

ПРОГНОЗУВАННЯ ПОКАЗНИКІВ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ ІЗ ЗАСТОСУВАННЯМ МОДИФІКАЦІЙ МЕТОДУ SHAP

¹Державний університет інформаційно-комунікаційних технологій, Київ

Запропоновано 5 пропозицій для поліпшення практичного застосування методу SHAP (Shapley additive explanations — додаткові пояснення за Шеплі) у контексті оцінки якості програмних систем (ПС), що дозволило дослідити можливості вдосконалення методу SHAP щодо прогнозування показників якості ПС. Метою дослідження є підвищення точності та адаптивності методу SHAP шляхом його модифікацій для врахування різноманітних параметрів якості ПС, зокрема продуктивності, надійності, масштабованості та зручності використання. В роботі визначено такі завдання: провести детальний огляд проблем, пов'язаних із застосуванням методу SHAP під час оцінки якості ПС; математично описати п'ять модифікацій методу SHAP, спрямованих на підвищення його точності, адаптивності та швидкості оцінки показників якості ПС; провести експериментальну перевірку запропонованих модифікацій для оцінки їхньої ефективності в порівнянні з оригінальним методом. В ході досліджень розглянуто п'ять підходів, щодо вдосконалення методу SHAP: LSTM (Long Short-Term Memory — довга короткочасна пам'ять); CNN (Convolutional Neural Networks — згорткові нейронні мережі), адаптивний SHAP MLP (multilayer perceptron — багатосаровий перцептрон), RNN (Recurrent Neural Network — рекурентна нейронна мережа (розглядалися ансамблі моделей та Баєсівське оновлення)). Проведено практичне порівняння результатів, яке показало, що запропоновані варіації SHAP можуть значно підвищити точність та швидкість прогнозування, особливо в умовах динамічних змін та великих обсягів даних. В процесі розробки та обґрунтування п'яти оптимізаційних модифікацій методу SHAP досягнуто підвищення точності та ефективності прогнозування показників якості ПС. У ході дослідження встановлено, що методи, які використовують глибокі нейронні мережі (LSTM, CNN), показують вищу точність та адаптивність у порівнянні з оригінальним SHAP, проте ціною цього є складність реалізації та тривалий час виконання. Адаптивний SHAP та Ensemble є оптимальними з погляду балансу між точністю, адаптивністю та інтерпретованістю, проте потребують певної оптимізації для поліпшення часу виконання. Оригінальний SHAP показав задовільну точність (MAE 0,84) та інтерпретованість (7/10), проте поступається сучасним підходам у питаннях адаптивності (6/10) та часу виконання (0,4 години). SHAP потребує подальшої оптимізації, зокрема в умовах динамічних середовищ, де важлива швидка адаптація до нових даних. Часові ряди (LSTM) забезпечили найвищу точність серед інших методів (MAE 0,91) та хорошу адаптивність (8/10).

Ключові слова: оптимізація, модель глибокого навчання, Баєсівське оновлення, адаптивність.

Вступ

Аналізуючи спеціальну літературу щодо доцільності застосування методу SHAP у контексті оцінки якості ПС, варто зазначити, що наукові погляди на цю тему різняться, оскільки використання SHAP в цій галузі є відносно новим підходом і залежить від конкретних завдань. Зокрема, у праці [1] підкреслюється, що SHAP є потужним інструментом для інтерпретації та прогнозування якості ПС, оскільки дозволяє точно оцінювати внесок вхідних параметрів у результати роботи моделей. Через це SHAP є корисним для аналізу продуктивності та надійності складних систем, особливо в умовах динамічних навантажень.

У дослідженні [2] SHAP успішно застосовано для оцінки масштабованості великих ПС, що дозволило виявити основні фактори, які впливають на їхню продуктивність. До того, у праці [3] зазначається, що

цей метод дозволяє глибше аналізувати вплив різних вхідних параметрів, що є критично важливим для точних прогнозів, оскільки SHAP виявляє приховані зв'язки між вхідними даними та показниками якості ПС, що сприяє поліпшенню процесу ухвалення рішень під час розробки та тестування.

Проте, незважаючи на переваги, метод SHAP має також низку практичних недоліків. У праці [4] відзначається, що його адаптивність для складних архітектур є обмеженою. Зокрема, варто зазначити, що SHAP може демонструвати недостатню точність в оцінюванні якості складних програмних архітектур, таких як мікросервісні системи, або розподілені системи. У таких випадках реалізація SHAP може бути складною і вимагати глибокої адаптації під специфічні архітектури.

У дослідженні [5] вказується, що метод SHAP не завжди добре адаптується для оцінки продуктивності та стійкості хмарних сервісів з високим рівнем динамічності навантажень. Додатково, у праці [6] наголошується на проблемі надмірної залежності від вибору базових значень. Ця проблема полягає в тому, що метод SHAP дуже залежить від вибраних базових значень, що може бути проблематичним для систем з великою кількістю змінних факторів.

Неправильний вибір базових значень може суттєво знизити точність оцінки якості програмних систем. Зокрема, у праці [7] виявлено, що із застосуванням SHAP для прогнозування надійності ПС з динамічними конфігураціями, вибір базових значень спричинив значні похибки в інтерпретації результатів. Це означає, що використовуючи SHAP для прогнозування надійності систем, існує ризик отримання неточних або хибних результатів через неправильний вибір базових значень.

Правильний вибір базових значень є критично важливим для отримання точних прогнозів і забезпечення високої надійності програмних систем, особливо в умовах динамічних конфігурацій. Некоректні базові значення можуть спричинити такі практичні наслідки:

- неправильні висновки щодо факторів, які впливають на надійність: метод може некоректно ідентифікувати ключові параметри, що ускладнює процес оптимізації ПС [8].
- проблеми з адаптацією ПС до динамічних змін: некоректні базові значення можуть спричинити помилкові прогнози стосовно впливу цих змін на надійність програмного забезпечення [9].
- підвищення ризику помилок в експлуатації: помилкова інтерпретація результатів може спричинити неправильні рішення у процесі підтримки чи оновленні системи, підвищуючи ризик збоїв у роботі програмного забезпечення [10].

Отже, правильний вибір базових значень є критично важливим для досягнення точності прогнозів і забезпечення надійності програмних систем, особливо в умовах динамічних конфігурацій.

У сучасній науковій літературі, щодо застосування методу SHAP (для оцінки якості програмних систем (ПС)), виявлено низку невирішених проблем, які потребують детального вивчення. Хоча SHAP є потужним інструментом для інтерпретації моделей машинного навчання, його використання в специфічних контекстах, таких як оцінка якості ПС, стикається з кількома обмеженнями. Зокрема, дослідження [11] вказують на труднощі в адаптації методу для моделей з великою кількістю вхідних параметрів або динамічними конфігураціями.

Основні проблеми практичного застосування SHAP під час оцінювання якості ПС такі:

- високі обчислювальні витрати: SHAP може бути ресурсоємним, особливо в контексті великих моделей, або наборів даних, що ускладнює його застосування в реальному часі [12];
- складність інтерпретації: у випадках, коли моделі мають багато взаємопов'язаних параметрів, результати SHAP можуть бути важкими для інтерпретації, що знижує їхню практичну цінність [13];
- нестабільність в динамічних умовах: SHAP може неадекватно відображати важливість параметрів у системах з частими змінами конфігурацій, що може спричинити помилкові висновки [14].

В роботі [15] розглянуто питання, щодо обмеженої масштабованості SHAP, через, що виникають проблеми з продуктивністю цього методу у разі обробки великих наборів даних, опрацювання яких вимагає значних ресурсів. Це особливо критично в сценаріях, де швидкість реагування є ключовою.

В праці [16] підкреслено, що в умовах сильної кореляції між вхідними змінними SHAP може неправильно розподілити значення важливості, що веде до хибних висновків про вплив окремих параметрів на результати моделі.

Відповідно до [17] SHAP вимагає ретельного налаштування для досягнення оптимальних результатів з різними моделями. Ця складність може стати перешкодою для аналітиків, які не мають достатнього досвіду у використанні методів інтерпретації.

Дослідження, які висвітлено в праці [18] вказують на обмежену гнучкість SHAP у врахуванні кількох критеріїв якості одночасно, що ускладнює його використання для комплексної оцінки якості програмних систем. Також у цій роботі зазначається, що SHAP може не відповідати певним галузевим стандартам, що обмежує його застосування в специфічних сферах, де вимагаються інші методи інтерпретації.

В праці [19] зазначається, що ефективність SHAP може значно варіювати залежно від вибраної моделі, що ставить під сумнів його універсальність у різних сценаріях.

Натомість в роботі [20] відмічається, що SHAP має обмеження у застосуванні до неструктурованих даних, таких як текст або зображення, що потребує додаткових адаптацій, а в праці [8], зазначено, що SHAP не завжди ефективно працює з новими, або складними архітектурами, такими як графові нейронні мережі, що обмежує його використання у сучасних дослідженнях.

У праці [21] вказується, що метод SHAP може вимагати великих об'ємів пам'яті для обчислень, що стає критичним у випадках обмежених ресурсів, також в цій праці підкреслюється, що застосування методу SHAP під час опрацювання категоріальних змінних призводить до втрати важливої інформації. Це пов'язано з тим, що SHAP спочатку перетворює категоріальні змінні на числові через процеси кодування.

Врахування зазначених проблем є критично важливим для вдосконалення методу SHAP в контексті оцінки якості програмних систем. У подальших дослідженнях варто зосередитися на пошуку рішень для усунення цих недоліків, через що інтерпретація моделей машинного навчання може бути точнішою та надійнішою.

Актуальність роботи зумовлена такими факторами:

1. Зростання складності сучасних нейронних мереж. Методи інтерпретації, такі як SHAP, грають ключову роль у розумінні того, як саме моделі ухвалюють рішення, особливо у випадках глибоких нейронних мереж (LSTM, CNN, RNN тощо), що використовуються для аналізу великих масивів даних;

2. Потребою у підвищенні прозорості та довіри до рішень, що приймаються штучним інтелектом. Інтерпретованість моделей є надзвичайно важливою у галузях, де рішення можуть мати значний вплив, як-от медицина, фінанси чи безпека. Оригінальні методи, такі як SHAP, допомагають пояснити, на яких факторах ґрунтується модель, але їхні можливості обмежені у випадку роботи зі складнішими типами моделей;

3. Необхідністю адаптивності моделей до змінюваних умов та нових даних. У реальних умовах дані можуть постійно змінюватися, і методи, які можуть швидко адаптуватися (наприклад, адаптивний SHAP, або баєсівські методи), стають вкрай важливими для забезпечення надійності та точності рішень;

4. Обмеженнями традиційних підходів до інтерпретації. Хоча оригінальний SHAP є потужним інструментом для пояснення рішень моделей, він не завжди підходить для нових архітектур, таких як часові ряди (LSTM), або багаторівнева агрегація (CNN), що вимагають удосконалених підходів для забезпечення точності та ефективності.

5. Вимогами до обчислювальної ефективності. В умовах швидкого зростання обсягів даних стає важливим знайти баланс між точністю, адаптивністю та обчислювальними витратами. Багато сучасних методів є дуже обчислювально складними, що впливає на час їхнього виконання та можливість застосування в реальних проектах.

Таким чином, робота є актуальною через необхідність розробки нових підходів до інтерпретації, адаптації та оптимізації нейронних мереж, що враховують специфіку різних типів моделей та нові виклики, пов'язані з обробленням великих обсягів даних. В результаті чого виникає нагальна практична потреба в пошуку шляхів удосконалення методу SHAP у разі оцінки якості ПС.

Метою роботи є підвищення точності та адаптивності методу SHAP шляхом його модифікацій для врахування різноманітних параметрів якості ПС, зокрема продуктивності, надійності, масштабованості та зручності використання. В роботі визначено такі завдання: провести детальний огляд проблем, пов'язаних із застосуванням методу SHAP під час оцінювання якості ПС; математично описати п'ять модифікацій методу SHAP, спрямованих на підвищення його точності, адаптивності та швидкості оцінки показників якості ПС; провести експериментальну перевірку запропонованих модифікацій для оцінювання їхньої ефективності у порівнянні з оригінальним методом.

Об'єктом дослідження є метод SHAP у контексті забезпечення оцінювання якості ПС.

Предметом дослідження є математичний апарат та алгоритмічні рішення щодо вдосконалення методу SHAP.

В рамках дослідження передбачається визначення п'яти оптимізаційних адаптаційних підходів для поліпшення методу SHAP.

Результати теоретичного дослідження

Математичне моделювання. Стандартний SHAP (MLP) базується на концепції ігор Шаплі, де кожний фактор оцінюється за його внеском у результат роботи моделі [13]. Для кожного фактора обчислюється його середній внесок у результат роботи моделі шляхом розрахунку різниці між прогнозами роботи з урахуванням та без урахування впливу цього фактора:

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{i\}) - f(S)], \quad (1)$$

де φ_i — значення Шаплі для фактора i ; N — набір всіх факторів; S — підмножина без фактора i ; $f(S)$ — прогноз для підмножини S .

Обмеження: висока обчислювальна складність за великої кількості факторів.

Шляхи удосконалення методу SHAP.

1. *Інтеграція часових рядів у SHAP для LSTM.*

LSTM (Long Short-Term Memory) моделі враховують часові залежності в даних, використовуючи спеціалізовані структури для збереження інформації про попередні стани. SHAP інтегрується в LSTM для оцінки впливу кожної ітерації на прогноз. Використовується стандартний SHAP для оцінки, але для LSTM включаються приховані стани та їхні ваги в розрахунок

$$SHAP_t^{(i)} = \sum_{k=1}^N \varphi_k^{(i)} \cdot \omega_k \cdot \Delta t_k, \quad (2)$$

де $SHAP_t^{(i)}$ — значення SHAP на момент часу t , $\varphi_k^{(i)}$ — ваговий коефіцієнт, ω_k — вага відповідного показника, Δt_k — часовий інтервал. В цьому випадку висока обчислювальна складність і потреба у великій кількості даних для навчання.

2. *Багаторівнева агрегація для CNN.*

CNN (Convolutional Neural Networks) використовують згорткові шари для оброблення просторових залежностей. Багаторівнева агрегація включає кілька рівнів згорткових шарів для виділення різних аспектів даних

$$Y = CNN(X) = \sigma(W \cdot X + b), \quad (3)$$

де Y — вихід, X — вхідні дані, W — ваги згорткових шарів, b — зсув, σ — активаційна функція. Це вимагає значних обчислювальних ресурсів та великих наборів даних.

3. *Адаптивний ваговий SHAP для MLP.*

Адаптивний ваговий SHAP коригує ваги кожного фактора в залежності від змін у даних. Вага для кожного фактора адаптується в залежності від його історичного впливу на результат моделі

$$\varphi_i = \frac{1}{N} \sum_{n=1}^N [f(x_i^n) - f(x_i^{n-1})], \quad (4)$$

де x_i^n — фактор на n -му етапі, f — функція прогнозу.

Такий підхід може бути менш стабільним за значних змін в даних.

4. *Комбінація ансамблів для змішаних моделей.*

Ансамбль моделей поєднує результати декількох моделей для підвищення точності прогнозу. Кожна модель (наприклад, LSTM, CNN, MLP) оцінює дані з різних перспектив

$$Y = \frac{1}{M} \sum_{m=1}^M Y_m, \quad (5)$$

де Y — середній прогноз, Y_m — прогноз m -ї моделі, M — кількість моделей.

Має місце висока обчислювальна вартість та складність інтеграції результатів.

5. *Баєсівське оновлення для RNN.*

RNN (Recurrent Neural Networks) з баєсівським оновленням використовують ймовірнісні методи для оновлення прогнозів на основі нових даних. Це дозволяє моделі адаптуватися до змін у даних

$$P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)}, \quad (6)$$

де $P(\theta|D)$ — апостеріорна ймовірність, $P(D|\theta)$ — ймовірність даних для заданих параметрів, $P(\theta)$ — апіорна ймовірність; D — це дані, які спостерігаються, або отримуються в результаті

вимірювань, експериментів, або спостережень (у контексті нейронних мереж це можуть бути вхідні дані (наприклад, зображення, текст, аудіо тощо)); θ — параметри моделі (наприклад, ваги нейронної мережі або гіперпараметри), які потрібно оцінити, або оптимізувати на основі даних. Обмеженням є необхідність в значних обчислювальних ресурсах для ймовірнісних розрахунків.

Експериментальне дослідження

Дані для експериментів отримано в рамках моделювання роботи ПС у реальних та синтетичних умовах. Для симуляції навантажень задіяно спеціалізовані інструменти для генерації користувацьких запитів і вимірювання показників системи. Apache JMeter (інструмент з відкритим кодом для тестування продуктивності, навантаження та функціональних можливостей програмного забезпечення): в нашому випадку цей інструмент застосовувався для моделювання різної кількості запитів від користувачів. Gatling (інструмент з відкритим кодом для тестування продуктивності та навантаження, який розроблений для симуляції великої кількості користувачів, що взаємодіють з веб-додатками, або API. Він широко використовується для оцінки ефективності, масштабованості та стабільності систем): в нашому випадку застосовувався для створення навантаження на сервер і моніторингу часу відповіді. При цьому навантаження змінювалося від 50 до 1350 одночасних користувачів.

1. Моніторинг продуктивності передбачав застосування системи Prometheus (система моніторингу та оповіщення з відкритим кодом, розроблена для збирання, зберігання та аналізу метрик з різних джерел у реальному часі. Вона широко використовується в сучасних інфраструктурах для спостереження за продуктивністю систем, серверів, додатків та інших компонентів) [11], зокрема для збору метрик продуктивності, таких як: завантаження процесора (CPU Load), об'єм пам'яті, що використовується (Memory Usage), час відповіді (Response Time). Моніторинг інтегровано з контейнерами Docker та Kubernetes для забезпечення стабільного оточення.

2. Збір даних про збої: Custom logging tools інтегрувалися з додатками, щоб відстежувати частоту збоїв (Failure Rate). Для цього аналізувалися: логи помилок, час простоїв компонентів.

3. Обладнання:

- серверна частина — хмарні сервери з AWS EC2;
- клієнтська частина — віртуальні машини з базовими налаштуваннями (2 ядра CPU, 4 GB RAM).

4. Програмні засоби:

- Python з бібліотеками Pandas і Scikit-learn для обробки даних;
- контейнеризація — Docker для стандартизації оточення;
- система оркестрації — Kubernetes для масштабування експериментів;
- бази даних — PostgreSQL для збереження отриманих показників.

5. Підготовка даних: зібрані дані проходили обробку для видалення шумів (аномалій) за допомогою методів виявлення викидів (Isolation Forest).

6. Нормалізація метрик виконувалася для забезпечення коректного аналізу параметрів, що мають різні шкали.

Початкові дані для практичного експериментального дослідження подано в табл. 1.

Таблиця 1

Початкові дані для практичного дослідження

Кількість користувачів (Users)	Час відповіді системи (ResponseTime), мс	Частота збоїв (FailureRate)	Завантаження процесора (CPULoad), %	Об'єм пам'яті, що використовується (MemoryUsage), GB
50	200	0,02	50	1
120	180	0,01	60	1,5
300	220	0,03	70	2
450	250	0,05	75	2,5
600	300	0,08	80	3
750	280	0,07	85	3,5
900	320	0,10	90	4
1050	340	0,12	95	4,5
1200	360	0,15	95	5
1350	380	0,18	98	5,5

В табл. 2 подано результати проведеного практичного дослідження.

Таблиця 2

Результати практичного дослідження

Метод	Параметр	Внесок у надійність	Внесок у продуктивність	Внесок у зручність	Внесок у масштабованість
Оригінальний SHAP	Users	0,30	0,25	0,20	0,25
	Response Time	0,25	0,30	0,25	0,20
	Failure Rate	0,20	0,15	0,15	0,20
	CPU Load	0,15	0,20	0,25	0,20
	Memory Usage	0,10	0,10	0,15	0,15
Часові ряди (LSTM)	Users	0,28	0,24	0,22	0,20
	Response Time	0,26	0,32	0,24	0,22
	Failure Rate	0,22	0,16	0,16	0,22
	CPU Load	0,14	0,18	0,26	0,20
	Memory Usage	0,10	0,10	0,12	0,18
Багаторівнева агрегація (CNN)	Users	0,32	0,23	0,21	0,22
	Response Time	0,24	0,33	0,27	0,20
	Failure Rate	0,21	0,17	0,15	0,21
	CPU Load	0,13	0,17	0,26	0,18
	Memory Usage	0,10	0,10	0,11	0,19
Адаптивний SHAP (MLP)	Users	0,30	0,25	0,20	0,25
	Response Time	0,25	0,30	0,25	0,20
	Failure Rate	0,20	0,15	0,15	0,20
	CPU Load	0,15	0,20	0,25	0,20
	Memory Usage	0,10	0,10	0,15	0,15
Ансамблі (Ensemble)	Users	0,35	0,22	0,20	0,23
	Response Time	0,22	0,35	0,29	0,18
	Failure Rate	0,19	0,15	0,14	0,19
	CPU Load	0,14	0,18	0,27	0,18
	Memory Usage	0,10	0,10	0,10	0,22
Басівське оновлення (RNN)	Users	0,33	0,23	0,21	0,22
	Response Time	0,23	0,34	0,28	0,19
	Failure Rate	0,20	0,15	0,14	0,20
	CPU Load	0,14	0,18	0,27	0,18
	Memory Usage	0,10	0,10	0,10	0,21

Подані в табл. 2 результати, отримані для різних методів оцінки показників якості, мають певні подібності та відмінності з тими, що описані в працях [1], [3], [14], [17], [18].

Подібності: для методів оригінальний SHAP та адаптивний SHAP (MLP) результати внесків до надійності, продуктивності, зручності та масштабованості практично ідентичні, що відповідає результатам, описаним у [1]. Це свідчить про стабільність результатів у різних застосуваннях цих методів.

Відмінності: ансамблеві методи (Ensemble) у дослідженні показали більший внесок у продуктивність через показник часу відповіді (0,35), тоді як у [3], [14] цей показник рівномірніше розподілений між іншими параметрами. Це може бути пов'язано з різними налаштуваннями моделей або характером використаних даних. Басівське оновлення (RNN) у цьому аналізі мало суттєвий внесок у масштабованість, особливо через параметри навантаження CPU і пам'яті, тоді як у [17], [18] ці показники впливали на масштабованість меншою мірою.

В табл. 3. подано результати практичного порівняння методів: оцінки показників, таких як точність, інтерпретованість, адаптивність та складність, визначені на основі результатів експериментального дослідження, яке порівнювало різні модифікації методу SHAP для прогнозування показників якості ПС.

1. *Точність (MAE — середня абсолютна помилка)*: точність кожного методу оцінювалася шляхом порівняння реальних значень метрик якості ПС з прогнозованими значеннями, отриманими за допомогою відповідних моделей. Для кожної моделі обчислювалася *середня абсолютна помилка (MAE)* між прогнозами та фактичними значеннями (що менша MAE, то вища точність методу).

2. *Інтерпретованість* оцінювалась за допомогою власних авторських оцінок, щодо того, наскільки легко і зрозуміло кожен метод пояснює внесок кожного параметра у прогнозовану метрику.

Таблиця 3

Результати практичного порівняння методів

Метод	Точність (MAE)	Інтерпретованість (оцінка)	Адаптивність (оцінка)	Час виконання, год	Складність реалізації (оцінка)
Оригінальний SHAP	0,84	7/10	6/10	0,4	5/10
Часові ряди (LSTM)	0,91	8/10	8/10	1,2	7/10
Багаторівнева агрегація (CNN)	0,88	9/10	7/10	1,5	8/10
Адаптивний SHAP (MLP)	0,86	8/10	9/10	1,0	6/10
Ансамблі (Ensemble)	0,89	8/10	8/10	1,3	7/10
Бассівське оновлення (RNN)	0,87	7/10	7/10	1,1	6/10

Для оцінки інтерпретованості використовувалася шкала від 1 до 10:

- 1/10 — дуже складно зрозуміти внесок параметрів.
- 10/10 — дуже чітко та прозоро пояснюється, як кожен параметр впливає на результат.

Згідно з [14] методи, які більше покладаються на складні алгоритми, такі як LSTM, або CNN, зазвичай мають меншу інтерпретованість, в той час як простіші підходи, наприклад, оригінальний SHAP або адаптивний SHAP (MLP), можуть бути інтуїтивно зрозумілішими.

3. *Адаптивність* вимірювалася за здатністю методу швидко реагувати на зміни в даних, зокрема аналізувалося:

- як метод справляється з новими, або зміненими даними в реальному часі (наприклад, у разі динамічних змін системи).

- оцінка адаптивності проводилася за допомогою авторських оцінок на шкалі від 1 до 10, де 1/10 — метод має низьку адаптивність, і 10/10 — метод здатен миттєво і точно адаптуватися до нових умов.

Згідно з [15] методи, як LSTM або RNN, можуть продемонструвати високий рівень адаптивності завдяки своїй здатності працювати з часовими рядами та новими даними.

4. *Складність реалізації* оцінювалася на основі таких факторів, як:

- ресурси, необхідні для впровадження методу (апаратні вимоги, програмне забезпечення);
- час, необхідний для налаштування та інтеграції;
- кількість етапів налаштування (до прикладу, тренування моделі, обробка даних, налаштування параметрів);
- оцінка складності проводилася за шкалою від 1 до 10, де 1/10 — дуже проста реалізація, а 10/10 — дуже складна, потребує багато часу та ресурсів.

У цьому аналізі методи для оцінки якості продемонстрували різну точність, адаптивність, інтерпретованість та складність реалізації. У порівнянні з працею [8], де застосовано подібні методи, спостерігаються такі подібності та відмінності:

Оригінальний SHAP:

- подібне: у нашому дослідженні точність (MAE) 0,84 збігається з результатами [10], які також показують задовільний рівень точності. Інтерпретованість та час виконання залишаються помірними, що підтверджує висновки [1]. Адаптивність (6/10) у нашому випадку оцінена дещо нижче ніж у [9], можливо через особливості даних або середовища.

- неподібне: адаптивність (6/10) у нашому випадку оцінена дещо нижче ніж у [9], можливо через особливості даних або середовища.

Часові ряди (LSTM):

- подібне: точність (MAE) 0,91 у нашому випадку відповідає високим показникам точності, згаданим у [11]. Метод демонструє високу адаптивність (8/10), аналогічно результатам [14].

- неподібне: у [12] час виконання LSTM може бути трохи ліпшим, що вказує на можливість оптимізації нашого процесу.

Багаторівнева агрегація (CNN):

- подібне: інтерпретованість (9/10) у нашому аналізі відображає чіткі візуалізації, які також зазначені у [10].

- неподібне: точність (0,88) дещо нижча ніж зазначено у [20], що може бути пов'язано з відмінностями в архітектурі або даних.

Таким чином, це дослідження загалом підтверджує результати, описані в праці [21], але також виявляє певні відмінності в точності та адаптивності деяких методів, що можуть бути зумовлені різними умовами проведення експериментів.

Висновки

Методи, що використовують глибокі нейронні мережі (LSTM, CNN), показують вищу точність та адаптивність у порівнянні з оригінальним SHAP, проте ціною цього є складність реалізації та тривалий час виконання. Адаптивний SHAP та Ensemble є оптимальними з погляду балансу між точністю, адаптивністю та інтерпретованістю, проте потребують певної оптимізації для зменшення часу виконання. Оригінальний SHAP показав задовільну точність (MAE 0,84) та інтерпретованість (7/10), проте поступається сучасним підходам у питаннях адаптивності (6/10) та часу виконання (0,4 години). Цей метод потребує подальшої оптимізації, зокрема в умовах динамічних середовищ, де важлива швидка адаптація до нових даних. Часові ряди (LSTM) забезпечили найвищу точність серед інших методів (MAE 0,91) та хорошу адаптивність (8/10). Висока складність реалізації (7/10) виправдана, оскільки метод дозволяє ефективно враховувати часові залежності. Проте час виконання (1,2 години) можна зменшити для підвищення ефективності. Багаторівнева агрегація (CNN) показала високу інтерпретованість (9/10) і хорошу точність (MAE 0,88), хоча й поступається LSTM. Метод має складнішу реалізацію (8/10), але це компенсується якісними візуалізаціями, що полегшують аналіз. Тривалий час виконання (1,5 години) вимагає оптимізації. Адаптивний SHAP (MLP) забезпечив хорошу точність (MAE 0,86) та високу адаптивність (9/10), що виділяє його одним з найгнучкіших методів. Час виконання (1 година) є конкурентним, а складність реалізації (6/10) нижча порівняно з CNN, через що він є доступним для ширшого кола користувачів. Ансамблі (Ensemble) показали високу точність (MAE 0,89) завдяки поєднанню кількох моделей. Адаптивність (8/10) та інтерпретованість (8/10) також на високому рівні, проте метод потребує більше часу для виконання (1,3 години). Реалізація є досить складною (7/10) через необхідність інтеграції різних моделей. Баєсівське оновлення (RNN) продемонструвало задовільну точність (MAE 0,87) та помірну адаптивність (7/10), що свідчить про його придатність для динамічних середовищ, хоча він поступається LSTM. Час виконання (1,1 години) є конкурентним, а складність реалізації (6/10) свідчить, що цей метод є менш складним порівняно з CNN.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] A. Altmann, L. Tološi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, no. 26 (10), pp.1340-1347, 2010. <https://doi.org/10.1093/bioinformatics/btq134>.
- [2] T. Amoriello, R. Ciccoritti, and P. Ferrante, "Prediction of strawberries' quality parameters using artificial neural networks," *Agronomy*, no. 12 (4), pp. 963, 2022. <https://doi.org/10.3390/agronomy12040963>.
- [3] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using Shapley Additive Explanations," *Expert Systems with Applications*, no. 186, pp.115736, 2021. <https://doi.org/10.1016/j.eswa.2021.115736>.
- [4] K. N. Dheeraj, et al., "Crop quality prediction using ml and neural networks," *International Journal on Cybernetics & Informatics*, no. 10 (2), pp. 07-11, 2021. <https://doi.org/10.5121/ijci.2021.100202>.
- [5] R. Ejrnæs, E. Aude, B. Nygaard, and B. Münier, "Prediction of habitat quality using ordination and neural networks," *Ecological Applications*, no. 12 (4), pp.1180-1187, 2002. [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] F. Fumagalli, M. Muschalik, E. Hüllermeier, and B. Hammer, "Incremental permutation feature importance (iPFI): Towards online explanations on data streams," *Machine Learning*, 2023. <https://doi.org/10.1007/s10994-023-06385-y>.
- [7] H. Ghanmi, A. Ghith, and T. Benameur, "Ring spun yarn quality prediction using hybrid neural networks," *The Journal of the Textile Institute*, pp.1-9, 2021. <https://doi.org/10.1080/00405000.2021.2022826>.
- [8] K. N. Jha, and C. T. Chockalingam, "Prediction of quality performance using artificial neural networks," *Journal of Advances in Management Research*, no. 6 (1), pp.70-86, 2009. <https://doi.org/10.1108/09727980910972172>.
- [9] H. Kaneko, "Cross-validated permutation feature importance considering correlation between features," *Analytical Science Advances*, 2022. <https://doi.org/10.1002/ansa.202200018>.
- [10] A. Michalski, K. Duraj, and B. Kupcewicz, "Leukocyte deep learning classification assessment using Shapley additive explanations algorithm," *International Journal of Laboratory Hematology*, 2023. <https://doi.org/10.1111/ijlh.14031>.
- [11] R. Miković, B. Arsić, and Đ. Gligorijević, "Importance of social capital for knowledge acquisition – DeepLIFT learning from international development projects," *Information Processing & Management*, no. 61(4), pp.103694, 2024. Accessed: 8 Nov. 2024. [Online]. <https://doi.org/10.1016/j.ipm.2024.103694>.
- [12] A. Movsessian, D. G. Cava, and D. Tcherniak, "Interpretable machine learning in damage detection using shapley additive explanations," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, no. 8 (2), 2022. <https://doi.org/10.1115/1.4053304>.
- [13] H. Ni, and H. Yin, "Exchange rate prediction using hybrid neural networks and trading indicators," *Neurocomputing*, no. 72 (13-15), pp. 2815-2823, 2009. <https://doi.org/10.1016/j.neucom.2008.09.023>.
- [14] S. Nirmalraj, A. S. M. Antony, et al. "Permutation feature importance-based fusion techniques for diabetes prediction," *Soft Computing*, 2023. <https://doi.org/10.1007/s00500-023-08041-y>.

- [15] I. Poligné, B. Broyart, G. Trystram, and A. Collignan, "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*, no. 35 (8), pp. 748-756, 2002. <https://doi.org/10.1006/fstl.2002.0939> .
- [16] N. H. A. Rahman, M. H. Lee, Suhartono, and M. T. Latif, "Artificial neural networks and fuzzy time series forecasting: An application to air quality," *Quality and Quantity*, no. 49 (6), pp. 2633-2647, 2014. <https://doi.org/10.1007/s11135-014-0132-6> .
- [17] A. M. Schweidtmann, et al., "Graph neural networks for prediction of fuel ignition quality," *Energy & Fuels*, no. 34(9), pp. 11395-11407. <https://doi.org/10.1021/acs.energyfuels.0c01533> .
- [18] C. H. Tan, K. M. Yusof, and S. R. W. Alwi, "Quality prediction for polypropylene extrusion based on neural networks," *IOP Conference Series: Materials Science and Engineering*, no. 1257 (1), pp. 012034, 2022. <https://doi.org/10.1088/1757-899x/1257/1/012034> .
- [19] J. S. Tan, "Ablation study on feature group importance for automated essay scoring," *Asia-Pacific Journal of Information Technology and Multimedia*, no. 11 (01), pp. 90-101, 2022. <https://doi.org/10.17576/apjitm-2022-1101-08> .
- [20] M. M. T. Thwin, and T.-S. Quah, "Application of neural networks for software quality prediction using object-oriented metrics," *Journal of Systems and Software*, no. 76 (2), pp. 147-156, 2005. <https://doi.org/10.1016/j.jss.2004.05.001> .
- [21] S. Wang, and Y. Zhang, "Grad-CAM: Understanding AI models," *Computers, Materials & Continua*, no. 76 (2), pp. 1321-1324, 2023. <https://doi.org/10.32604/cmc.2023.041419> .

Рекомендована кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ

Стаття надійшла до редакції 3.10.2024

Шантур Антон Сергійович — канд. техн. наук, доцент кафедри штучного інтелекту, e-mail: anton.shantyr@gmail.com .

Державний університет інформаційно-комунікаційних технологій, Київ

A. S. Shantyr¹

Predicting Software System Quality Indicators Using Modifications of the SHAP Method

¹State University of Information and Communication Technologies, Kyiv

This paper proposes five suggestions for improving the practical application of the SHAP method in the context of software system (SS) quality assessment. The study explores the potential for enhancing the SHAP method (Shapley Additive Explanations) in forecasting the quality indicators of software systems. The aim of the research is to increase the accuracy and adaptability of the SHAP method through modifications that account for various SS quality parameters, including performance, reliability, scalability, and usability. The objectives of the study are the following: to conduct a detailed review of the issues associated with the application of the SHAP method in SS quality assessment; to mathematically describe five methods for modifying the SHAP method to improve its accuracy, adaptability, and speed in evaluating SS quality indicators; and to experimentally verify the proposed modifications to assess their effectiveness compared to the original method. The study examines five methods for enhancing the SHAP method: LSTM (Long Short-Term Memory); CNN (Convolutional Neural Networks); adaptive SHAP; MLP (Multilayer Perceptron); and RNN (Recurrent Neural Network, including model ensembles and Bayesian updating). A practical comparison of the results demonstrated that the proposed variations of SHAP could significantly improve accuracy and prediction speed, especially in dynamic conditions and large data volumes. The development and justification of the five optimization modifications of the SHAP method achieved improvements in accuracy and forecasting efficiency for SS quality indicators. The research found that methods using deep neural networks (LSTM, CNN) show higher accuracy and adaptability compared to the original SHAP; however, this comes at the cost of implementation complexity and longer execution times. Adaptive SHAP and Ensemble are optimal in terms of balancing accuracy, adaptability, and interpretability, but require some optimization to improve execution time. The original SHAP showed satisfactory accuracy (MAE 0.84) and interpretability (7/10), but lagged behind modern approaches in adaptability (6/10) and execution time (0.4 hours). SHAP needs further optimization, especially in dynamic environments where rapid adaptation to new data is crucial. Time series (LSTM) provided the highest accuracy among the methods (MAE 0.91) and good adaptability (8/10).

Keywords: optimization, deep learning model, Bayesian updating, adaptability.

Shantyr Anton S. — Cand. Sc. (Eng.), Associate Professor of the Chair of Artificial Intelligence, e-mail: anton.shantyr@gmail.com