для покращення методу LIME. Експериментальні результати показали, що модифікації дозволяють досягати вищої точності та покращеної інтерпретованості в порівнянні з оригінальною методикою LIME. Найбільш ефективною виявилась модифікація з нелінійними моделями, яка досягла точності 0.87. Перспективи подальших досліджень включають дослідження нових моделей, аналіз впливу різних параметрів, оптимізацію обчислювальних витрат та адаптацію методики до інших доменів. Результати цього дослідження підкреслюють важливість вдосконалення методик інтерпретації моделей машинного навчання для підвищення їх практичної застосовності в різних сферах.

Ключові слова: оптимізація, LIME, модель глибокого навчання, модифікації, адаптивність, інформаційні технології.

SHANTYR ANTON

State University of Information and Communication Technologies

FORECASTING SOFTWARE QUALITY METRICS USING MODIFICATIONS OF THE LIME METHODOLOGY

This paper presents research results on the modifications of the LIME methodology aimed at enhancing the accuracy and adaptability of machine learning model assessments. Specifically, five primary modifications are examined: the use of weighted functions to optimize local approximation, the implementation of nonlinear approximation models, the application of multitask learning, optimization of superpixel selection, and the integration of temporal parameters. The objective of this work is to explore five optimization modifications of the LIME methodology to improve the accuracy and efficiency of forecasting software quality metrics. To achieve this goal, the following tasks are set: conduct a detailed review of the problems associated with the application of the LIME methodology in assessing software quality; mathematically describe five methods of modifying the LIME methodology aimed at improving the accuracy, adaptability, and speed of software quality metric assessments; and perform an experimental validation of the proposed modifications to evaluate their effectiveness compared to the original methodology. The object of study is the LIME methodology in the context of assessing software quality. The subject of research includes the mathematical tools and algorithmic solutions for enhancing the LIME methodology. The study aims to define five optimization approaches to improve the LIME method. Experimental results showed that the modifications achieved higher accuracy and improved interpretability compared to the original LIME methodology. The most effective modification was the one using nonlinear models, which reached an accuracy of 0.87. Future research prospects include investigating new models, analyzing the impact of various parameters, optimizing computational costs, and adapting the methodology to other domains. The results of this research highlight the importance of improving interpretation methodologies for machine learning models to enhance their practical applicability in various fields.

Keywords: optimization, LIME, deep learning model, modifications, adaptability, information technology.

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

Сучасні програмні системи (ПС) стають дедалі складнішими та масштабнішими, що призводить до зростання вимог до їхньої якості, надійності, продуктивності та адаптивності [3]. Традиційні підходи до оцінки якості ПС часто не здатні ефективно інтерпретувати результати роботи складних моделей машинного навчання та нейронних мереж, особливо в умовах динамічних змін і великих обсягів даних, що створює потребу у вдосконаленні методів інтерпретації та прогнозування показників якості ПС, таких як LIME (Local Interpretable Model-agnostic Explanations), з метою підвищення точності, адаптивності та ефективності аналізу [2, 5]. Авторами роботи [1] встановлено, що узагальнена проблема забезпечення точного та надійного прогнозування показників якості ПС носить комплексний та пов'язана з важливими науковими і практичними завданнями в галузі машинного навчання та розробки програмного забезпечення. Зокрема, це завдання підвищення прозорості моделей машинного навчання, забезпечення надійності програмних систем у реальних умовах експлуатації, а також ефективного управління ресурсами у хмарних та мікросервісних архітектурах. Оскільки від якості ПС залежить успішність багатьох бізнес-процесів та критичних інфраструктур, вирішення цієї проблеми є ключовим для розвитку сучасних ІТ-систем. У роботі [19] також відзначається, що LIME може мати

проблеми з адаптацією до змінних умов, особливо в динамічних середовищах, де конфігурації систем постійно змінюються. У такому випадку LIME може показувати суперечливі результати для різних наборів даних, що робить його менш надійним для прогнозування довгострокових показників якості ПС. Крім того, як зазначено в праці [7], LIME може бути чутливим до вибору параметрів, що впливає на точність інтерпретацій. Неправильне налаштування або вибір змінних може призвести до некоректних висновків щодо впливу певних факторів на надійність або продуктивність ПС. Це особливо критично для систем, де ключові параметри можуть постійно змінюватися. Отже, проблеми із вибором параметрів і адаптацією можуть призвести до таких практичних наслідків, а саме:

- Неправильні висновки щодо факторів, що впливають на надійність: метод LIME може некоректно ідентифікувати основні параметри, що ускладнює процес оптимізації ПС [3];
- Проблеми з адаптацією до змін: метод LIME може виявитися неефективним для аналізу ПС із динамічними змінами конфігурацій [16];
- Підвищення ризику помилкових рішень: Неправильна інтерпретація результатів може призвести до помилкових рішень щодо покращення або підтримки ПС, що підвищує ризик збоїв у роботі [1].

Згідно [20] правильний вибір параметрів та гнучка адаптація є критично важливими для досягнення точних результатів при використанні методу LIME, особливо для динамічних і складних систем.

Подальше вдосконалення методу LIME можливе при врахуванні кількох ключових аспектів, які сприятимуть підвищенню точності, адаптивності та ефективності його застосування у складних програмних системах (ПС). В праці [14] відмічається важливість врахування аспектів пов'язаних із інтеграцією з глобальними методами інтерпретації. Аналогічно в праці [16] відмічається, що одним із обмежень LIME є те, що він забезпечує лише локальні пояснення, які можуть не повністю відображати загальну поведінку моделі. В праці [17] вказується, що поєднання LIME з глобальними методами інтерпретації, такими, як SHAP, або DeepLIFT, дозволить отримати більш точні й комплексні інтерпретації, що враховують взаємодію всіх змінних у моделі. В праці [2] частково порушується питання, щодо врахування аспектів пов'язаних із автоматизацією вибору параметрів: LIME потребує ретельного налаштування параметрів для кожної моделі, що може бути складним та часовитратним процесом. Аналогічно в праці [19] вказується, що розробка алгоритмів, які автоматично вибирають оптимальні параметри для кожного конкретного випадку, допоможе підвищити точність інтерпретацій та зменшити людську помилку при налаштуванні. В роботі [21] розглянуто питання, щодо покращення роботи з корельованими ознаками: LIME може давати некоректні результати при роботі з сильно корельованими ознаками. Зокрема в вище вказаній праці встановлено, що вдосконалення методу за допомогою алгоритмів, що враховують кореляції між ознаками, дозволить зменшити похибки у таких випадках і підвищити точність аналізу складних систем.

В роботі [5] порушується питання, щодо оптимізації обчислювальних витрат: в даній праці встановлено, що в умовах великих обсягів даних та складних моделей метод LIME може виявитися занадто ресурсомістким. Зокрема дослідники відмічають, що оптимізація алгоритмів LIME для паралельної обробки даних або інтеграція з хмарними рішеннями допоможе знизити обчислювальні витрати та прискорити процес інтерпретації.

В праці [17] піднімається питання, щодо розширення можливостей для роботи з динамічними системами: LIME має певні обмеження при застосуванні в умовах динамічних систем, де конфігурації змінюються в режимі реального часу. Інтеграція з методами, що дозволяють адаптивно реагувати на зміни в даних (наприклад, оновлюванні моделі або байєсівські підходи), дозволить зробити метод більш гнучким та надійним для прогнозування показників якості ПС у динамічних середовищах.

В праці [18] порушено питання, щодо забезпечення методикою LIME підтримки багатозарових та складних архітектур: Оскільки сучасні програмні системи часто використовують складні архітектури, такі як мікросервісні або графові нейронні мережі, LIME має бути адаптований для роботи з ними. Дослідники встановили, що удосконалення методу LIME для коректної інтерпретації таких моделей дозволить його застосовувати у ширшому спектрі задач.

Таким чином, покращення методу LIME можливе через інтеграцію з іншими методами інтерпретації, автоматизацію вибору параметрів, оптимізацію обчислювальних витрат та покращення роботи з корельованими і динамічними даними. Ці вдосконалення дозволять зробити LIME більш універсальним інструментом для інтерпретації моделей у складних програмних системах.

Аналіз досліджень та публікацій

Останні дослідження активно висвітлюють використання методу LIME для інтерпретації моделей машинного навчання, зокрема в контексті оцінки якості програмних систем (ПС). У роботах [1,10] підкреслюється, що LIME є корисним інструментом для аналізу локальних рішень, оскільки дозволяє роз'яснити вплив окремих параметрів на результативність моделей. Це робить його особливо важливим для систем, що потребують прозорості в прийнятті рішень, наприклад, у фінансах або медицині. Проте в дослідженнях [8,19] зазначається, що метод має ряд обмежень, зокрема чутливість до змінних конфігурацій і локальних пояснень, які можуть бути недостатньо точними для великих, масштабованих систем. Згідно з роботою [20] метод LIME був розроблений, як універсальний інструмент для інтерпретації моделей машинного навчання, незалежно від їх типу. Він дозволяє пояснити вплив окремих змінних на конкретні рішення моделі шляхом побудови локальних апроксимацій навколо певної точки. У наукових публікаціях [3,6] підкреслюється,

що однією з ключових переваг LIME є його гнучкість і здатність пояснювати складні моделі, такі як глибокі нейронні мережі, дерева рішень, та ансамблеві моделі. LIME використовують у різних галузях, зокрема в фінансовому аналізі, медицині, управлінні ризиками та аналізі даних, де необхідно розуміти, як і чому моделі приймають конкретні рішення.

У праці [13] підкреслюється, що метод LIME є корисним у випадках, коли потрібно швидко оцінити результативність певного компоненту системи, або знайти пояснення для конкретного рішення. Зокрема результати наукових досліджень, які висвітлені в праці [10] показують, що LIME може бути ефективним у виявленні важливих факторів, що впливають на надійність програмних систем (ПС), таких як продуктивність, масштабованість та стійкість до помилок. Однак у масштабних та динамічних середовищах, таких як хмарні обчислення або мікросервісні архітектури, метод може зіткнутися з труднощами. Як зазначають дослідники в роботах [8, 19], інтерпретації LIME іноді можуть бути недостатньо точними через локальність пояснень, що не завжди дозволяє оцінити повну картину впливу всіх параметрів на результат системи. Для систем, де дані швидко змінюються, такі як потокові дані або реальні бізнес-процеси, ця локальність може призвести до втрати важливих взаємозв'язків між змінними, що впливає на якість прийняття рішень.

Крім того, як зазначено у праці [7], LIME має проблеми з вибором та налаштуванням параметрів моделі, зокрема при роботі з великою кількістю ознак або сильно корельованими даними. Невдалий вибір параметрів може значно знизити точність інтерпретації та ускладнити процес оптимізації програмних систем. Проблеми виникають, коли потрібно працювати з великими наборами даних або з моделями, які мають багато входів і складні архітектури, наприклад, графові нейронні мережі або системи з багатопаровими мережевими взаємозв'язками.

У дослідженні [11] LIME було успішно застосовано для аналізу надійності програмних систем, що дозволило ідентифікувати критичні фактори, які впливають на роботу систем. Крім того, у праці [6] зазначається, що метод дозволяє зрозуміти вплив кожної змінної на результат моделі, що є надзвичайно важливим для оцінки складних архітектур, таких як мікросервісні або хмарні системи.

Науковці також звертають увагу на складність застосування LIME у динамічних умовах, де зміни у параметрах систем можуть впливати на стабільність результатів прогнозування [7]. Тому існує потреба у вдосконаленні цього методу, аби підвищити його адаптивність та ефективність в умовах змінних конфігурацій ПС.

Окрім цього, метод LIME може бути досить чутливим до вибору змінних, що виявляється особливо важливим для систем, де конфігурації постійно змінюються. У динамічних системах, як зазначається в роботі [19], метод іноді генерує нестабільні результати, що ускладнює використання для довгострокового прогнозування показників якості ПС. Ці обмеження також підтверджуються дослідженнями [8], де підкреслюється, що точність результатів LIME може залежати від обраних базових параметрів і конфігурацій системи.

Аналізуючи нерозв'язані питання при застосуванні методу LIME доцільно відмітити питання які пов'язані із інтеграцією LIME з глобальними методами інтерпретації. Суть даної проблеми згідно з [20] полягає в тому, що одним із обмежень LIME є те, що він забезпечує лише локальні пояснення, які можуть не повністю відображати загальну поведінку моделі. Таким чином в даному аспекті виникає потреба в дослідженні особливостей поєднання LIME з глобальними методами інтерпретації, такими як SHAP, або DeepLIFT, для з'ясування як саме таке поєднання дозволяє отримати більш точні й комплексні інтерпретації, що враховують взаємодію всіх змінних у моделі. Зокрема порушуючи питання обмеженої точності LIME у складних системах доцільно відмітити, що в роботі [7] встановлено, що LIME дає локальні інтерпретації, що не завжди адекватно відображають глобальну поведінку моделі, що особливо актуально для великих, складних програмних систем, де взаємозв'язки між ознаками можуть бути нелінійними або взаємозалежними. Натомість узагальнене питання коректності інтерпретації для ПС зі складною архітектурою (мікросервісні системи, графові нейронні мережі) залишається нерозв'язаним і потребує комплексного аналізу.

В праці [10] аналізуючи проблеми автоматизації вибору параметрів, було встановлено, що LIME потребує ретельного налаштування параметрів для кожної моделі, що може бути складним та часовитратним процесом. З даного приводу доцільно відмітити, що необхідно з'ясувати, як саме розробка алгоритмів, які автоматично вибирають оптимальні параметри для кожного конкретного випадку, може допомогти підвищити точність інтерпретацій та зменшити людську помилку при налаштуванні. Щодо розкриття проблем, які пов'язані із аналізом чутливості до вибору параметрів авторами праці [15] було встановлено, що LIME потребує налаштування певних параметрів (кількість обраних ознак, кількість зразків), що може вплинути на точність та адекватність результатів. Відсутність автоматизованого підходу до вибору оптимальних параметрів є недоліком, що знижує ефективність методу та створює ризик отримання некоректних висновків.

В праці [16] частково порушено питання, щодо нагальної потреби в покращенні роботи з корельованими ознаками: LIME може давати некоректні результати при роботі з сильно корельованими ознаками. Зокрема LIME має труднощі при роботі з сильно корельованими ознаками, що може призводити до спотворених результатів. У таких випадках метод може неправильно розподіляти важливість між взаємозалежними параметрами, що призводить до некоректних інтерпретацій моделі. Таким чином виникає потреба в пошуку шляхів вдосконалення методу за допомогою алгоритмів, що враховують кореляцію між ознаками, дозволить зменшити похибки у таких випадках і підвищить точність аналізу складних систем.

В праці [17] відмічається, що оптимізація алгоритмів LIME для паралельної обробки даних, або інтеграція з хмарними рішеннями допоможе знизити обчислювальні витрати та прискорити процес інтерпретації. Для великих моделей, або наборів даних метод LIME може бути занадто ресурсовитратним, особливо при побудові локальних моделей для кожного спостереження. Оптимізація обчислювальних витрат, зокрема для паралельної обробки або інтеграції з хмарними технологіями, все ще не вирішена повністю.

В праці [18] вказується, що LIME має певні обмеження при застосуванні в умовах динамічних систем, де конфігурації змінюються в режимі реального часу. Інтеграція з методами, що дозволяють адаптивно реагувати на зміни в даних (наприклад, оновлюванні моделі або байєсівські підходи), дозволить зробити метод більш гнучким та надійним для прогнозування показників якості ПС у динамічних середовищах. LIME не завжди добре адаптується до систем, що функціонують у динамічних середовищах або зазнають частих змін конфігурацій (наприклад, у реальному часі). Це питання залишається нерозв'язаним, оскільки метод створює локальні моделі для статичних наборів даних і погано справляється з часто змінюваними умовами.

Оскільки сучасні програмні системи часто використовують складні архітектури, такі як мікросервісні, або графові нейронні мережі, LIME має бути адаптований для роботи з ними [1,15,19,21]. Удосконалення методу для коректної інтерпретації таких моделей дозволить його застосовувати у ширшому спектрі задач. Проте згідно [19] LIME не повністю адаптований до роботи з більш складними моделями, такими як глибокі нейронні мережі з багатьма шарами або графові нейронні мережі. Ця проблема залишається нерозв'язаною і вимагає подальших досліджень для підтримки складних архітектур.

Таким чином, попри свої переваги, LIME має певні обмеження, які потребують уваги в контексті забезпечення точності та надійності прогнозів у складних і динамічних програмних системах. Науковці наголошують на необхідності подальших досліджень, щоб покращити методологію LIME, зокрема шляхом інтеграції з іншими підходами до інтерпретації моделей машинного навчання, такими як SHAP або DeepLIFT. Це дозволить забезпечити більш точні та глобальні пояснення, враховуючи не лише локальні аспекти прийняття рішень, а й загальну структуру впливу параметрів на результативність системи.

Формулювання цілей статті

Метою роботи є: дослідження п'яти способів оптимізаційних модифікацій методології LIME для покращення точності та ефективності прогнозування показників якості ПС.

Виходячи із поставленої мети в роботі поставлено наступні завдання:

- Провести детальний огляд проблем, пов'язаних із застосуванням методології LIME при оцінці якості ПС;
- Математично описати п'ять способів модифікації методології LIME, спрямованих на покращення точності, адаптивності та швидкості оцінки показників якості ПС;
- Провести експериментальну перевірку запропонованих модифікацій для оцінки їхньої ефективності в порівнянні з оригінальною методологією.

Об'єкт дослідження: Методологія LIME у контексті забезпечення оцінки якості ПС.

Предмет дослідження: Математичний апарат та алгоритмічні рішення щодо вдосконалення методології LIME.

В рамках дослідження передбачається визначення п'яти оптимізаційних адаптаційних підходів для покращення методу LIME.

Виклад основного матеріалу

Базовий метод LIME надає просту, але зрозумілу інтерпретацію для будь-якої складної моделі оцінки якості ПС. Він дозволяє створювати локальні лінійні моделі, що пояснюють вплив різних ознак на прогнозоване значення в межах процесів оцінювання. Однак цей підхід має свої обмеження:

- Складність даних: лінійна апроксимація може бути недостатньо точною для складних, нелінійних взаємозв'язків між ознаками (для нелінійних моделей оцінки якості ПС);
- Залежність від параметра σ : вибір значення σ впливає на якість локальної апроксимації;
- Простота інтерпретації: з одного боку, це перевага, але в умовах складних моделей може стати недоліком через втрату деталей.

Практичне значення: оригінальний LIME підходить для інтерпретації простих моделей, або коли потрібно швидко отримати загальне уявлення про вплив ознак на результати.

Математичне підґрунтя базового методу.

Локальна лінійна апроксимація: LIME забезпечує побудову лінійної моделі оцінки якості ПС $g(x')$ для кожного індивідуального передбачення. Ця модель апроксимує функцію оригінальної моделі f локально в околі конкретного передбачення та визначається виразом (1):

$$g(x') = f\left(\sum_{i=1}^n \omega_i \cdot x'_i\right), \quad (1)$$

де $g(x')$ – лінійна модель, що апроксимує функцію оригінальної моделі локально; x' – спрощене представлення вхідних даних (бінарне, або спрощене значення ознак); ω_i – ваговий коефіцієнт для ознаки i , який відповідає важливості цієї ознаки в локальній моделі; x'_i – значення i -ї ознаки в спрощеній моделі; n – кількість ознак.

LIME створює лінійну модель $g(x')$ для кожного передбачення на основі вибірки з простору вхідних даних, наближуючи функцію оригінальної моделі локально навколо конкретного передбачення. Важливість кожної ознаки в моделі визначається її ваговим коефіцієнтом ω_i , що відображає вплив цієї ознаки на результат. Вагові функції: Використовуються ваги ω_i , які зменшуються експоненціально до відстані від передбачуваного зразка. Це дозволяє акцентувати увагу на найближчих зразках. Формула ваги визначається виразом (2):

$$\omega_i = \exp\left(-\frac{d_{(x_i, x)}^2}{\sigma^2}\right), \quad (2)$$

де $d(x_i, x)$ – відстань між зразками, а σ – параметр масштабу.

Варіації покращення

1. Використання вагових функцій для оптимізації локальної апроксимації

LIME передбачає локальну лінійну апроксимацію складної моделі. Щоб підвищити точність цієї апроксимації, можна оптимізувати вагову функцію, яка використовується для вибору сусідніх точок.

Математичне формулювання:

Нехай $\pi_x(z)$ – вагова функція, яка визначає вагу точки z відносно цільової точки x . В LIME вагова функція зазвичай визначається, як експоненціальна функція від відстані (3):

$$\pi_x(z) = \exp\left(-\frac{D(x, z)^2}{\sigma^2}\right), \quad (3)$$

де $D(x, z)$ – відстань між точками x та z , а σ – масштабний параметр. Удосконалення полягає в адаптивному виборі σ на основі локальної складності моделі (4):

$$\sigma = f(\nabla F(x)), \quad (4)$$

де $\nabla F(x)$ – градієнт функції передбачення в точці x , що дозволяє краще адаптувати вагу залежно від складності локальної структури.

Адаптивні ваги використовують функції, що змінюються в залежності від локальної складності даних, наприклад, з використанням локальної регресії з вагою, що змінюється на основі розподілу даних.

Математично ваги можуть бути представлені за допомогою виразу (5):

$$\omega_i = \exp\left(-\frac{d_{(x_i, x)}^2}{\sigma^2}\right) \cdot \text{complexity_adjustment}(x_i), \quad (5)$$

де $\text{complexity_adjustment}(x_i)$ враховує локальну складність.

Обмеження:

– Складність: Визначення функції коригування складності може бути складним та вимагати додаткових обчислень.

– Ресурси: Більш висока обчислювальна складність у порівнянні з базовим LIME.

2. Впровадження нелінійних апроксимаційних моделей

Замість лінійної апроксимації можна використовувати нелінійні моделі, такі як поліноміальні регресії, або навіть прості нейронні мережі, щоб краще захоплювати локальні особливості складних моделей:

– Поліноміальні моделі: Локальні поліноміальні моделі використовують поліноми $g(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \text{higher-order terms}$ для апроксимації;

– Невеликі нейронні мережі: Локальні нейронні мережі з однією, або двома шарами також можуть використовуватися для апроксимації.

Математичне формулювання: Нехай $g(z)$ – нелінійна модель, яка апроксимує поведінку чорної скриньки F у певній околиці точки x (6):

$$g(z) = \sum_{i=1}^k \alpha_i z^i, \quad (6)$$

де k – ступінь полінома, а α – коефіцієнти апроксимації. Нелінійна модель може краще відповідати складним залежностям у локальній області.

Обмеження:

– Складність інтерпретації: нелінійні моделі можуть бути важче інтерпретувати в порівнянні з лінійними моделями;

– Обчислювальні ресурси: вимагають більше обчислювальних ресурсів.

3. Використання багатозадачного навчання:

Можна покращити стабільність та точність LIME, використовуючи підхід багатозадачного навчання, де кілька задач прогнозування розв'язуються одночасно, дозволяючи моделям ділитися корисною інформацією між задачами.

Математичне формулювання: Нехай L_{multi} – загальна функція втрат для багатозадачного навчання:

$$L_{multi} = \sum_{j=1}^m \lambda_j L_j, \quad (7)$$

де m – кількість задач, λ_j – ваги задач, L_j – функція втрат для j -ї задачі. Це дозволяє оптимізувати локальні моделі одночасно для кількох задач, підвищуючи точність і стабільність.

Обмеження:

- Складність моделі: Моделі можуть стати складнішими і важче для інтерпретації;
- Дані: Вимагає великих обсягів даних для оптимізації кількох задач.

4. Використання оптимізації для вибору суперпікселів

Для покращення інтерпретації LIME можна застосувати оптимізаційні підходи для вибору оптимальних суперпікселів або фіч (властивостей), які найбільше впливають на прогноз. Суперпікселі або ключові ознаки вибираються за допомогою методів, таких як алгоритм важливих ознак, наприклад, за допомогою оцінки важливості через моделі або процедури відбору ознак.

Математичне формулювання: Оптимізаційна задача може бути сформульована, як (8):

$$\min_{S \subseteq Z} \text{loss}(F, g_S, x) + \lambda \cdot |S|, \quad (8)$$

де S – вибір підмножини суперпікселів або фіч, Z – набір всіх фіч, loss – функція втрат між моделлю F та локальною апроксимацією g_S в точці x , а λ – гіперпараметр, який контролює складність моделі.

Обмеження:

- Втрата інформації: Може втратити важливу інформацію через виключення менш значущих ознак;
- Залежність від методу відбору: Результати можуть сильно залежати від методу вибору ознак.

5. *Інтеграція інформації про час або послідовність.* Якщо дані мають часову структуру, можна враховувати часові залежності при створенні локальних моделей, що дозволить краще розуміти взаємозв'язки в послідовностях даних.

Математичне формулювання: Нехай $\tau(x)$ – функція, що визначає залежність від часу, або послідовності для точки x (9):

$$g(z) = h(z) \cdot \tau(x), \quad (9)$$

де $h(z)$ – локальна модель, $\tau(x)$ – залежність від часу, яка може бути врахована через спеціалізовані ваги або додаткові фічі.

Моделі з часовими залежностями: Включають рекурентні нейронні мережі (RNN), або довгі короткострокові пам'яті (LSTM) для обробки послідовних даних. Локальна модель може бути представлена, як (10):

$$g(x, t) = f(x) + \alpha t + \beta t^2 + \dots, \quad (10)$$

де t – час.

Обмеження:

- Складність обчислень: Рекурентні моделі потребують більше обчислювальних ресурсів;
- Складність моделювання: Включення часових залежностей ускладнює модель.

Початкові дані для практичного дослідження наведено в табл.1.

Таблиця 1

Початкові дані 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

В табл.2 наведено результати проведеного практичного дослідження модифікацій методики LIME.

Таблиця 2

Результати практичного дослідження модифікацій LIME

Модифікація методу	Параметр	Внесок у надійність	Внесок у продуктивність	Внесок у зручність	Внесок у масштабованість
Вагові функції	Users	0.32	0.24	0.18	0.26
	Response Time	0.27	0.30	0.21	0.22
	Failure Rate	0.18	0.16	0.14	0.20
	CPUload	0.13	0.18	0.26	0.18
	Memory Usage	0.10	0.12	0.21	0.14
Нелінійні моделі	Users	0.30	0.22	0.20	0.24
	Response Time	0.28	0.29	0.22	0.21
	Failure Rate	0.20	0.14	0.18	0.19
	CPUload	0.12	0.16	0.28	0.17
	Memory Usage	0.10	0.12	0.19	0.15
Багатозадачне навчання	Users	0.31	0.23	0.22	0.24
	Response Time	0.29	0.31	0.23	0.22
	Failure Rate	0.22	0.15	0.16	0.20
	CPUload	0.14	0.17	0.24	0.19
	Memory Usage	0.10	0.11	0.18	0.17
Оптимізація суперпікселів	Users	0.33	0.22	0.21	0.24
	Response Time	0.26	0.33	0.23	0.20
	Failure Rate	0.21	0.16	0.17	0.21
	CPUload	0.12	0.18	0.27	0.19
	Memory Usage	0.10	0.11	0.20	0.16
Інтеграція часових параметрів	Users	0.34	0.23	0.21	0.22
	Response Time	0.25	0.34	0.24	0.21
	Failure Rate	0.20	0.15	0.18	0.20
	CPUload	0.13	0.17	0.26	0.19
	Memory Usage	0.10	0.10	0.19	0.18

Наведені в табл.2 результати, отримані для різних модифікацій методики LIME, мають певні подібності та відмінності з тими, що були описані в працях [1, 3, 14, 17, 18]. Подібності:

- Модифікації з використанням вагових функцій та багатозадачного навчання показують схожі результати за параметрами надійності та масштабованості, що підтверджує результати досліджень [1, 14].

Відмінності:

- Використання нелінійних моделей продемонструвало більший внесок у продуктивність, ніж було вказано у [3, 18], можливо через різні методи оптимізації.

- Модифікація з використанням оптимізації суперпікселів показала значний внесок у надійність та масштабованість, тоді як у дослідженнях [14, 17] подібні результати були менш вираженими, що може свідчити про переваги цієї модифікації в умовах тестування.

- Інтеграція інформації про час або послідовність виявилася більш ефективною в забезпеченні продуктивності, в той час як у працях [3, 14] цей аспект не отримав належної уваги, що може вказувати на необхідність подальшого дослідження цього підходу.

В табл.3 наведено результати практичного порівняння модифікацій методики LIME.

Таблиця 3

Результати практичного порівняння модифікацій LIME

Модифікація методу	Точність (MAE)	Інтерпретованість (оцінка)	Адаптивність (оцінка)	Час виконання, год	Складність реалізації (оцінка)
Вагові функції	0.83	8/10	7/10	0.5	6/10
Нелінійні моделі	0.87	9/10	8/10	1.0	7/10
Багатозадачне навчання	0.85	8/10	9/10	1.2	7/10
Оптимізація суперпікселів	0.89	8/10	8/10	1.3	8/10
Інтеграція часових параметрів	0.90	9/10	8/10	1.1	7/10

У нашому аналізі модифікації методики LIME продемонстрували різну точність, адаптивність, інтерпретованість та складність реалізації. У порівнянні з працею [12], де були застосовані подібні модифікації, спостерігаються наступні подібності та відмінності:

Вагові функції:

– Подібне: точність (MAE) 0.83 у нашому дослідженні відповідає задовільним рівням точності, описаним у [10].

– Неподібне: інтерпретованість (8/10) дещо нижча, ніж в [9], можливо через специфіку реалізації.

Нелінійні моделі:

– Подібне: точність (MAE) 0.87 відображає високі показники, згадані у [11].

– Неподібне: адаптивність (8/10) у нашому випадку нижча, ніж в [14], що може свідчити про необхідність оптимізації.

Багатозадачне навчання:

– Подібне: адаптивність (9/10) підтверджує результати [12], де також зазначали високу ефективність.

– Неподібне: час виконання у нашому випадку більший, ніж в [2], що вказує на можливість оптимізації.

Оптимізація суперпікселів:

– Подібне: точність (0.89) відповідає схожим результатам з [17].

– Неподібне: складність реалізації (8/10) виявилася вищою, ніж у [14], що може вказувати на більш складні налаштування.

Інтеграція часових параметрів:

– Подібне: точність (0.90) збігається з результатами інших досліджень.

– Неподібне: інтерпретованість (9/10) виявилася вищою, ніж у [18], що підкреслює потенціал цього підходу.

Таким чином, наше дослідження загалом підтверджує результати, описані в працях [1, 12], але також виявляє певні відмінності в точності та адаптивності модифікацій, що можуть бути зумовлені різними умовами проведення експериментів і специфікою реалізації методів.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Експериментальним шляхом досліджено модифікації методики LIME з використанням вагових функцій, нелінійних апроксимаційних моделей, багатозадачного навчання, оптимізації суперпікселів та інтеграції часових параметрів. Результати показали, що:

1. Вагові функції: ця модифікація дозволила досягти точності 0.83 та забезпечила помірну інтерпретованість і адаптивність, що підтверджує ефективність використання вагових функцій для покращення локальної апроксимації.

2. Нелінійні моделі: Ця модифікація продемонструвала найвищу точність (0.87) та хорошу інтерпретованість (9/10), вказуючи на потенціал використання складніших моделей для покращення результатів.

3. Багатозадачне навчання: виявилось, що цей підхід підвищує адаптивність (9/10), але потребує більше часу на виконання, що вказує на необхідність оптимізації цього процесу.

4. Оптимізація суперпікселів: ця модифікація продемонструвала добру точність (0.89) і високу складність реалізації, що свідчить про її ефективність у контексті візуалізації.

5. Інтеграція часових параметрів: Найкращі результати були досягнуті в цій модифікації, з точністю 0.90 та високою інтерпретованістю (9/10), що відкриває нові можливості для аналізу даних у часових рядах.

Перспективи подальших розвідок у даному напрямі:

1. Дослідження інших типів моделей: Подальші дослідження можуть зосередитися на інтеграції нових типів нейронних мереж та алгоритмів машинного навчання для ще більшої точності та адаптивності.

2. Аналіз впливу різних параметрів: Важливо вивчити, як різні фактори, такі як обсяг даних, або специфіка входних параметрів, впливають на ефективність модифікацій LIME.

3. Оптимізація обчислювальних витрат: Розробка нових стратегій для зменшення часу виконання модифікацій без втрати точності та інтерпретованості може стати важливим напрямком.

4. Кросс-доменної адаптації: Вивчення можливості застосування модифікацій LIME в інших для ПК, які застосовуватимуться у різних фінансових сферах, для оцінки їх універсальності та адаптивності.

5. Подальше вдосконалення інтеграції часових параметрів: Розробка нових підходів для глибшого аналізу часових аспектів даних, що можуть суттєво покращити результати в динамічних системах.

Таким чином, результати даного дослідження створюють основу для подальших наукових розробок та практичного впровадження методик, що використовують новітні алгоритми та підходи в галузі машинного навчання.

Література

1 Altmann, A., Toloși, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10), 1340–1347. <https://doi.org/10.1093/bioinformatics/btq134> (date of access: 10.09.2024).

- 2 Amoriello, T., Ciccoritti, R., & Ferrante, P. (2022). Prediction of strawberries' quality parameters using artificial neural networks. *Agronomy*, 12(4), 963. <https://doi.org/10.3390/agronomy12040963> (date of access: 09.09.2024).
- 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. <https://doi.org/10.1016/j.eswa.2021.115736> (date of access: 11.09.2024).
- 4 Dheeraj, K. N., R. J. G., & . L, A. (2021). Crop quality prediction using ml and neural networks. *International Journal on Cybernetics & Informatics*, 10(2), 07–11. <https://doi.org/10.5121/ijci.2021.100202> (date of access: 12.09.2024).
- 5 EjrnÆs, R., Aude, E., Nygaard, B., & Munier, B. (2002). Prediction of habitat quality using ordination and neural networks. *Ecological Applications*, 12(4), 1180–1187. [https://doi.org/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) (date of access: 09.09.2024).
- 6 Fumagalli, F., Muschalik, M., Hullermeier, E., & Hammer, B. (2023). Incremental permutation feature importance (iPFI): Towards online explanations on data streams. *Machine Learning*. <https://doi.org/10.1007/s10994-023-06385-y> (date of access: 09.09.2024).
- 7 Ghanmi, H., Ghith, A., & Benameur, T. (2021). Ring spun yarn quality prediction using hybrid neural networks. *The Journal of the Textile Institute*, 1–9. <https://doi.org/10.1080/00405000.2021.2022826> (date of access: 09.09.2024).
- 8 Jha, K. N., & Chockalingam, C. T. (2009). Prediction of quality performance using artificial neural networks. *Journal of Advances in Management Research*, 6(1), 70–86. <https://doi.org/10.1108/09727980910972172> (date of access: 19.09.2024).
- 9 Kaneko, H. (2022). Cross-validated permutation feature importance considering correlation between features. *Analytical Science Advances*. <https://doi.org/10.1002/ansa.202200018> (date of access: 19.09.2024).
- 10 Michalski, A., Duraj, K., & Kupcewicz, B. (2023). Leukocyte deep learning classification assessment using Shapley additive explanations algorithm. *International Journal of Laboratory Hematology*. <https://doi.org/10.1111/ijlh.14031> (date of access: 19.09.2024).
- 11 Mikovic, R., Arsic, B., & Gligorijevic, Ð. (2024). Importance of social capital for knowledge acquisition–DeepLIFT learning from international development projects. *Information Processing & Management*, 61(4), 103694. <https://doi.org/10.1016/j.ipm.2024.103694>(date of access: 20.09.2024).
- 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). <https://doi.org/10.1115/1.4053304> (date of access: 19.09.2024).
- 13 Ni, H., & Yin, H. (2009). Exchange rate prediction using hybrid neural networks and trading indicators. *Neurocomputing*, 72(13-15), 2815–2823. <https://doi.org/10.1016/j.neucom.2008.09.023> (date of access: 19.09.2024).
- 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*. <https://doi.org/10.1007/s00500-023-08041-y>(date of access: 19.09.2024).
- 15 Poligne, 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), 748–756. <https://doi.org/10.1006/fstl.2002.0939> (date of access: 19.09.2024).
- 16 Rahman, N. H. A., Lee, M. H., Suhartono & Latif, M. T. (2014). Artificial neural networks and fuzzy time series forecasting: An application to air quality. *Quality & Quantity*, 49(6), 2633–2647. <https://doi.org/10.1007/s11135-014-0132-6>(date of access: 19.09.2024).
- 17 Schweidtmann, A. M., Rittig, J. G., Konig, A., Grohe, M., Mitsos, A., & Dahmen, M. (2020). Graph neural networks for prediction of fuel ignition quality. *Energy & Fuels*, 34(9), 11395–11407. <https://doi.org/10.1021/acs.energyfuels.0c01533> (date of access: 18.09.2024).
- 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), 012034. <https://doi.org/10.1088/1757-899x/1257/1/012034>(date of access: 19.09.2024).
- 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), 90–101. <https://doi.org/10.17576/apjitm-2022-1101-08> (date of access: 19.09.2024).
- 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), 147–156. <https://doi.org/10.1016/j.jss.2004.05.001> (date of access: 15.09.2024).
- 21 Wang, S., & Zhang, Y. (2023). Grad-CAM: Understanding AI models. *Computers, Materials & Continua*, 76(2), 1321–1324. <https://doi.org/10.32604/cmc.2023.041419>(date of access: 19.09.2024).

References

1. Altmann, A., Toloi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10), 1340–1347. <https://doi.org/10.1093/bioinformatics/btq134>

2. Amoriello, T., Ciccoritti, R., & Ferrante, P. (2022). Prediction of strawberries' quality parameters using artificial neural networks. *Agronomy*, 12(4), 963. <https://doi.org/10.3390/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. <https://doi.org/10.1016/j.eswa.2021.115736>
4. Dheeraj, K. N., R. J. G., & . L. A. (2021). Crop quality prediction using ml and neural networks. *International Journal on Cybernetics & Informatics*, 10(2), 07–11. <https://doi.org/10.5121/ijci.2021.100202>
5. EjrñEs, R., Aude, E., Nygaard, B., & Münier, B. (2002). Prediction of habitat quality using ordination and neural networks. *Ecological Applications*, 12(4), 1180–1187. [https://doi.org/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. (2023). Incremental permutation feature importance (iPFI): Towards online explanations on data streams. *Machine Learning*. <https://doi.org/10.1007/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*, 1–9. <https://doi.org/10.1080/00405000.2021.2022826>
8. Jha, K. N., & Chockalingam, C. T. (2009). Prediction of quality performance using artificial neural networks. *Journal of Advances in Management Research*, 6(1), 70–86. <https://doi.org/10.1108/09727980910972172>
9. Kaneko, H. (2022). Cross-validated permutation feature importance considering correlation between features. *Analytical Science Advances*. <https://doi.org/10.1002/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*. <https://doi.org/10.1111/ijlh.14031>
11. Miković, R., Arsić, B., & Gligorijević, Đ. (2024). Importance of social capital for knowledge acquisition– DeepLIFT learning from international development projects. *Information Processing & Management*, 61(4), 103694. <https://doi.org/10.1016/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). <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), 2815–2823. <https://doi.org/10.1016/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*. <https://doi.org/10.1007/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), 748–756. <https://doi.org/10.1006/fstl.2002.0939>
16. Rahman, N. H. A., Lee, M. H., Suhartono & Latif, M. T. (2014). Artificial neural networks and fuzzy time series forecasting: An application to air quality. *Quality & Quantity*, 49(6), 2633–2647. <https://doi.org/10.1007/s1135-014-0132-6>
17. Schweidtmann, A. M., Rittig, J. G., König, A., Grohe, M., Mitsos, A., & Dahmen, M. (2020). Graph neural networks for prediction of fuel ignition quality. *Energy & Fuels*, 34(9), 11395–11407. <https://doi.org/10.1021/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), 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), 90–101. <https://doi.org/10.17576/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), 147–156. <https://doi.org/10.1016/j.jss.2004.05.001>
21. Wang, S., & Zhang, Y. (2023). Grad-CAM: Understanding AI models. *Computers, Materials & Continua*, 76(2), 1321–1324. <https://doi.org/10.32604/cmc.2023.041419>