ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)*

УДК 004.85:004.94 DOI: https://doi.org/10.30837/2522-9818.2025.3.152

О. Соловей

ЗВАЖЕНА МЕТРИКА ЧУТЛИВОСТІ ДЛЯ ПРОГНОЗУВАННЯ ЗАТРИМКИ В *KAFKA*-КЛАСТЕРІ

Предметом дослідження є комбінований підхід до аналізу чутливості для комплексного оцінювання впливу конфігураційних параметрів *Kafka*-кластера на кінцеву затримку в системах потокового оброблення інформації. **Мета роботи** – розроблення нового підходу до аналізу чутливості, що поєднує класичні методи оцінювання впливу параметрів (методи Морріса та Соболя) з метриками, що відтворюють структурні зміни розподілу вихідних даних (евклідова відстань, хеллінгерова відстань, *J*-дивергенція). Такий підхід дає змогу дослідити вплив параметра не лише з погляду амплітуди його ефекту, а й щодо змін у формі та структурі ймовірнісного розподілу результатів. Для досягнення мети розв'язуються такі **завдання**: формальне визначення нового підходу до аналізу чутливості; розроблення мережі Баєса для моделювання наскрізної затримки *Kafka*-кластера; аналіз чутливості за запропонованим підходом; експериментальне дослідження нового підходу з використанням розрахованих відповідно до нього ваг впливу для ініціалізації матриці ваг нейронної мережі, що прогнозує наскрізну затримку в *Kafka*-кластері залежно від обраної конфігурації. Для реалізації поставлених завдань у дослідженні впроваджено такі **методи**: теорія експериментів, евклідова геометрія, статистична теорія розподілів, інформаційна теорія, теорія машинного навчання, баєсівська статистика й теорія графів. **Досягнуті результати.** Для оцінювання ефективності підходу проведено порівняльне навчання нейронної мережі з різними стратегіями ініціалізації ваг. Аналіз функції витрат, побудованої за критерієм мінімізації середньоквадратичної похибки, продемонстрував, що найменших значень вона досягає саме для моделі, яка була ініціалізована вагами, отриманими за запропонованим підходом до оцінювання впливу параметрів на вихідну змінну моделі. **Висновки.** У дослідженні запропоновано новий підхід до аналізу чутливості. Новизна підходу полягає в інтеграції переваг як причинно-орієнтованих, так і дисперсійно-оцінних методів у межах єдиної зваженої метрики чутливості. Практична цінність підходу полягає в тому, що його застосування під час аналізу чутливості або ініціалізації матриці ваг нейронної мережі дає змогу підвищити точність оцінювання впливу параметрів, покращити збіжність моделі та скоротити час її навчання.

Ключові слова: метод Морріса; індекс Соболя; евклідова відстань; хеллінгерова відстань; *J*-дивергенція; аналіз чутливості.

Вступ

Інформаційні технології для управління великими даними є ключовим інструментом управління проєктами міського будівництва, оскільки забезпечують оброблення значних обсягів різнорідної інформації з різних джерел. До таких джерел належать системи управління бізнес-процесами (ERP), взаємодією із зацікавленими сторонами (CRM), охороною праці та ризиками на будівництві, експлуатацією об'єктів, а також системи для проєктування, створення моделей просторових об'єктів, інженерного аналізу й застосунки доповненої та віртуальної реальності (AR/VR) (рис. 1). Ефективне функціювання цих технологій вимагає створення гнучкої, масштабованої та надійної інформаційної архітектури, здатної обробляти структуровані, напівструктуровані та неструктуровані потоки даних, які частково надходять у режимі реального часу й частково в режимі пакетного оброблення [1].

Отже, архітектура інформаційної технології для управління інформацією проєктів міського будівництва передбачає такі шари: прийом потокових даних; прийом пакетних даних; оброблення та збереження потокових даних; аналітика потокових даних (рис. 2).

Шар прийому потокових даних (рис. 2) включає кластер *Apache Kafka*, що забезпечує отримання інформації в реальному часі з низькою затримкою. Висока пропускна здатність у процесі передачі повідомлень від *Kafka*-продюсера до *Kafka*-споживача є однією з ключових вимог до інформаційних технологій, призначених для управління різнорідними даними проєктів міського будівництва. Конфігурація *Kafka*-кластера забезпечує високу продуктивність за умови відповідності її параметрів потребам системи. Втім, велика гнучкість у налаштуванні конфігурації *Kafka* спричиняє суттєву невизначеність у продуктивності кластера *Apache Kafka*, зокрема щодо кінцевої затримки, яка визначається як час від моменту надсилання інформації продюсером до їх отримання споживачем.

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

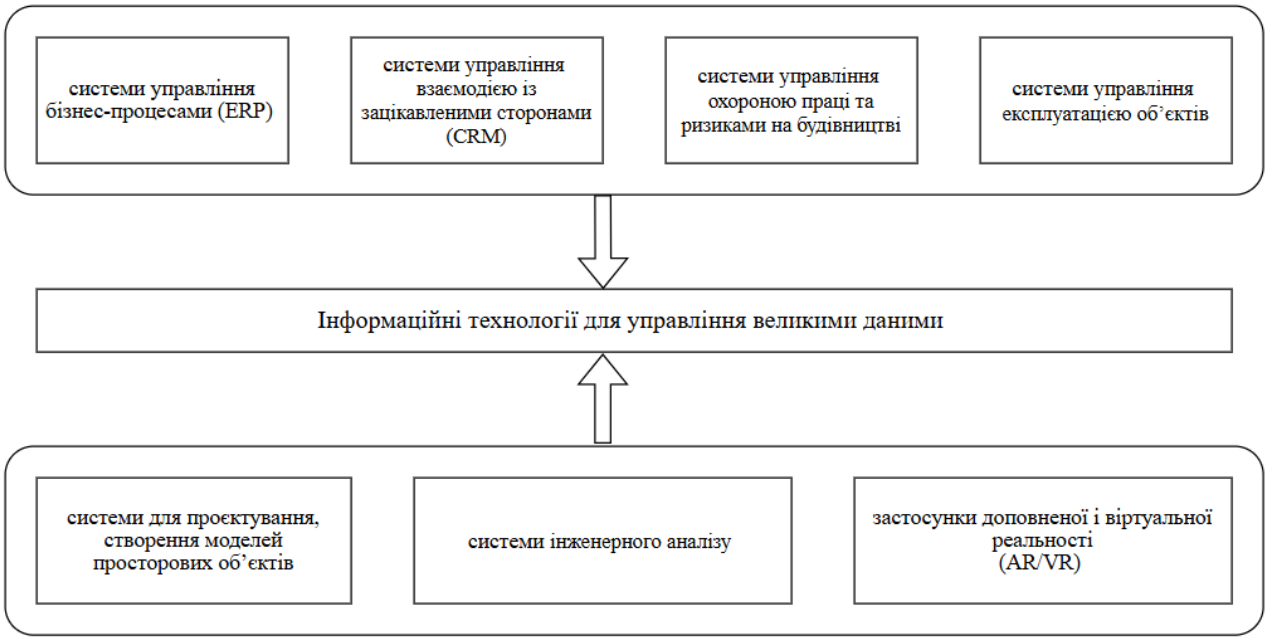

Рис. 1. Джерела даних "Інформаційні технології для управління великими даними проєктів міського будівництва"

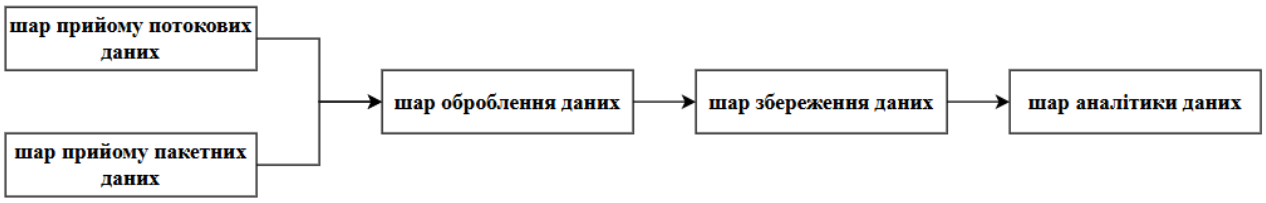

Рис. 2. Архітектура інформаційної технології для управління даними проєктів міського будівництва

Наявні дослідження здебільшого спрямовані на покращення окремих показників продуктивності, таких як затримка або відмовостійкість, залишаючи поза увагою приховані взаємозв'язки між параметрами конфігурації та чутливість системи до локальних змін значень. Унаслідок цього можуть формуватися субоптимальні або нестійкі конфігурації, що погано масштабуються в умовах реального використання.

Попри важливість цього аспекту, вибір оптимальної *Kafka* конфігурації залишається складним завданням. Труднощі полягають в необхідності динамічного налаштування параметрів з метою запобігання перевантаженням, мінімізації витрат, а також забезпечення високої доступності й низької затримки.

Аналіз останніх досліджень і публікацій

Коли модель глибокого навчання використовується для визначення конфігурації кластера *Apache Kafka* на основі прогнозування затримки в кластері, вибір початкової матриці ваг має значний вплив на збіжність навчання та кінцеву продуктивність моделі, оскільки суттєво впливає на активацію нейронів [2]. За замовчуванням застосовується метод Ксав'єра (*Xavier Uniform*) [3], що ґрунтується на рівномірному розподілі ваг залежно від кількості нейронів на вході та виході:

$$w \sim U\left(-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}}\right), \qquad (1)$$

де U – рівномірний розподіл; *nin, nout* – кількість нейронів, що додані в шари та пов'язані матрицею ваг.

Однак нещодавні дослідження [4] вказують на необхідність зважати на причинно-наслідкові зв'язки в прийнятті рішень, що актуалізує пошук нових підходів до ініціалізації ваг. Серед альтернативних методів: ініціалізація на основі підкласового аналізу дискримінантів (SDA) [5], *ZerO*-ініціалізація з використанням матриць Адамара [6], методи масштабування ваг [7] та адаптивне масштабування під час навчання [8]. Проте зазначені методи

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

здебільшого не беруть до уваги причинно-наслідкові впливи, що обмежує їх ефективність у роботі з неповними або структурно складними даними.

Аналіз чутливості є важливим підходом для визначення впливу вхідних параметрів на вихідні характеристики системи або моделі. У праці [9] запропоновано чотириетапну структуру, що поєднує метамоделювання з методами Морріса та Соболя. Це дає змогу ефективно виявляти ключові фактори, що впливають на енергоспоживання, знижуючи водночас обчислювальні витрати. У роботі [10] порівняно різні методи чутливості, зокрема DLOR і FAST, для аналізу ознак, релевантних до продуктивності моделей CNN. Найбільш ефективними виявились методи Морріса та Соболя.

Подальші дослідження вдосконалюють методику обчислення індексів Соболя. Автори праці [10] запропонували спрощений підхід із використанням індексів Шеплі, де глобальні ефекти параметрів оцінюються першими, а решта – виводяться лінійними перетвореннями. Дослідження [11] демонструє застосування глобального аналізу чутливості для визначення пріоритетних параметрів у життєвому циклі виробництва. У статті [12] запропоновано підхід для аналізу дискретної баєсівської мережі за допомогою евклідової відстані, хеллінгерової відстані та *J*-дивергенції. Ці метрики дають змогу оцінити схожість і розбіжності між імовірнісними розподілами, що формуються внаслідок варіацій параметрів.

Метод Морріса дає змогу оцінити локальний вплив кожного параметра через варіацію вихідної функції, тоді як індекси Соболя дають глобальну уяву про внесок параметра до загальної дисперсії. Їх комбіноване застосування розкриває повну картину впливів, однак не зважає на причинно-наслідкові залежності між параметрами.

Метрики відстані між імовірнісними розподілами (евклідова, хеллінгерова відстані, *J*-дивергенція) можуть бути інтегровані в причинно-орієнтовані моделі, зокрема баєсівські мережі, що забезпечує взяття до уваги структурних впливів на зміну розподілів. Проте вони не фіксують змінність самої функції результату, як це роблять класичні методи аналізу чутливості.

Отже, наразі відсутній уніфікований підхід, який би поєднував такі фактори:

1) взяття до уваги структурної подібності між результатами на основі метричних відстаней;

2) оцінювання прямого впливу параметрів (за допомогою індексів Морріса μ_i);

3) взяття до уваги взаємодій між параметрами (за допомогою індексів Соболя S_i).

Це створює передумови для розроблення нового підходу до аналізу чутливості, який поєднує переваги як причинно-орієнтованих, так і дисперсійно-оцінних методів у єдиній зваженій метриці.

Мета роботи й завдання

Мета дослідження – запропонувати новий підхід для аналізу чутливості, який поєднує класичні методи оцінювання впливу параметрів (методи Морріса та Соболя) з метриками, що відтворюють структурні зміни розподілу вихідних даних (евклідова відстань, хеллінгерова відстань, *J*-дивергенція). Такий підхід дасть змогу дослідити вплив параметра не лише з погляду амплітуди його ефекту, але й щодо змін у формі та структурі ймовірнісного розподілу результатів.

Для досягнення окресленої мети необхідно розв’язати такі завдання:

1) формально визначити новий підхід до аналізу чутливості, який поєднає методи Морріса μ_i та Соболя S_i з одним із методів (евклідова відстань, хеллінгерова відстань або *J*-дивергенція);

2) розробити мережу Баєса для моделювання наскрізної затримки *Kafka*-кластера;

3) проаналізувати чутливість за методами евклідової відстані, хеллінгерової відстані, *J*-дивергенції на розробленій мережі Баєса;

4) проаналізувати чутливість за методами Морріса μ_i та Соболя S_i на моделі *XGBRegressor*;

5) проаналізувати чутливість за запропонованим підходом;

6) провести експериментальне дослідження запропонованого підходу, використовуючи ваги впливу для ініціалізації матриці ваг нейронної мережі, що прогнозує наскрізну затримку в *Kafka*-кластері залежно від обраної конфігурації.

Матеріали й методи

Нехай $L(X)$ – це затримка в кластері *Kafka* з конфігурацією X, змодельована функцією $f(X)$, де $X = \{x_1, x_2, \ldots, x_n\}$ – набір незалежних параметрів

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

кластера *Kafka*. Загальна дисперсія затримки $L(X)$, розкладена за вхідними параметрами, описується як

$$Var\left(L(X)\right)=\sum_{i=1}^{N}V_i+\sum_{1\le i<j\le N}V_{ij}+\ldots+V_{1,2,\ldots n}, \quad (2)$$

$$V_i=Var_{X_i}E_{X_{\sim i}}\left(f(X)\,|\,X_i\right), \quad (3)$$

$$V_{ij}=Var_{X_{ij}}E_{X_{\sim ij}}\left(f(X)\,|\,X_i,X_j\right)-V_i-V_j, \quad (4)$$

де $V_{1,\,2,\ldots,\,N}$ – дисперсія, що виникає внаслідок взаємодії всіх ознак у наборі X.

Індекс Соболя першого порядку оцінює, яка частка дисперсії затримки $L(X)$, зумовлена впливом лише одного вхідного параметра x_i:

$$S_i=\frac{V_i}{Var(Y)}. \quad (5)$$

Метод Морріса для обчислення μ_i:

$$\mu_i=\frac{1}{R}\sum_{r=1}^{R}\left|d_i^r\right|, \quad (6)$$

де d_i^r – це елементарний ефект вхідного параметра на r-й ітерації.

$$d_i^r=\frac{f\left(x^r+\Delta e_i\right)-f\left(x^r\right)}{\Delta}, \quad (7)$$

де x^r – випадково обраний вектор на r-й ітерації;

Δe_i – одиничний вектор, що змінює тільки параметр x_i;

Δ – крок у сітці пошуку.

Нехай загальний розподіл затримки незалежно від окремих значень параметрів визначено:

$$P(L)=\sum_{X}P(L|X)P(X). \quad (8)$$

Умовний розподіл за фіксованого параметра $x_i=a$:

$$P\left(L\,|\,x_i=a\right)=\sum\nolimits_{X_{\sim i}}P\left(L|x_i=a,X_{\sim i}\right)\cdot P\left(X_{\sim i}\right), \quad (9)$$

де $X_{\sim i}$ – всі параметри, крім x_i.

Евклідова відстань (D_E) двох розподілів імовірностей визначається за сумою квадратів різниць розподілів:

$$D_E=\sqrt{\sum_{l\in L}\left(P(L=l)-P\left(L=l\,|\,x_i=a\right)\right)^2}. \quad (10)$$

Відстань Хеллінгера (D_H) вимірює відстань між двома ймовірнісними розподілами й основана на квадратичній різниці їх імовірностей:

$$D_H=\frac{1}{\sqrt{2}}\sqrt{\sum_{l\in L}\left(\sqrt{P(L=l)}-\sqrt{P\left(L=l\,|\,x_i=a\right)}\right)^2}. \quad (11)$$

Відстань J-дивергенції (D_J) оцінює, наскільки один розподіл відрізняється від іншого за допомогою порівняння їх "втрат" інформації щодо середнього розподілу:

$$D_J=\sum_{l\in L}P(L=l)\log\frac{P(L=l)}{P\left(L=l\,|\,x_i=a\right)}+ +P\left(L=l\,|\,x_i=a\right)\log\frac{P\left(L=l\,|\,x_i=a\right)}{P(L=l)}. \quad (12)$$

Підхід до аналізу чутливості, який поєднує методи Морріса μ_i (5) та Соболя S_i (6) з одним із методів (10)–(12), формально опишемо як зважену комбінацію:

$$W_{D_E}=\alpha\cdot S_i+\beta\cdot\mu_i+\gamma\cdot D_E; \quad (13)$$

$$W_{D_H}=\alpha\cdot S_i+\beta\cdot\mu_i+\gamma\cdot D_H; \quad (14)$$

$$W_{D_J}=\alpha\cdot S_i+\beta\cdot\mu_i+\gamma\cdot D_J. \quad (15)$$

Вибір значень для параметрів α, β, γ залежить від завдання, а саме: варто збільшити параметр α, якщо необхідно визначити параметри з найвищим глобальним впливом; потрібно збільшити параметр β, якщо необхідно взяти до уваги середній вплив параметра незалежно від його локальних або глобальних властивостей; потрібно збільшити параметр γ, якщо необхідно взяти до уваги структурну подібність між результатами на основі метричних відстаней.

Для порівняльного аналізу результатів оцінювань за методами (13)–(15) скористаємося класифікацією:

$$O(W_i)=\begin{cases}"H", & \text{якщо } W_i\ge Q_{67}^i\\ "M", & \text{якщо } Q_{33}^i\le W_i<Q_{67}^i\\ "L", & \text{якщо } W_i<Q_{33}^i\end{cases}, \quad (16)$$

де $"H"$, $"M"$, $"L"$ – класи оцінки W_i – "сильний", "середній" та "низький";

$W_i=\left(W_{ij}-\min(W_i)\right)/\left(\max(W_i)-\min(W_i)\right)$ – нормалізована оцінка;

i – індекс методу з множини $O\left[W_{De},W_{D_H},W_{D_J}\right]$;

j – індекс параметра конфігурації з множини X;

Q_{67}^i, Q_{33}^i – 67-й, 33-й процентилі, обчислені на W_i.

Для ініціалізації нейронів першого прихованого шару нейронної мережі з метою прогнозування затримок *Kafka*-кластера з вагами впливу $O(W_i)$,

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

обчисленими відповідно до (16), запропонуємо правило "більшості голосів":

$$W(x) = argmax_{o \in O} \mid i : O(W_i) = o \mid . \quad (17)$$

Ініціалізація нейронів першого прихованого шару нейронної мережі вагами W виконується за виразом

$$h^1 = f\left(W\vec{X} + b^1\right), \quad (18)$$

де h^1 – вихід першого прихованого шару нейронної мережі;

b^1 – вектор зсуву для першого прихованого шару;

$f(\bullet)$ – функція активації, що застосовується до лінійної комбінації вхідних даних.

Архітектура *Kafka*-кластера містить сервери, застосунки, що надсилають потокову інформацію на сервер (продюсерів) та клієнтів (*consumers*), які під'єднуються для отримання даних та сервісу *ZooKeeper*, який використовується *Kafka* для управління та координації серверів, доданих у *Kafka*-кластер [14, 15]. Наскрізну затримку (*end-to-end latency*) потокового оброблення подій у *Kafka*-кластері можна виміряти часом *TL*, який дорівнює сумі таких часових інтервалів: час, необхідний продюсеру для збирання подій у пакет перед їх відправленням на сервер (далі – *Tproducer*); час від моменту отримання події в буфері сервера до її збереження на диску сервера-лідера (далі – *Tleadercommit*); час від моменту збереження повідомлення на диску сервера-лідера до завершення його копіювання на інші сервери в кластері *Kafka* (далі – *Treplica*); час від моменту, коли продюсер *Kafka* отримав підтвердження від сервера-лідера (параметр "*acks_config*" = 1 або "*acks_config*" = "*all*") або коли певна кількість інформації була збережена на диску сервера-лідера ("*acks_config*" = 0), до моменту, коли клієнт *Kafka* отримує подію (далі – *Tfetch*). Отже, формальне визначення наскрізної затримки в кластері *Kafka* має такий вигляд:

$$TL = T_{producer} + T_{leadercommit} + T_{replica} + T_{fetch}. \quad (19)$$

На рис. 3 подано запропоновану структуру дискретної мережі Баєса для моделювання наскрізної затримки в кластері *Kafka*, побудовану на основі аналізу архітектури *Kafka*-кластера. Ця структура містить шар спостережувальних змінних $X = \{x_1, \ldots, x_{11}\}$, що відповідають параметрам конфігурації *Kafka*-кластера, а саме: $x_1(acks\_config)$ – визначає рівень підтвердження, який вимагає сервер-лідер від продюсерів; $x_2(buffer.memory)$ – визначає обсяг пам'яті в байтах, яку продюсер може використовувати для формування пакета даних перед надсиланням на сервер; $x_3(max.inflight.requests.per.connection)$ – встановлює максимальну кількість непідтверджених запитів, які можна надіслати на сервер за допомогою одного з'єднання; $x_4(socket.receive.buffer.bytes)$ – задає розмір мережевого буфера сокета для прийняття інформації сервером; $x_5(max.poll.records)$ – визначає максимальну кількість записів, що *Kafka*-клієнт може обробити за один виклик методу $poll()$; $x_6(log.flush.interval.ms)$ – встановлює час, протягом якого повідомлення може залишатися в пам'яті сервера перед записом на диск; $x_7(replica.fetch.min.bytes)$ – вказує на мінімальний обсяг інформації, який має бути скопійований на сервері перед надсиланням запиту на отримання нових даних від сервера-лідера; $x_8(linger.ms)$ – визначає, як довго продюсер накопичуватиме інформацію для формування пакета перед відправленням на сервер-лідер; $x_9(batch.size)$ – максимальний розмір пакета даних у байтах, який продюсер надсилає на сервер-лідер за один раз; $x_{10}(compression.type)$ – тип стиснення інформації, який використовує продюсер перед надсиланням; $x_{11}(fetch.min.bytes)$ – мінімальна кількість байтів, що має бути записана на диск сервера-лідера перед тим, як стати доступною для клієнтів.

Обрані параметри конфігурації $X = \{x_1, \ldots, x_{11}\}$ безпосередньо впливають на показники продуктивності кластера *Kafka* $Y = \{y_1, \ldots, y_6\}$. Визначимо їх як приховані змінні мережі Баєса та включимо у перший прихований шар, а саме: $y_1(RequestQueueTimeMs)$ – протягом якого дані, надіслані продюсером, перебувають у черзі запитів перед прийняттям для збереження в журналі подій сервера; $y_2(ResponseQueueTimeMs)$ – час у мілісекундах, протягом якого дані, збережені на сервері, перебувають у черзі відповідей; $y_3(LogFlushRateAndTimeMs)$ – частота й час, з якими дані, збережені в журналі логів сервера, переміщуються з пам'яті сервера на диск; $y_4(BytesInPerSec)$ – загальна кількість байтів, які Kafka сервер отримує щосекунди від усіх продюсерів. Високі значення можуть свідчити про необхідність додати обчислювальні ресурси або скоригувати конфігурацію продюсерів; $y_5(BytesOutPerSec)$ –

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

вимірює швидкість, з якою дані надсилаються з сервера до клієнтів; $y_6(batch\_size\_avg)$ – середній розмір пакета повідомлень, який продюсер надсилає до брокера *Kafka*.

Другий прихований шар містить складники виразу (19), оскільки їх значення формуються зі значень показників продуктивності $Y=\{y_1,\ldots,y_6\}$, а саме: y_1^1 – загальний час продюсера (*Tproducer*) в мілісекундах; y_2^1 – загальний час збереження на диску сервера-лідера (*Tleadercommit*) в мілісекундах; y_3^1 – загальний час копіювання даних на інші сервери (*Treplica*) в мілісекундах; y_4^1 – загальний час для отримання даних клієнтами (*Tfetch*) в мілісекундах. Вихідний шар містить єдину змінну – наскрізну затримку в кластері *Kafka* (*TL*), для якої запропонована мережа Баєса здійснюватиме прогнозування. Для побудови дискретної баєсівської мережі значення кожного вузла має бути дискретизоване. Для цього подаємо кожну змінну за нотацією

$$\langle N,\ S,\ G(S)\rangle\ , \qquad (20)$$

де N – назва змінної; S – множина станів $S_{i=\overline{1,N}}=\{s_i\}$, поданих у вигляді лінгвістичних термів; $G(S)$ – множина значень для кожного стану $s_i\in S$.

Для визначення таблиць умовних імовірностей для вузлів мережі Баєса скористаємося алгоритмом максимальної правдоподібності (MLE) [16].

Для обчислення оцінок впливу за методами Соболя та Морріса необхідна модель, здатна точно прогнозувати значення вихідної функції як для випадкових варіацій вхідних змінних, згенерованих за методом Морріса, так і для дисперсій вхідних змінних, згенерованих за методом Соболя. Для цього дослідження оберемо модель *XGBRegressor*, алгоритм якої полягає в побудові ансамблю дерев рішень, де кожне нове дерево додається до ансамблю з метою мінімізації помилок, припущених попередніми моделями дерев, збільшуючи в такий спосіб точність моделі [17]. Модель *XGBRegressor* побудуємо для параметрів конфігурації, що належать до множини X спостережувальних змінних мережі Баєса та мають неперервні значення.

Результати досліджень та їх обговорення

На рис. 4–6 подано обчислені за методами (10)–(12) оцінки впливу на запропонованій мережі Баєса (рис. 3). Порівняння результатів отриманих значень сил впливу за методами *DE*, *DH*, *DJ* вказує на певні розбіжності.

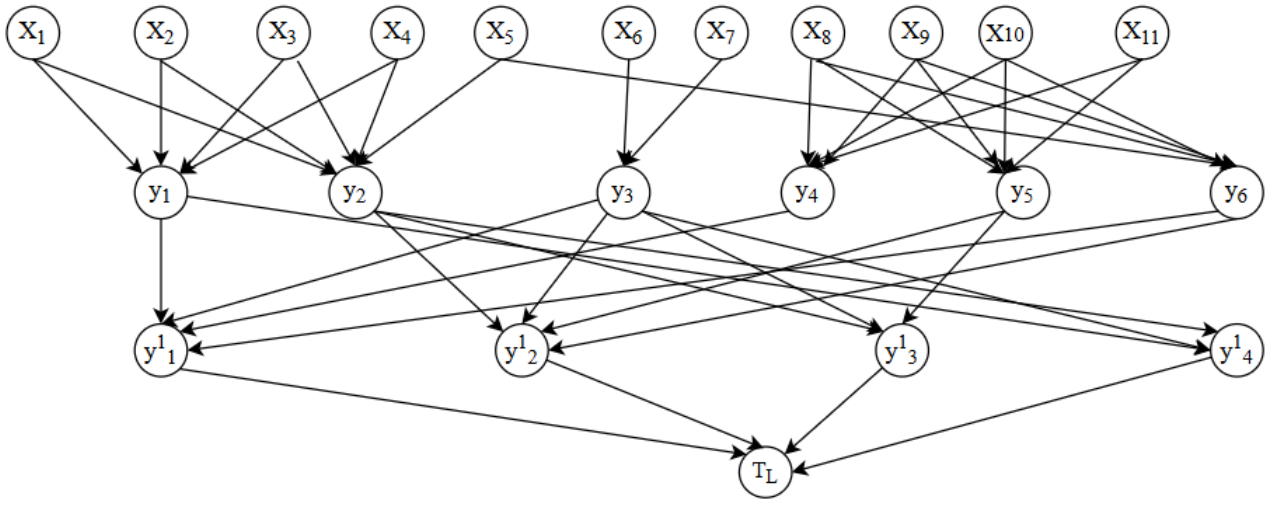

Рис. 3. Розроблена структура дискретної мережі Баєса для моделювання наскрізної затримки в кластері *Kafka*

1. Шар із спостережувальними змінними $X=\{x_1,\ \ldots,x_{11}\}$.

1.1. За методом D_E сильний вплив мають: $x_6(log.flush.interval.ms)$ на $y_3(LogFlushRateAndTimeMs)$; $x_8(linger.ms)$ на $(y_3(LogFlushRateAndTimeMs)$; $y_4(BytesInPerSec))$; $x_9(batch.size)$ на $y_4(BytesInPerSec)$; $x_{10}(compression.type)$ на $y_4(BytesInPerSec)$. Середній вплив оцінено для $x_8(linger.ms)$ на $y_5(BytesOutPerSec)$;

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

$x_9(batch.size)$ на $y_5(BytesOutPerSec)$; $x_{10}(compression.type)$ на $y_5(BytesOutPerSec)$.

1.2. За методом D_H сильний вплив мають: $x_5(max.poll.records)$ на $y_6(batch\_size\_avg)$; $x_8(linger.ms)$ на $y_6(batch\_size\_avg)$; $x_9(batch.size)$ на $y_6(batch\_size\_avg)$; $x_{10}(compression.type)$ на $y_6(batch\_size\_avg)$. Середній вплив оцінено $x_1(acks\_config)$ на $y_1(RequestQueueTimeMs)$; $x_2(buffer.memory)$ на $y_1(RequestQueueTimeMs)$; $x_9(batch.size)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_8(linger.ms)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_{10}(compression.type)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_3(max.inflight.requests.per.connection)$ на $y_2(ResponseQueueTimeMs)$; $x_4(socket.receive.buffer.bytes)$ на $y_1(RequestQueueTimeMs)$.

1.3. За методом D_J сильний вплив мають: $x_5(max.poll.records)$ на $y_6(batch\_size\_avg)$; $x_8(linger.ms)$ на $y_6(batch\_size\_avg)$; $x_9(batch.size)$ на $y_6(batch\_size\_avg)$; $x_{10}(compression.type)$ на $y_6(batch\_size\_avg)$. Середній вплив оцінено $x_1(acks\_config)$ на $y_1(RequestQueueTimeMs)$; $x_2(buffer.memory)$ на $y_1(RequestQueueTimeMs)$; $x_9(batch.size)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_{11}(fetch.min.bytes)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_8(linger.ms)$ на $(y_4(BytesInPerSec); y_5(BytesOutPerSec))$; $x_3(max.inflight.requests.per.connection)$ на $y_1(RequestQueueTimeMs)$; $x_4(socket.receive.buffer.bytes)$ на $y_1(RequestQueueTimeMs)$.

2. Перший шар з прихованими змінними $Y=\{y_1,...,y_6\}$.

2.1. За методом *DE* сильний вплив мають: $y_1(RequestQueueTimeMs)$ на $y_4^1(Tfetch)$; $y_2(ResponseQueueTimeMs)$ на $(y_3^1(Treplica); y_4^1(Tfetch))$; $y_3(LogFlushRateAndTimeMs)$ на $(y_3^1(Treplica); y_4^1(Tfetch))$; $y_5(BytesOutPerSec)$ на $y_3^1(Treplica)$. Середній вплив оцінено $y_6(batch\_size\_avg)$ на $(y_1^1(Tproducer); y_2^1(Tleadercommit))$; $y_4(BytesInPerSec)$ на $y_1^1(Tproducer)$; $y_5(BytesOutPerSec)$ на $y_2^1(Tleadercommit)$; $y_1(RequestQueueTimeMs)$ на $y_1^1(Tproducer)$; $y_2(ResponseQueueTimeMs)$ на $y_2^1(Tleadercommit)$; $y_3(LogFlushRateAndTimeMs)$ на $(y_1^1(Tproducer); y_2^1(Tleadercommit))$.

2.2. За методом *DH* сильний вплив мають: $y_1(RequestQueueTimeMs)$ на $y_4^1(Tfetch)$; $y_2(ResponseQueueTimeMs)$ на $(y_2^1(Tleadercommit); y_4^1(Tfetch))$; $y_3(LogFlushRateAndTimeMs)$ на $(y_2^1(Tleadercommit); y_4^1(Tfetch))$, $y_5(BytesOutPerSec)$ на $y_2^1(Tleadercommit)$; $y_6(batch\_size\_avg)$ на $y_2^1(Tleadercommit)$. Середній вплив оцінено для $y_2(ResponseQueueTimeMs)$ на $y_3^1(Treplica)$; $y_5(BytesOutPerSec)$ на $y_3^1(Treplica)$.

2.3. За методом *DJ* сильний вплив мають: $y_1(RequestQueueTimeMs)$ на $y_4^1(Tfetch)$; $y_3(LogFlushRateAndTimeMs)$ на $(y_2^1(Tleadercommit); y_4^1(Tfetch))$; $y_5(BytesOutPerSec)$ на $y_2^1(Tleadercommit)$; $y_6(batch\_size\_avg)$ на $y_2^1(Tleadercommit)$.

3. Другий шар з прихованими змінними $\{y_1^1, y_2^1, y_3^1, y_4^1\}$.

Оцінка впливу є узгодженою між трьома методами. Усі змінні другого прихованого шару мають сильний вплив на вихідну змінну *TL*.

У табл. 1 наведено оцінки, отримані за методами Морріса та Соболя, обчислені на моделі *XGBRegressor*, і на їх основі визначено зважені оцінки *WDE, WDH, WDJ* за формулами (13)–(15), де значення *α, β, γ* становлять 0.2, 0.2 та 0.6 відповідно. Значення параметрів підібрано так, щоб брати до уваги структурні подібності між результатами, отриманими на основі метричних відстаней. У табл. 2 визначено класи зважених оцінок, обчислені за класифікацією (16). Порівняльний аналіз вказує на повну узгодженість оцінок за методами *WDH, WDJ*, які визначили сильний вплив параметрів

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

batch.size, fetch.min.bytes, linger.ms, max.poll.records buffer.memory, max.inflight.requests.per.connection, та сердній вплив параметрів *acks_config, socket.receive.buffer.bytes*.

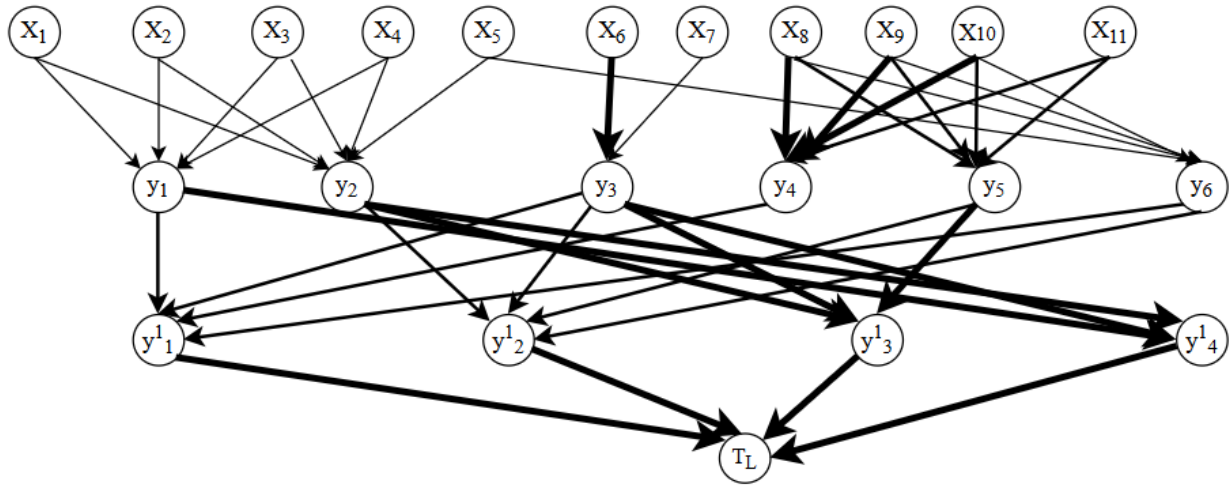

Рис. 4. Оцінювання впливу в мережі Баєса за методом евклідової відстані (*DE*)

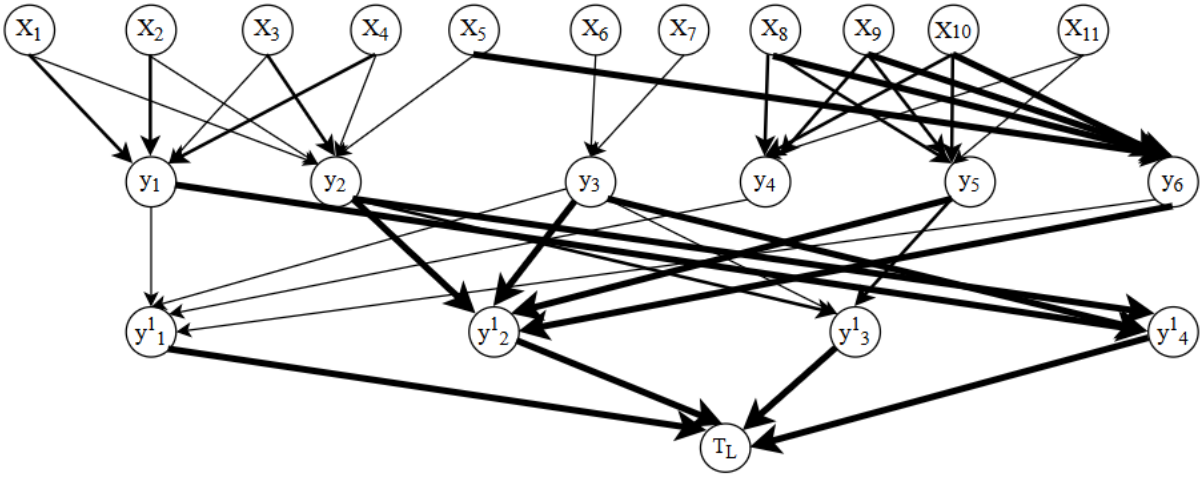

Рис. 5. Оцінювання впливу в мережі Баєса за методом хеллінгерової відстані (*DH*)

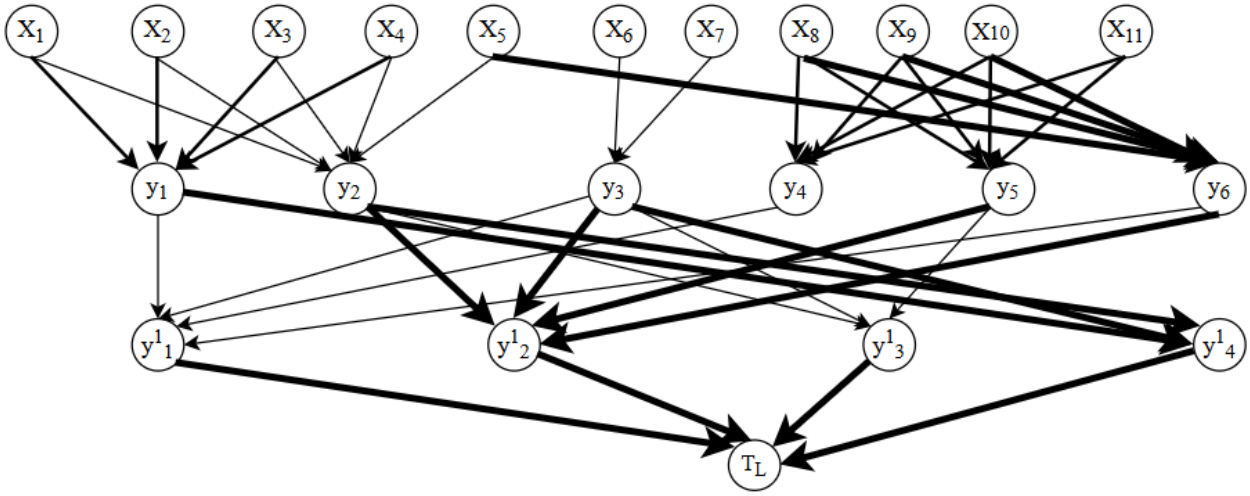

Рис. 6. Оцінювання впливу в мережі Баєса за методом *J*-дивергенції (*DJ*)

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

Класи зважених оцінок за методом *WDE* відрізняються від результатів за методами *WDH, WDJ*, а саме: параметр конфігурації *log.flush.interval.ms* має сильний вплив за методом *WDE* і низький за методами *WDH, WDJ*; параметр *max.poll.records* має низький вплив за методом *WDE* і високий за методами *WDH, WDJ*; параметр *replica.fetch.min.bytes* має сердній вплив за методом *WDE*, але низький за методами *WDH, WDJ*.

Таблиця 1. *Оцінки впливу параметрів конфігурації Kafka-кластера, обчислені на мережі Баєса та моделі XGBRegressor*

Назва вузла мережі Баєса	***DE***	***DH***	***DJ***	**Індекс Морріса, *μi***	**Індекс Соболя, *Si***	**Оцінка за методом, *WDE***	**Оцінка за методом, *WDH***	**Оцінка за методом, *WDJ***
acks_config	0.003	0.042	0.287	0	0	0.002	0.025	0.172
batch.size	0.036	0.127	0.739	1	0.953	0.412	0.467	0.834
batch_size_avg	0.014	0.060	0.369	n/a	n/a	n/a	n/a	n/a
buffer.memory	0.003	0.042	0.287	0	0	0.002	0.025	0.172
BytesInPerSec	0.000	0.016	0.123	n/a	n/a	n/a	n/a	n/a
BytesOutPerSec	0.038	0.073	0.247	n/a	n/a	n/a	n/a	n/a
compression type	0.036	0.127	0.739	n/a	n/a	n/a	n/a	n/a
fetch.min.bytes	0.022	0.063	0.369	0	0	0.013	0.038	0.222
linger.ms	0.036	0.127	0.739	0.2	0.046	0.070	0.124	0.492
log.flush.interval.ms	0.025	0.009	0.000	0.00	0	0.015	0.005	0.000
LogFlushRateAndTimeMs	0.083	0.169	0.739	n/a	n/a	n/a	n/a	n/a
max.inflight.requests.per.connection	0.003	0.042	0.287	0	0	0.002	0.025	0.172
max.poll.records	0.002	0.060	0.410	0	0	0.001	0.036	0.246
replica.fetch.min.bytes	0.007	0.000	0.000	0	0	0.004	0.000	0.000
RequestQueueTimeMs	0.037	0.089	0.492	0	0	0.022	0.054	0.295
ResponseQueueTimeMs	0.073	0.146	0.616	0	0	0.044	0.087	0.370
socket.receive.buffer.bytes	0.003	0.042	0.287	0	0	0.002	0.025	0.172
Tfetch	1.000	1.000	1.000	n/a	n/a	n/a	n/a	n/a
Tleadercommit	1.000	1.000	1.000	n/a	n/a	n/a	n/a	n/a
Tproducer	0.995	0.997	1.000	n/a	n/a	n/a	n/a	n/a
Treplica	1.000	1.000	1.000	n/a	n/a	n/a	n/a	n/a

Познака " n/a " означає, що параметри прихованих шарів баєсівської мережі не додані до моделі *XGBRegressor*, оскільки значення цих параметрів не задаються під час конфігурації *Kafka*-кластера, а формуються як результат впливу визначеної конфігурації.

На рис. 7 зображено вплив кожного параметра конфігурації з множини X на затримку *Kafka*-кластера, обчислений за методами на основі метричних відстаней *DE, DH, DJ*, прямого впливу параметрів (за допомогою індексів Морріса μ_i) та з огляду на взаємодію між параметрами (за допомогою індексів Соболя S_i). Різна висота стовпців ілюструє неузгодженості в оцінках, отриманих відповідно до зазначених методів. Лінії є нормалізованими оцінками впливу параметрів, побудованими на основі зважених метрик *WDE, WDH, WDJ*. Близькість ліній для методів *WDH, WDJ* вказує на узгодженість оцінок.

За правилом класифікації "більшість голосів", відповідно до виразу (18), отримано узгоджені класи оцінки впливу $W(X)$, які після надання числових значень були використані для ініціалізації матриці ваг нейронної мережі прямого поширення, що прогнозує наскрізну затримку в *Kafka*-кластері залежно від обраної конфігурації. Для визначення ефективності запропонованого підходу оцінювання впливу $W(X)$ на навчання нейронної мережі також застосовано під час ініціалізації матриці ваг за оцінками, отриманими

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

на основі запропонованого підходу (13)–(15), і в процесі ініціалізації матриці ваг нейронної мережі прямого поширення за методом *Xavier*, який упроваджується за замовчуванням у нейронних мережах прямого поширення.

Таблиця 2. *Класи оцінок впливу параметрів, визначені за запропонованою класифікацією та правилом "більшості голосів"*

Назва вузла мережі Баєса	***WDJ***	***WDH***	***WDE***	**Клас оцінки, *WDJ***	**Клас оцінки, *WDJ***	**Клас оцінки, *WDJ***	**Клас оцінки за правилом "більшості голосів"**
acks_config	0.172	0.025	0.002	M	M	M	M
batch.size	0.834	0.467	0.412	H	H	H	H
batch_size_avg	0.222	0.036	0.008	M	M	M	M
buffer.memory	0.172	0.025	0.002	M	M	M	M
fetch.min.bytes	0.222	0.038	0.013	H	H	H	H
linger.ms	0.492	0.124	0.070	H	H	H	H
log.flush.interval.ms	0.000	0.005	0.015	L	L	H	L
max.inflight.requests.per.connection	0.172	0.025	0.002	M	M	M	M
max.poll.records	0.246	0.036	0.001	H	H	L	H
replica.fetch.min.bytes	0.000	0.000	0.004	L	L	M	L
socket.receive.buffer.bytes	0.172	0.025	0.002	M	M	M	M

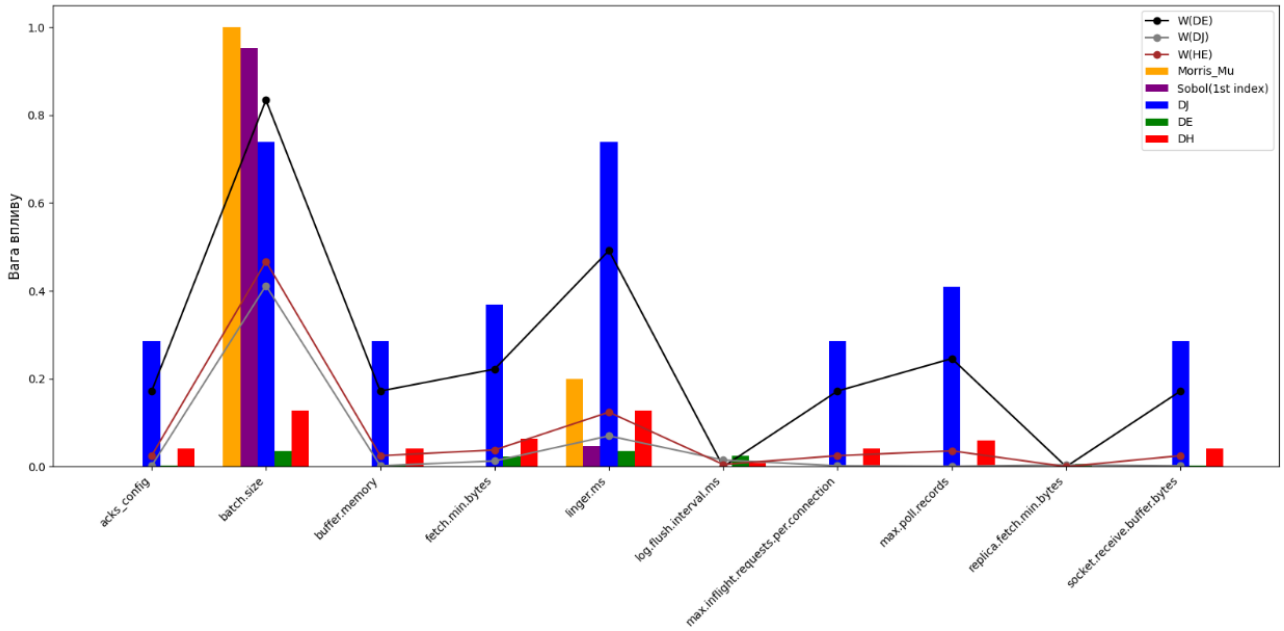

Рис. 7. Порівняння оцінок впливу параметрів конфігурації на затримку *Kafka*-кластера за різними методами

На рис. 8 згладжена функція витрат, отримана на основі критерію мінімізації середньоквадратичної похибки, демонструє найменші значення протягом перших 40 ітерацій навчання нейронної мережі, яка була ініціалізована вагами узгоджених класів оцінки впливу *W(X)*. Це свідчить про ефективність запропонованого підходу, оскільки що швидше мережа досягає мінімуму функції витрат, то ефективнішою є ініціалізація ваг.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)

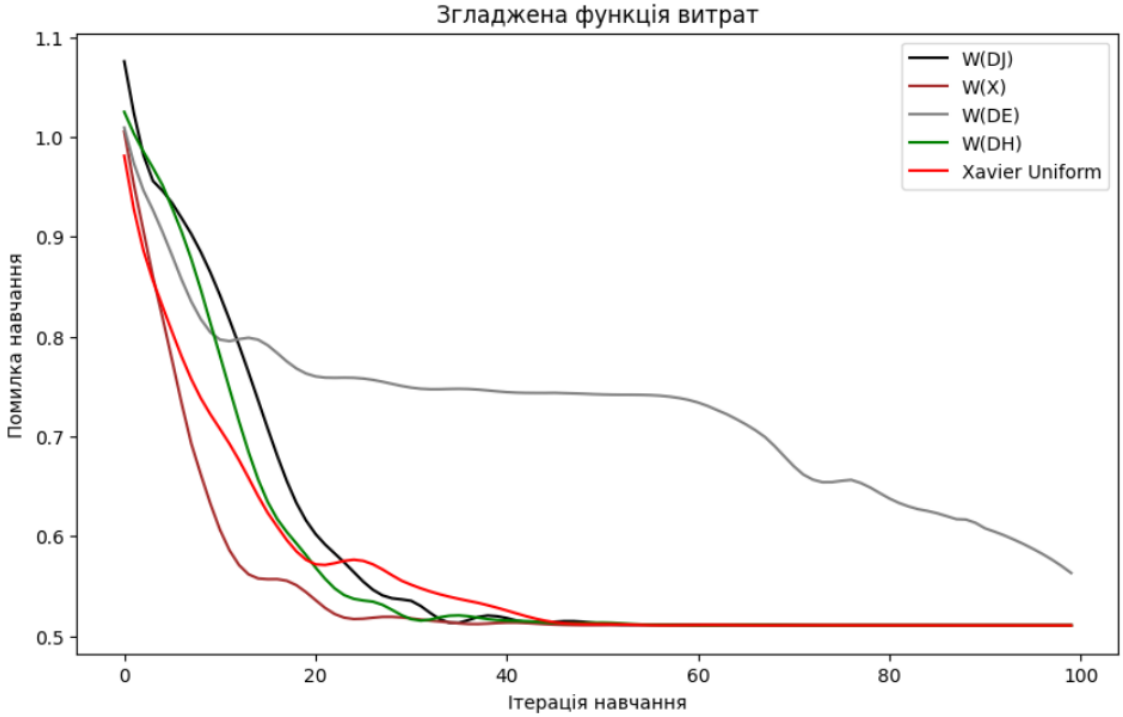

Рис. 8. Згладжена функція витрат, яка отримана на основі критерію мінімізації середньоквадратичної похибки в процесі навчання нейронної мережі

Висновки й перспективи подальшого дослідження

У статті запропоновано новий підхід до аналізу чутливості, який поєднує оцінку прямого впливу параметрів (за допомогою індексу Морріса μ_i), взяття до уваги взаємодій між параметрами (за допомогою індексу Соболя S_i) і причинно-наслідкові зв'язки між параметрами, що оцінюються за допомогою методів, застосовуваних у межах причинно-орієнтованих моделей, зокрема в баєсівських мережах (евклідова відстань, хеллінгерова відстань, *J*-дивергенція).

Запропонований підхід формалізовано у вигляді зваженої комбінації вищезазначених методів. Для його апробації розроблено класифікацію оцінок чутливості, а для узагальнення результатів – правило "більшості голосів".

Порівняльний аналіз оцінок впливу параметрів конфігурації *Kafka*-кластера на наскрізну затримку, проведений за допомогою метрик евклідової відстані, хеллінгерової відстані та *J*-дивергенції на моделі баєсівської мережі, показав узгодженість у визначенні сильного впливу змінних другого прихованого шару. Водночас виявлено розбіжності щодо оцінок спостережуваних змінних та змінних першого прихованого шару, де вплив здебільшого оцінювався як середній. Це свідчить про різницю в чутливості зазначених методів і потребує їх уточнення або адаптації.

Запропонований підхід на основі зважених метрик продемонстрував абсолютну узгодженість оцінок за методами *WDH* (зважена хеллінгерова відстань) і *WDJ* (зважена *J*-дивергенція), що підтверджує перевагу комбінованого підходу над окремими метриками *DE, DH, DJ*.

Для оцінювання ефективності підходу проведено порівняльне навчання нейронної мережі з різними стратегіями ініціалізації ваг: на основі оцінок *W(DE), W(DH), W(DJ)*; за стандартним методом ініціалізації *Xavier*; за узгодженими класами оцінки впливу *W(X)*. Аналіз згладженої функції витрат, побудованої за критерієм мінімізації середньоквадратичної похибки, продемонстрував, що найменших значень функція досягає впродовж перших 40 ітерацій саме в моделі, ініціалізованій вагами з узгоджених оцінок *W(X)*. Це свідчить про покращену збіжність мережі та більш якісну ініціалізацію ваг.

Новизна підходу полягає в інтеграції переваг причинно-орієнтованих і дисперсійно-оцінних методів у межах єдиної зваженої метрики чутливості.

Практична цінність підходу полягає в тому, що його застосування під час аналізу чутливості або

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

ініціалізації матриці ваг нейронної мережі дає змогу підвищити точність оцінок впливу параметрів, покращити збіжність моделі та скоротити час її навчання.

Подальші дослідження будуть спрямовані на впровадження метрик структурної подібності для аналізу чутливості та формалізацію критерію узгодженості оцінок, отриманих різними методами.

Список літератури

1. Solovei O., Honcharenko T., Fesan A. "Technologies to manager big data of urban building projects", *Management of Development of Complex Systems*, No. 60, P.121–128, 2024. DOI: 10.32347/2412-9933.2024.60.121-128
2. Narkhede, M. V., Bartakke, P. P., Sutaone, M. S. "A review on weight initialization strategies for neural networks", *Artificial intelligence review*, No. 55(1), P. 291–322. 2022. DOI: 10.1007/s10462-021-10033-z
3. Wong, K., Dornberger, R., Hanne, T. "An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks", *Evolutionary Intelligence*, No. 17(3), P.2081-2089, 2024. DOI: 10.1007/s12065-022-00795-y
4. Brand J.E., Zhou X., Xie Y. "Recent developments in causal inference and machine learning", *Annual Review of Sociology*, No. 49 (1), P. 81–110. 2023. DOI: 10.1146/annurev-soc-030420-015345
5. Chumachenko K., Iosifidis A., and Gabbouj M. "Feedforward neural networks initialization based on discriminant learning", *Neural Networks*, No. 146, P. 220–229. 2022. DOI: 10.1016/j.neunet.2021.11.020
6. Zhao J., Schäfer F., and Anandkumar A. "Zero initialization: Initializing neural networks with only zeros and ones", *Published in Transactions on Machine Learning Research*, № 11. 2021. URL: arXiv preprint arXiv:2110.12661
7. Pan Y., Wang C., Wu Z., Wang Q., Zhang M., and Xu Z. "IDInit: A Universal and Stable Initialization Method for Neural Network Training", 2025. URL: arXiv preprint arXiv:2503.04626
8. Zhu C., Ni R., Xu Z., Kong K., Huang W. R., and Goldstein T. "Gradinit: Learning to initialize neural networks for stable and efficient training", *Advances in Neural Information Processing Systems*, No. 34, P.16410–16422. 2021. DOI: https://doi.org/10.3390/app15042008
9. Nouri A., van Treeck C., & Frisch J. "Sensitivity Assessment of Building Energy Performance Simulations Using MARS Meta-Modeling in Combination with Sobol'Method", *Energies*, No. 17(3), 695 p. 2024. DOI: https://doi.org/10.3390/en17030695
10. Sadeghi Z., Matwin S. "A Review of Global Sensitivity Analysis Methods and a comparative case study on Digit Classification". 2024. URL: arXiv preprint arXiv:2406.16975
11. Mazo G., Tournier L. "An inference method for global sensitivity analysis", *Technometrics*, No. 67(2), P. 270-282. 2025.
12. Kozniewski M., Kolendo Ł., Chmur S., Ksepko M. "Impact of Parameters and Tree Stand Features on Accuracy of Watershed-Based Individual Tree Crown Detection Method Using ALS Data in Coniferous Forests from North-Eastern Poland", *Remote Sensing*, No. 17(4), 575 p. 2025. DOI: https://doi.org/10.3390/rs17040575
13. Kaddoura M., Majeau-Bettez G., Amor B., & Margni M. "Global sensitivity analysis reduces data collection efforts in LCA: A comparison between two additive manufacturing technologies", *Science of the Total Environment*, No. 975, 179269 p. 2025. DOI: https://doi.org/10.1016/j.scitotenv.2025.179269
14. Raptis T. P., Passarella A. "A survey on networked data streaming with apache kafka", *IEEE Access*, No. 11, P. 85333-85350. 2023. DOI: 10.1109/ACCESS.2023.3303810
15. Kafka Producer Configuration Reference for Confluent Platform. URL: https://docs.confluent.io/platform/current/installation/configuration/producer-configs.html
16. Wang J., Chen Z., Song Y., Liu Y., He J., Ma S. "Data-Driven Dynamic Bayesian Network Model for Safety Resilience Evaluation of Prefabricated Building Construction", *Buildings*, No. 14, 570 p. 2024. DOI: 10.3390/buildings14030570
17. Echabarri S., Do P, Vu H., Bornand B. "Machine learning and Bayesian optimization for performance prediction of proton-exchange membrane fuel cells", *Energy and AI*, No. 17, 100380 p. DOI: https://doi.org/10.1016/j.egyai.2024.100380

References

1. Solovei, O., Honcharenko, T., Fesan, A. (2024), "Technologies to manager big data of urban building projects", *Management of Development of Complex Systems*, No. 60, P.121–128, DOI: 10.32347/2412-9933.2024.60.121-128

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2025. No. 3 (33)*

2. Narkhede, M. V., Bartakke, P. P., Sutaone, M. S. (2022), "A review on weight initialization strategies for neural networks", *Artificial intelligence review*, No. 55(1), P. 291–322. DOI: 10.1007/s10462-021-10033-z
3. Wong, K., Dornberger, R., Hanne, T. (2024), "An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks", *Evolutionary Intelligence*, No. 17(3), P. 2081–2089, DOI: 10.1007/s12065-022-00795-y
4. Brand, J.E., Zhou, X., Xie, Y. (2023), "Recent developments in causal inference and machine learning", *Annual Review of Sociology*, No. 49 (1), P. 81–110. DOI: 10.1146/annurev-soc-030420-015345
5. Chumachenko, K., Iosifidis, A., Gabbouj, M. (2022), "Feedforward neural networks initialization based on discriminant learning", *Neural Networks*, No. 146, P. 220–229. DOI: 10.1016/j.neunet.2021.11.020
6. Zhao, J., Schäfer, F., Anandkumar, A. (2021), "Zero initialization: Initializing neural networks with only zeros and ones", *Published in Transactions on Machine Learning Research*, № 11. available at: arXiv preprint arXiv:2110.12661
7. Pan, Y., Wang, C., Wu, Z., Wang, Q., Zhang, M., Xu, Z. (2025), "IDInit: A Universal and Stable Initialization Method for Neural Network Training", available at: arXiv preprint arXiv:2503.04626
8. Zhu, C., Ni, R., Xu, Z., Kong, K., Huang, W. R., Goldstein, T. (2021), "Gradinit: Learning to initialize neural networks for stable and efficient training", *Advances in Neural Information Processing Systems*, No. 34, P. 16410–16422. DOI: https://doi.org/10.3390/app15042008
9. Nouri, A., van Treeck, C., Frisch, J. (2024), "Sensitivity Assessment of Building Energy Performance Simulations Using MARS Meta-Modeling in Combination with Sobol'Method", *Energies*, No. 17(3), 695 p. DOI: https://doi.org/10.3390/en17030695
10. Sadeghi, Z., Matwin, S. (2024), "A Review of Global Sensitivity Analysis Methods and a comparative case study on Digit Classification". URL: arXiv preprint arXiv:2406.16975
11. Mazo, G., Tournier, L. (2025), "An inference method for global sensitivity analysis", *Technometrics*, No. 67(2), P. 270–282.
12. Kozniewski, M., Kolendo, Ł., Chmur, S., Ksepko, M. (2025), "Impact of Parameters and Tree Stand Features on Accuracy of Watershed-Based Individual Tree Crown Detection Method Using ALS Data in Coniferous Forests from North-Eastern Poland", *Remote Sensing*, No. 17(4), 575 p. DOI: https://doi.org/10.3390/rs17040575
13. Kaddoura, M., Majeau-Bettez, G., Amor, B., Margni, M. (2025), "Global sensitivity analysis reduces data collection efforts in LCA: A comparison between two additive manufacturing technologies", *Science of the Total Environment*, No. 975, 179269 p. DOI: https://doi.org/10.1016/j.scitotenv.2025.179269
14. Raptis, T. P., Passarella, A. (2023), "A survey on networked data streaming with apache kafka", *IEEE Access*, No. 11, P. 85333–85350. DOI: 10.1109/ACCESS.2023.3303810
15. "Kafka Producer Configuration Reference for Confluent Platform". URL: https://docs.confluent.io/platform/current/installation/configuration/producer-configs.html.
16. Wang, J., Chen, Z., Song, Y., Liu, Y., He, J., Ma S. (2024), "Data-Driven Dynamic Bayesian Network Model for Safety Resilience Evaluation of Prefabricated Building Construction", *Buildings*, No. 14, 570 p. DOI: 10.3390/buildings14030570
17. Echabarri, S., Do, P, Vu, H., Bornand, B. (2024), "Machine learning and Bayesian optimization for performance prediction of proton-exchange membrane fuel cells", *Energy and AI*, No. 17, 100380 p. DOI: https://doi.org/10.1016/j.egyai.2024.100380

Надійшла (Received) 28.05.2025

Відомості про авторів / About the Authors

Соловей Ольга Леонідівна – кандидат технічних наук, Київський національний університет будівництва і архітектури, доцент кафедри інформаційних технологій проєктування та прикладної математики, Київ, Україна; e-mail: solovey.ol@knuba.edu.ua; ORCID ID: https://orcid.org/0000-0001-8774-7246

Solovei Olga – PhD (Technical Sciences), Kyiv National University of Construction and Architecture, Associate Professor at the Department of Information Technology Design and Applied Mathematic, Kyiv, Ukraine.

Сучасний стан наукових досліджень та технологій в промисловості. 2025. № 3 (33)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

A WEIGHTED SENSITIVITY METRIC FOR PREDICTING LATENCY IN A KAFKA CLUSTER

The subject of this article is sensitivity analysis methods used to assess how variations in input parameters affect the output results of a model or system. The aim of the study is to develop a new approach to sensitivity analysis that combines classical parameter impact assessment methods (Morris and Sobol methods) with metrics that capture structural changes in the distribution of output data (Euclidean distance, Hellinger distance, Jensen divergence). This approach allows for evaluating the influence of a parameter not only in terms of the amplitude of its effect, but also in terms of changes in the shape and structure of the probability distribution of the results. To achieve this objective, the article addresses the following tasks: formal definition of a new sensitivity analysis approach; development of a Bayesian network for modeling end-to-end latency in a Kafka cluster; performing sensitivity analysis using the proposed approach; and conducting an experimental study using the calculated parameter influence weights to initialize the weight matrix of a neural network that predicts Kafka cluster latency based on selected configuration parameters. To accomplish these tasks, the study applied methods from the theory of experiments, Euclidean geometry, statistical distribution theory, information theory, machine learning, Bayesian statistics, and graph theory. Results: To evaluate the effectiveness of the proposed approach, comparative training of a neural network was conducted using various weight initialization strategies. Analysis of the loss function, constructed using the mean squared error minimization criterion, showed that the lowest values were achieved by the model initialized with weights obtained using the proposed parameter influence estimation approach. Conclusions: The study proposes a novel approach to sensitivity analysis. The innovation lies in integrating the strengths of both causal-oriented and variance-based methods within a unified weighted sensitivity metric. The practical value of this approach is that its application in sensitivity analysis or neural network weight initialization improves the accuracy of parameter impact assessment, enhances model convergence, and reduces training time.

Keywords: Morris method; Sobol index; Euclidean distance; Hellinger distance; Jensen divergence; sensitivity analysis.

Бібліографічні описи / Bibliographic descriptions

Соловей О.Л. Зважена метрика чутливості для прогнозування затримки в KAFKA-кластері. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 3 (33). С. 152–165. DOI: https://doi.org/10.30837/2522-9818.2025.3.152

Solovei, O. (2025), " A weighted sensitivity metric for predicting latency in a KAFKA cluster", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (33), P. 152–165. DOI: https://doi.org/10.30837/2522-9818.2025.3.152