УДК 004.85:519.711:004.414.2 DOI: 10.31673/2412-9070.2026.318108

А. В. ЛЕБЕДЄВ, аспірант;
ORCID: 0009-0004-0665-5937
Н. В. ФЕДОРОВА, д-р техн. наук, професор,
ORCID ID 0000-0002-4548-4198
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

ПАРАМЕТРИЗОВАНИЙ МАТЕМАТИЧНИЙ АПАРАТ МУТАЦІЙНОГО АНАЛІЗУ СИСТЕМ МАШИННОГО НАВЧАННЯ

Забезпечення якості та надійності моделей машинного навчання є актуальним науковим завданням, особливо в контексті їх зростаючого застосування у критичних щодо безпеки прикладних галузях. Мутаційний аналіз є одним із найбільш зрілих методів оцінювання якості тестових наборів, однак його застосування до систем машинного навчання ускладнене через стохастичність навчання та інференсу, відсутність детермінованого оракула, а також вплив еквівалентних і тривіальних мутантів на результуючу оцінку. Ключовою проблемою є відсутність єдиної математичної формалізації, яка охоплювала б існуючі підходи як частинні випадки та забезпечувала їх коректне порівняння, оскільки поняття мутаційної оцінки у різних роботах визначається по-різному залежно від обраного правила фіксації відмінності між оригінальною моделлю і мутантом та від способу агрегування результатів.

У статті запропоновано формалізацію мутаційного аналізу для систем машинного навчання як параметризованої обчислювальної системи. Визначено базові об'єкти процесу: параметризоване відображення навченої моделі, множину операторів мутації, множину згенерованих мутантів, тестовий набір та узагальнену метрику якості, що охоплює поширені показники для задач класифікації та регресії. Узагальнену мутаційну оцінку визначено як параметризований функціонал, що задається парою компонентів - правилом «вбивства» мутанта у пороговому або статистично обґрунтованому варіанті та агрегатором результатів по множині мутантів. Статистичний варіант правила поєднує критерій значущості з оцінкою розміру ефекту, що забезпечує відтворюваність для стохастичних моделей. Показано, що існуючі підходи є частинними реалізаціями запропонованого апарату. Сформульовано оптимізаційну задачу відбору підмножини операторів мутації і тестових прикладів, що максимізує мутаційну оцінку за умов ресурсних обмежень. Запропонований апарат забезпечує формальну основу для коректного порівняння підходів до мутаційного тестування та для розроблення ефективних алгоритмів мутаційного аналізу моделей машинного навчання.

Ключові слова: мутаційне тестування, математична модель, машинне навчання, моделі машинного навчання, обробка великих масивів даних.

Вступ

Мутаційний аналіз є одним із найбільш зрілих методів оцінювання якості тестових наборів, однак його застосування до систем машинного навчання ускладнене через стохастичність навчання та інференсу, відсутність детермінованого оракула, а також вплив еквівалентних і тривіальних мутантів. Ключовою проблемою є відсутність єдиної математичної формалізації, яка охоплювала б існуючі підходи як частинні випадки, оскільки поняття мутаційної оцінки у різних роботах визначається по-різному залежно від правила фіксації відмінності між оригінальною моделлю і мутантом та способу агрегування результатів.

У статті запропоновано формалізацію мутаційного аналізу для систем машинного навчання як параметризованої обчислювальної системи. Узагальнену мутаційну оцінку визначено як функціонал, що задається парою компонентів – правилом «вбивства» мутанта у пороговому або статистично обґрунтованому варіанті та агрегатором результатів по множині мутантів. Показано, що існуючі підходи є частинними реалізаціями запропонованого апарату. Сформульовано оптимізаційну задачу відбору операторів мутації і тестових прикладів за умов ресурсних обмежень.

Постановка проблеми

Моделі машинного навчання (МН) набувають широкого застосування у системах, що безпосередньо впливають на безпеку та добробут людини: медична діагностика, автономне керування транспортом, фінансовий аналіз, енергетичні системи. У таких умовах якість і надійність моделей МН перестають бути суто інженерною характеристикою та стають предметом наукового забезпечення.

Одним із найбільш зрілих методів оцінювання якості тестових наборів у класичному програмному забезпеченні є мутаційний аналіз. Метод ґрунтується на введенні контрольованих змін або мутацій, у компоненти системи та перевірці здатності тестів виявляти ці зміни. Однак пряме перенесення мутаційного аналізу на системи МН є проблематичним через ряд факторів: стохастичність процесу навчання та інференсу, відсутність детермінованого оракула для значної частини задач, а також специфіка об'єкта мутації. Усе це має значний вплив на семантику ключових понять методу.

Незважаючи на активний розвиток інструментів мутаційного тестування для моделей МН, у літературі досі відсутня єдина математична формалізація процесу мутаційного аналізу, яка охоплювала б різні підходи як частинні випадки. Наявні роботи пропонують конкретні метрики та оператори, проте не забезпечують уніфікованого апарату, придатного для коректного порівняння методів, аналізу їхніх властивостей та постановки задач оптимізації. Зокрема, поняття «мутаційної оцінки» у різних роботах визначається по-різному, що унеможливлює їх пряме зіставлення та коректну інтерпретацію результатів.

Таким чином, розроблення формального математичного апарату мутаційного аналізу для систем МН, який дозволив би уніфіковано описати існуючі підходи та формально поставити задачу оптимізації процедури аналізу, є актуальним науковим завданням, що має безпосередній зв'язок із проблематикою забезпечення якості та верифікації систем МН у критичних застосуваннях.

Аналіз останніх досліджень і публікацій

Теоретичні засади мутаційного аналізу у традиційному програмному забезпеченні закладено у роботі Ю. Цзя та М. Гарман [1], де систематизовано розвиток методу, сформульовано гіпотезу компетентного програміста та гіпотезу зчеплення, а також проведено огляд існуючих підходів і інструментів. Питання підтримки процесів забезпечення якості засобами мутаційного тестування детально розглянуто у систематичному огляді Ц. Чжу, А. Панічелли та Е. Зайдмана [2]. Проте зазначені роботи орієнтовані на детерміновані системи з чітко визначеним оракулом і не розглядають специфіку систем МН.

Однією з перших спроб адаптації мутаційного аналізу до моделей глибокого навчання є робота DeepMutation Л. Ма та ін. [3], де запропоновано оператори мутації як на рівні тренувальної програми, так і на рівні навченої моделі, але без чіткого розмежування між цими двома класами мутацій. Розвитком цього підходу стала робота DeepMutation++ [4], де формалізовано набір операторів мутації на рівні моделі та запропоновано інструментальну підтримку їх автоматизованого застосування. Г. Джахангірова та П. Тонелла [5] показали, що визначення «вбивства» через порогове падіння точності не враховує стохастичну природу навчання та запропонували статистично обґрунтоване правило «вбивства», що забезпечує відтворюваність результатів.

Розвитком статистичного підходу стала робота DeepCrime [6], де оператори мутації засновані на таксономії реальних дефектів у системах МН і застосовуються на рівні тренувальної програми. Ф. Тамбон та ін. [7] запропонували ймовірнісний підхід до мутаційного тестування глибоких нейронних мереж, спрямований на усунення нестабільності результатів, притаманної існуючим статистичним методам. А. Панічелла та С. Ліем [8] критично переосмислили концептуальні засади мутаційного аналізу для МН та ідентифікували невідповідності класичним гіпотезам. З. Чжан та ін. [9] показали, що не всі оператори мутації однаково корисні для оцінювання якості тестів, що обґрунтовує доцільність їх відбору в умовах ресурсних обмежень.

Аналіз наведених публікацій засвідчує, що кожен із методів визначає мутаційну оцінку по-різному, залежно від правила фіксації відмінності між оригіналом і мутантом та способу агрегування результатів, і жоден не пропонує уніфікованого апарату для їх зіставлення. Наскільки відомо авторам, ця проблема не отримала систематичного висвітлення у науковій літературі, що додатково підтверджує актуальність дослідження.

Мета і задачі дослідження

Метою статті є розроблення параметризованого математичного апарату мутаційного аналізу систем машинного навчання, що охоплює існуючі підходи як частинні випадки і допускає постановку задач оптимізації. Для досягнення мети вирішуються такі задачі:

- формалізація базових об'єктів і операторів процесу мутаційного аналізу для моделей МН;
- визначення узагальненого правила «вбивства» мутанта та агрегатора результатів;
- порівняльний аналіз існуючих методів у термінах запропонованого апарату;
- формулювання оптимізаційної задачі за ресурсних обмежень.

Результати дослідження

Мутаційний аналіз є підходом до оцінювання якості тестів і здатності системи виявляти дефекти шляхом штучного внесення контрольованих змін у програмний артефакт та подальшого порівняння поведінки «оригіналу» і «мутантів» [1]. У класичному мутаційному тестуванні мутант вважається «вбитим», якщо тестовий набір фіксує відмінність у результатах виконання, а інтегральною оцінкою виступає частка вбитих мутантів [2].

Для систем машинного навчання пряме перенесення цього підходу є проблемним через стохастичність навчання й інференсу, відсутність детермінованого «оракула» для значної частини задач, а також через явища тривіальних та еквівалентних мутантів. Тому, доцільною є формалізація процесу мутаційного аналізу моделей МН як обчислювальної системи, у межах якої правило «вбивства» та мутаційна оцінка розглядаються як параметризовані компоненти, що допускають різні реалізації.

Формалізація об’єктів і базових операторів процесу

Нехай навчена модель машинного навчання задається параметризованим відображенням:

$$f_\theta = \chi \to \gamma, \tag{1}$$

де χ – простір вхідних даних, γ – простір виходів, θ – вектор параметрів моделі. Мутаційний аналіз розглядає контрольовані модифікації моделі f_θ з метою виявлення чутливості її поведінки до потенційних дефектів у параметрах або структурі.

У цій роботі розглядаються оператори мутації, що застосовуються до вже навченої моделі, тобто модифікуються параметри θ або структура f_θ без повторного навчання. Такий підхід обґрунтовано у попередній роботі авторів [10], де проаналізовано специфіку застосування мутаційного аналізу до моделей МН та визначено клас пост-тренувальних мутацій як базовий об'єкт дослідження.

Визначимо множину операторів мутації:

$$O = \{o_1, o_2, \dots, o_k\}, \tag{2}$$

де кожен оператор o_i породжує мутанта шляхом модифікації f_θ:

$$o_i: f_\theta \to f_{\theta_i}^m, i = 1,2, \dots, k, \tag{3}$$

де $f_{\theta_i}^m$ – мутована модель, а k – кількість операторів мутації.

Таким чином, множина всіх згенерованих мутантів має вигляд:

$$M = \{f_{\theta_1}^m, f_{\theta_2}^m, \dots, f_{\theta_N}^m\}, \tag{4}$$

де N – кількість згенерованих мутантів. У загальному випадку N може перевищувати k за рахунок параметризації мутацій або композиції мутаційних операторів, що важливо для подальшого узгодження з підходами, які оперують конфігураціями мутацій.

Нехай задано тестовий набір:

$$T = \{t_1, t_2, \dots, t_n\}, t_j = (x_j, y_j), x_j \in \chi, y_j \in \gamma, \tag{5}$$

де n – кількість тестових прикладів. Для кількісного порівняння поведінки оригінальної моделі та її мутантів визначимо узагальнену функцію якості:

$$\Phi(f, T) = \frac{1}{n} \sum_{j=1}^{n} \varphi(f(x_j), y_j), \tag{6}$$

де $\varphi(f(x_j), y_j)$ – локальна функція якості або втрат, що задається відповідно до класу задачі (класифікація, регресія тощо). Таке визначення охоплює поширені показники якості та забезпечує єдину основу для подальшого визначення відхилення між моделями [3].

Відхилення між результатами оригінальної та мутованої моделей оцінюється як:

$$\Delta_i = \left|\Phi(f_\theta, T) - \Phi\left(f_{\theta_i}^m, T\right)\right|. \tag{7}$$

Величина Δ_i інтерпретується як міра зміни поведінки моделі внаслідок застосування мутації. У контексті моделей машинного навчання Δ_i може залежати від стохастичних чинників, таких як ініціалізація, порядок обробки даних, випадкові компоненти інференсу [5]. На практиці стохастичність оцінки враховується через S незалежних повторів, що формує вибірку $\left\{\Delta_i^{(r)}\right\}_{r=1}^{S}$. Тому, надалі розглядаються як детерміновані порогові правила «вбивства», так і статистичні критерії, що враховують варіативність оцінок.

Правило «вбивства» мутанта та параметризована мутаційна оцінка

У класичному мутаційному тестуванні мутаційна оцінка визначається як частка мутантів, поведінка яких відрізняється від оригіналу на тестовому наборі [1], [2]:

$$MS_{classic} = \frac{1}{N} \sum_{i=1}^{N} J_\varepsilon\left(f_\theta, f_{\theta_i}^m, T\right), \tag{8}$$

де правило «вбивства» задається предикатом:

$$J_\varepsilon(f_\theta, f_{\theta_i}^m, T) = \begin{cases} 1, & \Delta_i > \varepsilon, \\ 0, & \Delta_i \le \varepsilon, \end{cases} \tag{9}$$

де $\varepsilon > 0$ – порогове значення, що задає мінімально значущу зміну якості, яка інтерпретується як «виявлення дефекту». Проте безпосереднє застосування такого визначення до систем МН є проблематичним через низку чинників, що суттєво впливають на коректність та інтерпретованість оцінки.

По-перше, суттєвий вплив на оцінку мають еквівалентні та тривіальні мутанти. **Еквівалентним** вважається мутант, який формально змінює модель, але не дає розрізненої поведінки на тестовому наборі, що занижує оцінку. **Тривіальним** є мутант, що спричиняє настільки грубу деградацію якості, що «вбивається» майже будь-яким тестовим набором і, відповідно, не характеризує здатність тестів виявляти нетривіальні помилки. Для задач із неперервними

виходами, такими як регресія чи ймовірнісні прогнози, ці проблеми посилюються, оскільки значущі зміни можуть проявлятися як невеликі поведінкові зсуви, які бінарний критерій відображає грубо або нестабільно.

По-друге, у багатьох задачах МН відсутній детермінований «оракул», тому фіксація «вбивства» не зводиться до перевірки рівності виходів. Практично це означає, що правило J має бути прив'язане до обраної метрики якості Φ або до поведінкових характеристик моделі, а не до бінарної різниці результатів.

По-третє, вибір ε є предметно залежним і, за відсутності емпіричної калібрації, може призводити до завищення частки «вбитих» мутантів за рахунок тривіальних змін або до заниження за рахунок нечутливості до суттєвих, але малих за величиною ефектів.

На основі виявлених обмежень формалізуємо правило «вбивства» та мутаційну оцінку як параметризовані компоненти. Для врахування стохастичності оцінювання, введемо простір випадкових факторів Ω та випадкову величину $\Delta_i(\omega)$, $\omega \in \Omega$. Тоді природним узагальненням порогового правила є його застосування до математичного сподівання:

$$J_\varepsilon^E(f_\theta, f_{\theta_i}^m, T) = I[\, E_{\omega\sim\Omega}[\Delta_i(\omega)] > \varepsilon\,]. \tag{10}$$

Однак така форма не відрізняє випадки, коли середній ефект є малим, але стабільним, від випадків, коли він формується за рахунок високої дисперсії. Тому, більш коректним з позицій відтворюваності є введення статистичного правила «вбивства», яке поєднує критерій значущості відмінності та оцінку розміру ефекту. Формально це можна подати як:

$$J_{stat}(f_\theta, f_{\theta_i}^m, T) = I[p(\Delta_i) \le \alpha \wedge\ g(\Delta_i) \ge \tau], \tag{11}$$

де $p(\Delta_i)$ – p-значення для гіпотези про відсутність відмінності між результатами оригіналу і мутанта за серією прогонів, α – рівень значущості, $g(\Delta_i)$ – оцінка розміру ефекту (наприклад, стандартизована різниця або інша міра), τ – мінімально прийнятний ефект. Такий підхід узгоджується з ідеєю «статистичного вбивства» мутантів [5], [6], що використовується у працях, де бінарна інтерпретація замінюється статистично обґрунтованою процедурою.

Побудувавши предикат J, визначимо спосіб агрегування результатів по множині M. Позначимо результат застосування правила «вбивства» до i-го мутанта як:

$$J_i = J(f_\theta, f_{\theta_i}^m, T) \in \{0,1\}, i = 1, \dots, N. \tag{12}$$

Тоді агрегатор A у найпростішому випадку задається як арифметичне середнє:

$$A(\{J_i\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N J_i. \tag{13}$$

Водночас для систем машинного навчання агрегатор A доцільно розглядати як параметризований оператор, що допускає зважування мутантів, групування за типами мутацій або нормування за підмножинами тестів:

$$A_w(\{J_i\}_{i=1}^N) = \frac{\sum_{i=1}^N w_i J_i}{\sum_{i=1}^N w_i}, \tag{14}$$

де $w_i \ge 0$ – ваги, що відображають складність мутації або пріоритетність для певного класу дефектів.

Тепер узагальнена мутаційна оцінка визначається як:

$$MS(J, A) = A(\{J(f_\theta, f_{\theta_i}^m, T)\}_{i=1}^N). \tag{15}$$

Визначена декомпозиція є принципово важливою для контексту МН, оскільки дозволяє узгоджено описувати різні підходи до мутаційної оцінки, що відрізняються як способом фіксації відмінності між оригіналом і мутантами, так і способом інтеграції цих відмінностей у підсумковий показник.

Уніфікована інтерпретація підходів мутаційних оцінок через компоненти

У підході DeepMutation [3] в контексті запропонованого апарату інтерпретуються мутації на рівні моделі, де оператори безпосередньо модифікують параметри навченої моделі без повторного навчання [10]. Правило «вбивства» формалізується через критерій KD3 [3], за яким фіксація «вбивства» здійснюється через зміну класифікаційного рішення на рівні окремих класів, а підсумкова оцінка нормується не лише за кількістю мутантів, але й за кількістю класів.

Нехай G – множина класів, M – множина мутантів. Для кожного мутанта $f_{\theta_i}^m \in M$ визначимо множину «вбитих класів» $K(T, f_{\theta_i}^m) \subseteq G$ як множину класів, для яких існує принаймні один приклад $t \in T$, на якому прогноз мутанта відрізняється від прогнозу оригіналу:

$$K(T, f_{\theta_i}^m) = \{c \in G | \exists t \in T: f_\theta(t) = c \wedge f_{\theta_i}^m(t) \neq c\}. \quad (16)$$

Тоді агрегатор мутаційної оцінки DeepMutation задається як нормована кількість «вбитих класів» по всіх мутантах:

$$A_{DM} = \frac{\sum_{f_{\theta_i}^m \in M} |K(T, f_{\theta_i}^m)|}{|M| \cdot |G|}. \quad (17)$$

Отже, DeepMutation узгоджується зі схемою $MS(J, A)$ з поведінково-орієнтованим J та агрегатором, нормованим по мутантах і класах.

У підході Джахангірової та Тонели [5] «вбивство» визначається статистично на основі серії повторних оцінювань. Нехай для кожного оператора мутації $o \in O$ виконується K повторів оцінювання (різні ω_r), у межах яких порівнюються результати оригінальної моделі f_θ та моделі, отриманої застосуванням оператора o. На основі цих K повторів обчислюються p-значення та розмір ефекту.

Тоді правило «вбивства» задається у вигляді предиката:

$$J_{JT}(o) = I\,[p_o \leq \alpha \ \wedge | \ d_o \ | \geq \tau], \quad (18)$$

де p_o – p-значення для гіпотези про відсутність відмінності між оригіналом і мутованим варіантом, d_o – оцінка розміру ефекту, α – рівень значущості, τ – мінімально прийнятний ефект.

На відміну від класичної агрегації за кількістю мутантів, у цьому підході підсумковий показник агрегується **на рівні операторів мутації.** У спрощеному записі, що відповідає структурі підходу [5], де кожен оператор оцінюється як єдиний предикат, агрегатор можна визначити як частку «вбитих» операторів:

$$A_{JT} = \frac{1}{|O|} \sum_{o \in O} K_o. \quad (19)$$

У термінах $MS(J, A)$ метод запроваджений Джахангіровою та Тонелою відповідає статистичному правилу J, що забезпечує відтворюваність рішення «вбитий» або «не вбитий», та агрегатору A, який узагальнює результати не по окремих мутантах, а по операторах мутацій, зменшуючи чутливість оцінки до кількості та параметризації мутантів усередині одного оператора.

У підході DeepCrime [6] оператори мутації O задаються як мутації на рівні тренувальної програми та формуються на основі таксономій реальних дефектів у системах МН. «Вбивство» також визначається статистично, для оригінальної моделі f_θ та відповідного мутанта виконуються повторні навчання, і порівнюються розподіли значень метрики якості Ф на фіксованому наборі даних.

Нехай $o \in O$ – оператор мутації, $c \in C_o$ – його конфігурація, а $f_\theta^{m(o,c)}$ – модель, отримана застосуванням пари (o, c) до оригіналу. Для фіксованого набору даних D (у роботі розглядаються тренувальні та тестові набори) формуються вектори значень метрики за n незалежних навчань:

$$\Phi(f_\theta, D) = \left\langle \Phi\left(f_\theta^{(1)}, D\right), \ldots, \Phi\left(f_\theta^{(n)}, D\right) \right\rangle, \quad (20)$$

$$\Phi(f_\theta^{m(o,c)}, D) = \langle \Phi((f_\theta^{m(o,c)})^{(1)}, D), \dots, \Phi((f_\theta^{m(o,c)})^{(n)}, D) \rangle. \quad (21)$$

Тоді правило «вбивства» у DeepCrime визначається як:

$$J_{DC}(o, c; D) = I[p(\Phi(f_\theta, D), \Phi(f_\theta^{m(o,c)}, D)) < \alpha \wedge d(\Phi(f_\theta, D), \Phi(f_\theta^{m(o,c)}, D)) \geq \beta], \quad (22)$$

де p – p-значення, d – розмір ефекту.

Далі DeepCrime оцінює тестовий набір через простір конфігурацій оператора:

$$C_o = C_1 \times \dots \times C_k. \quad (23)$$

Позначимо множину «вбитих конфігурацій», для яких оператор мутації здатен породити мутант, який «вбивається» на тренувальних даних:

$$K(o, TrainS) = \{ c \in C_o \mid J_{DC}(o, c; TrainS) = 1 \}. \quad (24)$$

Мутаційна оцінка для оператора визначається як частка конфігурацій з $K(o, TrainS)$, що є «вбитими» також і на тестових даних:

$$A_{DC}(o) = \frac{|\{c \in K(o, TrainS) \mid J_{DC}(o, c; TestS) = 1\}|}{|K(o, TrainS)|}. \quad (25)$$

Нормування на $| K(o, TrainS) |$ означає, що до оцінки включаються лише ті конфігурації, для яких оператор мутації здатен породити вбиваний мутант. Загальна мутаційна оцінка тестового набору у DeepCrime визначається як середнє значення $A_{DC}(o)$ за всіма операторами $o \in O$, тобто шляхом агрегування оператор-орієнтованих оцінок у єдиний показник.

Формальна постановка оптимізаційної задачі

Запропонована модель мутаційної оцінки $MS(J, A)$ дозволяє формально поставити задачу оптимізації процедури мутаційного аналізу з урахуванням обчислювальних обмежень. Оскільки повний перебір операторів, конфігурацій та тестів є ресурсомістким, а внесок різних операторів у оцінку нерівномірний [9], доцільно розглядати вибір підмножин, що забезпечують максимальну інформативність за заданого бюджету.

Нехай $O' \subseteq O$ – обрана підмножина операторів мутації, $T' \subseteq T$ – підмножина тестових прикладів, а $M(O')$ – множина мутантів, згенерованих на основі O'. Тоді оптимізаційну постановку можна записати як задачу максимізації мутаційної оцінки:

$$\max_{O' \subseteq O,\ T' \subseteq T} MS_{O', T'}(J, A), \quad (26)$$

де:

$$MS_{O', T'}(J, A) = A(\{J(f_\theta, f_{\theta_i}^m, T')\}_{f_{\theta_i}^m \in M(O')}). \quad (27)$$

Оскільки побудова множини мутантів та їх оцінювання потребують обчислювальних ресурсів, введемо ресурсну функцію $R(O', T')$, яка узагальнює витрати на генерацію мутантів, повторні запуски, за потреби статистичного J, та обчислення метрики Ф на підмножині тестів. Тоді оптимізаційна задача формулюється з бюджетним обмеженням:

$$\max_{O' \subseteq O,\ T' \subseteq T} MS_{O', T'}(J, A) \text{ за умови } R(O', T') \leq R_{max}, \quad (28)$$

Таким чином, у межах запропонованого апарату оптимізація мутаційного аналізу зводиться до вибору структури мутацій та обсягу оцінювання на тестах під задані ресурсні обмеження. Розв'язання цієї задачі становить окремий напрям подальших досліджень, пов'язаних з емпіричною калібрацією параметрів запропонованого апарату та розробленням алгоритмів.

Висновки та перспективи подальших досліджень

У роботі запропоновано уніфікований математичний апарат мутаційного аналізу для систем машинного навчання, що ґрунтується на формалізації процесу як параметризованої обчислювальної системи. Мутаційну оцінку визначено як функціонал, що задається парою компонентів – правилом «вбивства» мутанта та агрегатором результатів. Така декомпозиція дозволяє описати існуючі підходи як частинні реалізації єдиної схеми, які відрізняються способом

фіксації відмінності між оригінальною моделлю і мутантами та способом нормування результатів. Сформульовано оптимізаційну задачу відбору підмножини операторів мутації і тестових прикладів, що максимізує мутаційну оцінку за умов ресурсних обмежень.

Подальші дослідження пов'язані з емпіричною калібрацією параметрів апарату та розробленням алгоритмів розв'язання сформульованої оптимізаційної задачі.

Внесок авторів

Артем ЛЕБЕДЄВ – концептуалізація, розроблення математичного апарату, формалізація моделі, аналіз джерел, підготовка та написання рукопису; Наталія ФЕДОРОВА – наукове керівництво, верифікація математичного апарату, критичний перегляд та редагування рукопису.

Декларація про штучний інтелект

Під час підготовки рукопису автори використовували інструмент штучного інтелекту для лінгвістичної корекції тексту та перевірки граматичної узгодженості формулювань українською та англійською мовами. Усі наукові результати, математичні формулювання та висновки отримано авторами самостійно. Автори несуть повну відповідальність за зміст статті.

Конфлікт інтересів

Автори заявляють про відсутність конфлікту інтересів та підтверджує, що під час підготовки цієї роботи не існувало жодних комерційних, фінансових чи інших взаємовідносин, які могли б бути розцінені як такі, що здатні вплинути на результати дослідження або їх інтерпретацію. Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

Список використаної літератури

1. Jia, Y., & Harman, M. (2011). An analysis and survey of the development of mutation testing. IEEE Transactions on Software Engineering, 37(5), 649–678. https://doi.org/10.1109/tse.2010.62

2. Zhu, Q., Panichella, A., & Zaidman, A. (2018). A systematic literature review of how mutation testing supports quality assurance processes. Software Testing, Verification and Reliability, 28(6), Стаття e1675. https://doi.org/10.1002/stvr.1675

3. Ma, L., Zhang, F., Sun, J., Xue, M., Li, B., Juefei-Xu, F., Xie, C., Li, L., Liu, Y., Zhao, J., & Wang, Y. (2018). DeepMutation: Mutation testing of deep learning systems. У 2018 IEEE 29th international symposium on software reliability engineering (ISSRE). IEEE. https://doi.org/10.1109/issre.2018.00021

4. Hu, Q., Ma, L., Xie, X., Yu, B., Liu, Y., & Zhao, J. (2019). DeepMutation++: A mutation testing framework for deep learning systems. У 2019 34th IEEE/ACM international conference on automated software engineering (ASE). IEEE. https://doi.org/10.1109/ase.2019.00126

5. Jahangirova, G., & Tonella, P. (2020). An empirical evaluation of mutation operators for deep learning systems. У 2020 IEEE 13th international conference on software testing, validation and verification (ICST). IEEE. https://doi.org/10.1109/icst46399.2020.00018

6. Humbatova, N., Jahangirova, G., & Tonella, P. (2021). DeepCrime: Mutation testing of deep learning systems based on real faults. У ISSTA '21: 30th ACM SIGSOFT international symposium on software testing and analysis. ACM. https://doi.org/10.1145/3460319.3464825

7. Tambon, F., Khomh, F., & Antoniol, G. (2022). A probabilistic framework for mutation testing in deep neural networks. Information and Software Technology, 107129. https://doi.org/10.1016/j.infsof.2022.107129

8. Panichella, A., & Liem, C. C. S. (2021). What are we really testing in mutation testing for machine learning? A critical reflection. У 2021 IEEE/ACM 43rd international conference on soft-

ware engineering: New ideas and emerging results (ICSE-NIER). IEEE. https://doi.org/10.1109/icse-nier52604.2021.00022

9. Zhang, Z., Wang, Y., Yao, Y., Wang, Z., & Huang, Z. (2025). A fine-grained evaluation of mutation operators to boost mutation testing for deep learning systems. Empirical Software Engineering, 30(3). https://doi.org/10.1007/s10664-025-10613-5

10. Лебедєв, А. В., & Федорова, Н. В. (2025). Мутаційний аналіз та проблеми його використання для моделей машинного навчання. Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки, 36(75, 2), 112–117. https://doi.org/10.32782/2663-5941/2025.2.2/15

A. Lebediev, N. Fedorova

PARAMETERIZED MATHEMATICAL FRAMEWORK FOR MUTATION ANALYSIS OF MACHINE LEARNING SYSTEMS

Ensuring the quality and reliability of machine learning models is a pressing research challenge, particularly in the context of their growing adoption in safety-critical domains. Mutation analysis is one of the most established methods for evaluating test suite quality; however, its application to machine learning systems is complicated by the stochastic nature of training and inference, the absence of a deterministic oracle, and the influence of equivalent and trivial mutants on the resulting score. A key issue is the lack of a unified mathematical formalization that would encompass existing approaches as particular cases and enable their correct comparison, since the notion of mutation score is defined differently across works depending on the chosen rule for detecting differences between the original model and mutants and the method of aggregating results.

The paper proposes a formalization of mutation analysis for machine learning systems as a parameterized computational system. The basic objects of the process are defined: a parameterized mapping of the trained model, a set of mutation operators, a set of generated mutants, a test suite, and a generalized quality metric covering common measures for classification and regression tasks. The generalized mutation score is defined as a parameterized functional determined by two components: a mutant killing rule and a result aggregator. The killing rule is considered in two forms – a threshold-based variant and a statistically grounded variant that combines a significance criterion with an effect size measure, ensuring reproducibility for stochastic models. It is shown that existing approaches are particular realizations of the proposed framework. An optimization problem is formulated for selecting a subset of mutation operators and test inputs that maximizes the mutation score under resource constraints. The proposed framework provides a formal foundation for the correct comparison of mutation testing approaches and for the development of efficient mutation analysis algorithms for machine learning models.

Keywords: mutation testing, mathematical model, machine learning, machine learning models, processing of big data arrays.

Надійшла до редакції: 16.04.2026
Прийнята до друку: 03.06.2026
Опубліковано: 29.06.2026