https://doi.org/10.31891/2307-5732-2026-361-72
УДК 004.42

РИБАЛЬЧЕНКО ОЛЕКСІЙ
Державний університет інформаційно-комунікаційних технологій
https://orcid.org/0009-0004-5261-3391
e-mail: cerateg@gmail.com

МОДЕЛЮВАННЯ ПОВЕДІНКОВИХ РИЗИКІВ КОРИСТУВАЧІВ КОРПОРАТИВНИХ БАЗ ДАНИХ ІЗ ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

У статті запропоновано модель виявлення аномальних тенденцій у діях користувачів корпоративних баз даних на основі гібридної архітектури LSTM-Autoencoder. Підхід поєднує аналіз структурних і часових характеристик поведінки, що забезпечує ефективне виявлення як контекстних, так і послідовних відхилень. Валідацію моделі здійснено на реальних ERP-даних SALT обсягом понад 2,3 млн записів, агрегованих у часові вікна з 87 ознаками. Навчання проводилося з використанням оптимізатора Adam, що забезпечило стабілізацію функції втрат на рівні MSE = 0,0023. Динамічна адаптація порогового значення дозволила знизити частку хибних спрацювань до 3,1%. Модель продемонструвала високі показники якості (F1 = 0,938, AUC = 0,972), перевершивши класичні методи Isolation Forest та One-Class SVM. Експериментальні результати підтверджують здатність підходу виявляти складні поведінкові аномалії, що створює підґрунтя для реалізації інтелектуальних систем оцінки поведінкових ризиків корпоративних баз даних у режимі реального часу.

***Ключові слова**: помилка реконструкції, аномальні поведінкові тенденції, динамічний поріг адаптації, SQL-запит, хибні тривоги, проміжне програмне забезпечення.*

RYBALCHENKO OLEKSII
State University of Information and Communication Technologies

MODELING BEHAVIORAL RISKS OF CORPORATE DATABASE USERS USING MACHINE LEARNING METHODS

A model for anomaly detection in user behavior within corporate databases was presented, based on a hybrid LSTM-Autoencoder architecture. It is emphasized that the proposed approach integrates both structural and temporal behavioral factors, allowing effective detection of contextual and sequential deviations. For validation, real ERP-oriented data from the SALT (Sales Autocompletion Linked Tables) dataset were used, comprising more than 2.3 million records reflecting transactions, clients, and logistics processes. The data were aggregated into temporal windows of length m = 20 queries with 87 features, formalizing the dynamics of user activity. Training was conducted on a Tesla T4 GPU (16 GB) using the Adam optimizer with a learning rate of $1e^{-3}$, batch size 128, and 50 epochs, during which the loss function stabilized at MSE = 0.0023. The threshold value dynamically adapted to the current risk distribution, reducing false positives to 3.1%. The mean reconstruction error for normal windows was Lrec = 0.0017, while for anomalous windows it was 0.0079, providing more than a fourfold separation between clusters. The model achieved Precision = 0.946, Recall = 0.931, F1-score = 0.938, and AUC = 0.972, outperforming classical methods such as Isolation Forest and One-Class SVM by 7–15%. The results show that that the dynamic threshold mechanism θt enables the system to adapt its sensitivity to varying workloads, maintaining balance between accuracy and robustness. Experimental results confirm the model's ability to distinguish between structural and behavioral anomalies, including sudden shifts in query types, actions inconsistent with user roles, and unusual geographic sources of access. Thus, the proposed method forms the basis for an intelligent real-time behavioral risk assessment system for corporate databases, capable of integration into existing DBMS environments through a middleware interface without compromising performance.

***Keywords:** reconstruction error, anomalous behavioral patterns, dynamic adaptation threshold, SQL query, false alarms, middleware.*

Стаття надійшла до редакції / Received 20.12.2025
Прийнята до друку / Accepted 11.01.2026
Опубліковано / Published 29.01.2026

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Враховуючи ріст обсягів даних та кількості користувачів баз даних, варто звернути увагу на також стале підвищення ризику, пов'язаного з дедалі інтенсивнішим та масштабнішим використанням баз даних. Цей ризик полягає у проявах аномальної поведінки з боку користувачів БД, що сприяє виникненню ситуацій порушення цілісності даних та несанкціонованого доступу. Класичні засоби захисту від такої динаміки, а саме традиційні системи контролю доступу, журнали аудиту та правила безпеки пов'язані з реагуванням на подібні події постфактум, не забезпечуючи своєчасного виявлення прихованих загроз.

В таких умовах зростає актуальність застосування методів машинного навчання для моделювання поведінкових ризиків. Особливої значущості набувають підходи, які здатні одночасно враховувати структурні та часові аспекти поведінки користувачів. З одного боку, окремі SQL-запити можуть мати нетипові структуру або параметри, наприклад, незвичайний обсяг даних або звернення до конфіденційних таблиць. З іншого – необхідно врахувати, що звичайні на перший погляд запити можуть характеризуватися підозрілістю з точки зору послідовності їхнього виконання, що також вказує на тимчасове відхилення з розвитком поступового характеру.

Оскільки акцент в поточному дослідженні припадає на керування корпоративними БД, важливо врахувати, що вони характеризуються високими концентрацією конфіденційної інформації та кількістю рівнів доступу зі складною структурою взаємодії користувачів. У цьому контексті кожна операція може мати стратегічні наслідки, від порушення бізнес-логіки підприємства до прямої загрози його безпеки. Відтак, моделювання поведінкових ризиків у контексті корпоративних БД вимагає не лише аналізу запитів в якості

технічних операцій, але також і розуміння їхнього організаційного та контекстного сенсу. У цих системах має враховуватися роль користувача, ієрархія прав доступу, часові закономірності активності користувачів та особливості робочих процесів.

У таких умовах існує потреба у розробці архітектури, яка б була побудована таким чином, щоб забезпечити безперервне надходження даних з логів, їхню попередню обробку (препроцесинг), кількісне визначення показника ризику та інтеграцію результатів у робочий потік БД без суттєвого впливу на продуктивність системи загалом.

Аналіз останніх досліджень і публікацій

У межах дослідження [1] була запропонована система виявлення аномалій UCAD (Unsupervised Contextual Anomaly Detection for Database Systems, тобто система контекстуального виявлення аномалій на базі навчання без підкріплення для систем керування БД). Ця система, на відміну від методів, що орієнтуються на виявлення відомих атак та значних відхилень від нормальної поведінки, передбачає виявлення прихованих аномальних закономірностей, які можуть бути схожими зі звичайними операціями, але не відповідають їм на рівні контекстної мотивації. Для вирішення цієї задачі була розроблена трансформерна модель Trans-DAS, що передбачає використання спеціалізованого шару векторних представлень для кодування семантичних характеристик операцій, ігноруючи інформацію про їхній порядок, та використання механізму маскування, що дозволяє фіксувати семантичну інформацію з урахуванням як попередніх, так і подальших операцій у послідовності. До того ж, було введено новий навчальний критерій, спрямований на підвищення відмінностей між векторними представленнями семантичних характеристик операцій. Для ефективного застосування моделі у системі UCAD були розроблені два модулі: модуль попередньої обробки даних, який видаляє шумові дані; та модуль виявлення аномалій, в якому навчені семантичні закономірності використовуються для порівняння мотивованості виконання певних операцій користувачами. Результати експериментів, проведених на реальних даних з різними налаштуваннями (мінливі параметри та гібридні набори даних) показали, що Оцінка F1 в UCAD становить 0.94 у двох сценаріях, що значно перевершує базові методи.

Наступне дослідження [2] зосереджувалося на розробці системи виявлення аномалій у логах SQL-запитів, з використанням методів ML для аналізу даних медичної бази даних. В рамках роботи було створено синтетичний лог файл, що моделює дії 700 лікарів, що були по сумісництву і користувачами БД (20,000 пацієнтів) та виконували запити до неї. Основна задача дослідження полягала у застосуванні алгоритмів кластеризації, таких як K-Means та DBSCAN (Density-based spatial clustering of applications with noise), для виявлення таких поведінкових тенденцій, як багаторазові запити від різних лікарів до одного пацієнта, повторювані запити за один день та надто велика кількість запитів від одного лікаря. Також була застосована техніка "Elbow Method" для вибору оптимальної кількості кластерів, що дозволило підвищити точність кластеризації. Результати Means кластеризації та DBSCAN показали виявлення схожих аномалій, що підтверджує надійність обраного методу. Також у роботі був передбачений огляд детального процесу створення БД та лог-файлу, що імітує реальні дії лікарів при роботі з пацієнтами.

Однією з ключових проблем в області виявлення аномалій виявляється пошук балансу між чутливістю, тобто здатністю правильно виявляти аномалії, та специфічністю, тобто здатністю уникати хибних тривог. Для оптимізації цього балансу існує потреба у проведенні тонкого налаштування алгоритму виявлення з урахуванням особливостей конкретної області. На цьому тлі, враховуючи надбання роботи [3], значної актуальності набуває поєднання таких моделей ML як LSTM (long short-term memory – модель довгої короткострокової пам'яті) та Autoencoder (Автокодувальник). Запропонована гібридна модель застосовується для виявлення аномалій у даних якості повітря у приміщеннях. Модель передбачає поєднання сильних сторін довгострокових закономірностей, враховуваних LSTM, та здатності Автокодувальника виявляти аномальні дані, виходячи з помилок відновлення (реконструкції). В рамках роботи було використано набір даних про CO_2, зібраний за допомогою пристрою SKOMOBO. Модель передбачає проведення аналізу часових рядів з даними про CO_2 та застосування LSTM для врахування залежностей між часовими точками, і Автокодувальника для визначення порогового значення на базі помилок реконструкції. При перевищенні даними тестового набору порогу вони класифікуються як аномальні. Результати експерименту показали високу точність моделі в контексті виявлення аномалій 99,50%.

У дослідженні [4] було запропоновано інший метод виявлення аномалій у розподілених БД на основі багатовимірних логів (MultiLog), а саме – підхід обробки логів, зібраних з декількох вузлів. Для цієї мети було створено відповідний набір даних обсягом 216 ГБ и 900 млн записів, що містив 11 типів системних та внутрішніх збоїв, отриманих з Apache IoTDB. Було зазначено, що такі моделі, як RobustLog, LogAnomaly, PLELog, виявляються менш ефективними в контексті роботи з багатовузловими логами та характеризуються високим рівнем хибних тривог. Запропонований метод суміщає два рівні обробки, а саме окремої оцінки (вилучення послідовної, кількісної та семантичної інформації за допомогою LSTM та механізму самоуваги (self-attention)) та кластерного класифікатора (Автокодувальник та мета-класифікатора для об'єднання вірогідностей всіх вузлів). Експерименти показують, що MultiLog підвищує точність виявлення аномалій на 12% у кластерних сценаріях та на 16% при аналізі окремих вузлів.

У той же час, фокус роботи [5] зміщується на застосування гібридної моделі LSTM-Автокодувальника для виявлення аномалій в роботі електричних двигунів. Іншими словами, для виявлення змін у вібраційних закономірностях по трьом осям: осьовій (X), радіальній (Y) та тангенційній (Z), оскільки ці вібрації слугують в якості індикаторів потенційних несправностей та збоїв в роботі двигуна. Обидві моделі були створені з

використанням Python та фреймворку ML TensorFlow. Моделі були навчені та оцінені в межах однакового набору даних, з акцентом на 3 ключові метрики продуктивності, а саме, час навчання, функцію втрат та аномалії, виміряні за допомогою середньоквадратичної помилки (Mean squared error – MSE). Результати демонструють, що гібридна модель перевершує звичайний Автокодувальник за обома мітками, тобто значеннями втрат та аномалій MSW. Незважаючи на це, зазначається, що така архітектура характеризується вищою вимогливістю з точки зору витрат часу на навчання, з огляду на додаткову складність, пов'язану з використання шарів LSTM.

Схоже за принципом проведення дослідження [6] було вже спрямоване на виявлення аномалій в даних про вібрацію вітрових турбін. Пропонується застосування LSTM-Автокодувальника для виявлення аномалій без нагляду. Вібраційні дані, зібрані під час роботи турбін з акселерометрами для їхнього вимірювання у декількох напрямках, були попередньо оброблені з допомогою комбінації розкладання вейвлет-пактів (Wavelet packet decomposition – WPT) та високочастотної фільтрації (High-pass filter – HPF). Така процедура проводилася для підкреслення високочастотних компонентів даних, що вказують на аномальні тенденції в роботі. Далі для зменшення розмірності даних був застосований метод головних компонент (Principal component analysis – PCA). Ефективність моделі була оцінена шляхом порівняння з іншими методами навчання без нагляду, включаючи алгоритм випадкового лісу (Isolation Forest). Згідно з експериментальними результатами, модель LSTM-Автокодувальник у сумісництві з методом попередньої обробки WPT-HPF-PCA досягає 97 % продуктивності у виявленні аномалій.

Відтак, варто зробити висновок, що використання поєднання моделей LSTM та Автокодувальника для виявлення аномалій характеризується перспективністю у контексті застосування для керування базами даних. По-перше, бази даних часто містять послідовні дані, такі як часові мітки, транзакції або лог-файли, де існує потреба у врахуванні не тільки поточних значень, але й історичної інформації. LSTM-шари ефективно фіксують такі часові залежності, що дозволяє моделювати нормальну поведінку систем та виділяти аномальні відхилення виходячи з часових тенденцій. По-друге, подібна модель характеризується високою гнучкістю, оскільки навчається на нормальних даних та може виявляти аномалії на нових даних, не потребуючи попередньої розмітки.

Проте, беручи до уваги вище зазначену наукову документацію, питання, пов'язане з розробкою інтелектуального підходу моделювання поведінкових ризиків користувачів корпоративних баз даних, все ще залишається недостатньо дослідженим та потребує подальшого опрацювання.

Формулювання цілей статті

Метою роботи є розробка інтелектуального підходу моделювання поведінкових ризиків користувачів корпоративних баз даних з використання гібридної моделі на базі LSTM та автокодувальника з урахуванням контекстних, часових та структурних залежностей запитів і формування інтегрального показника ризику в динаміці.

Виклад основного матеріалу

Враховуючи теоретичну базу та дослідження [3-6], для побудови моделі, здатної прогнозувати вірогідність аномальної тенденції у даних, кожна дія користувача представляється у вигляді вектора ознак. Відтак, припускається, що множина операцій користувача у часі визначається як:

$$X = \{x_1, x_2, \dots, x_t, \dots, x_T\}, \quad (1)$$

де $x_t \in \mathbb{R}^n$ відповідає вектору ознак, що описує одну операцію користувача у час t. Набір ознак формується виходячи з характеристик конкретного SQL-запиту та контексту його виконання, включаючи тип операції (SELECT, UPDATE, DELETE, INSERT тощо); цільову таблицю або схему даних; часову мітку виконання; обсяг повернуваних або змінюваних даних; унікальний ідентифікатор користувача та його роль у системі; мережеві параметри, тобто IP-адресу та геологацію з'єднання; частоту попередніх звернень та часові інтервали між запитами.

Часова мітка слугує не тільки в якості ідентифікатору моменту виконання операції, але й забезпечує можливість аналізу часової послідовності дій для виявлення прихованих поведінкових закономірностей, наприклад постійного зростання обсягу завантаження даних, або контраст між звичайним їхнім читанням та масовою зміною.

Для навчання моделі з наглядом вводиться бінарна змінна:

$$a_t \in \{0,1\}, \quad (2)$$

де $a_t = 0$ відповідає нормальній операції користувача, а $a_t = 1$ – операції, пов'язаній з ризиком порушення регламентацій безпеки або правил на заборону несанкціонованого доступу. У контексті навчання без нагляду, значення at не використовується безпосередньо, та модель навчається виходячи з розподілу ознак нормальних тенденцій у даних, таким чином зменшуючи помилку реконструкції або прогнозування [7].

Таким чином, вхідні дані для моделі можуть бути представлені у вигляді впорядкованої послідовності векторів операцій користувача:

$$X = \{(x_t, t)\}_{t=1}^{T}, \quad (3)$$

Архітектура Автокодувальника в контексті аналізу поведінкових ризиків побудована за принципом симетричної нейромережевої моделі, що складається з вхідного, прихованого та вихідного шарів. Нехай вхідний вектор $x_t \in \mathbb{R}$ являє собою операцію користувача в момент часу t, де n є кількістю ознак, що описують запит [8]. Ця підмодель виконує два взаємопов'язаних перетворення, а саме кодування та декодування. Перша процедура виражається так:

$$z_t = f_{\text{enc}}(x_t) = \sigma(W_{\text{enc}} x_t + b_{\text{enc}}), \quad (4)$$

де $z_t \in \mathbb{R}^k$ позначає вектор у прихованому просторі ознак, що відображає компактне числове

представлення поведінкових ознак.

Декодування, у той же час, формулюється таким чином:

$$\hat{x}_t = f_{\text{dec}}(z_t) = \sigma(W_{\text{dec}} z_t + b_{\text{dec}}), \quad (5)$$

де $\hat{x}_t$ є реконструйованим вектором ознак, який має якомога точніше відповідати вихідному x_t.

Функція втрат Автокодувальника задається як середньоквадратичне відхилення між вихідним вектором ознак x_t та його реконструйованим представленням $\hat{x}_t$:

$$L_{\text{rec}} = \|x_t - \hat{x}_t\|_2^2 \quad (6)$$

Ця величина і відображає помилку реконструкції, тобто різницю між реальними поведінковими тенденціями користувачів та тим, шо модель «вважає нормальним» на підставі вивчених закономірностей.

Враховуючи, що LSTM розглядає послідовність операцій в часі [9], оцінюючи контекст поведінки, приспускається, що послідовність останніх операцій користувача за виділений часовий інтервал визначається так:

$$X = \{x_{t-m+1}, x_{t-m+2}, \dots, x_t\}, \quad (7)$$

де m – довжина часового вікна.

На вхід LSTM-підмережі надходить матриця $X \in \mathbb{R}^{m \times n}$, де кожний рядок відповідає операції, а стовпчики – окремим поведінковим ознакам. Мережа послідовно оновлює прихований стан hy та внутрішню пам'ять ct, зберігаючи інформацію щодо попередніх дій:

$$h_t, c_t = \text{LSTM}(x_t, h_{t-1}, c_{t-1}). \quad (8)$$

Прихований стан ht накопичує контекстну інформацію попередніх операцій, формуючи представлення індивідуального принципу роботи користувача. Враховуючи цей стан, модель обчислює вірогідність того, що поточна операція є аномальною:

$$P(a_t|h_{t-1}) = \sigma(W_h h_{t-1} + b) \quad (9)$$

де $P(a_t|h_{t-1}) \in [0,1]$ відповідає оцінці вірогідності аномалії.

Відтак, гібридний показник R_t формулюється:

$$R_t = \alpha L_{\text{rec}} + (1-\alpha)\big(1 - P(a_t|h_{t-1})\big) \quad (10)$$

Параметр α відіграє роль вагового коефіцієнта балансу між двома складовими функції ризику. Цей параметр регулює те, який вплив у загальній оцінці Rt матиме поточна помилка реконструкції Автокодувальника порівняно з вірогідною оцінкою LSTM.

Якщо α наближається до 1, тоді підвищується вплив локальних, структурних аномалій в оцінці ризику, тобто система переводить акцент на нетипові запити, навіть якщо вони загалом не протирічать поведінковим тенденціям користувача.

Якщо цей параметр є ближчим до 0, тоді вищий ваговий коефіцієнт отримує часову контекстну інформацію, і модель оцінює ризик залежно від послідовності операцій.

Значення α підбирається емпіричним чином в процесі валідації моделі, наприклад за допомогою мінімізації сукупної помилки хибних тривог та пропущених аномалій.

Для підвищення чутливості системи до випадкових коливань показника ризику та хибних тривог, застосовується динамічний поріг адаптації θt, що автоматично підлаштовується під поточний розподіл значень Rt у часі. Він визначається за допомогою такого формулювання:

$$\theta_t = \mu_t + k\sigma_t, \quad (11)$$

де μ_t є стандартним ковзним середнім значенням ризику Rt за останні N спостережень, σ_t – відповідним стандартним відхиленням, а k – коефіцієнтом чутливості. Такий підхід дозволяє моделі враховувати такі явища, як періоди інтенсивної роботи (масові звіти наприклад). Тільки ті значення R_t, що суттєво перевищують межі нормального діапазону ($> \mu_t + k\sigma_t$), інтерпретуються як дійсно аномальні.

Отже, оцінка поведінкового ризику користувачів БД в реальному часі виконується за рахунок фреймворку на базі проміжного програмного забезпечення (ППК, middleware), ключова функція якого полягає у тому, щоб здійснювати аналіз кожного SQL-запиту на етапі його надходження до СУБД, ще до виконання описаних процедур над даними.

Реалізація обчислення вихідного результату моделі в реальному часі може бути виконана різними способами в залежності від архітектури корпоративної системи та можливостей використовуваної СУБД. Одним з найбільш універсальних підходів є використання gRPC або REST API. У цьому випадку ППК фреймворк встановлюється як окремий мікросервіс, що приймає запити від БД [10]. При надходженні SQL-запиту, ППК вилучає ознаки, формує вектор xt, серіалізує його у формат JSON, приклад чого наведений нижче, та надсилає його на зовнішній сервіс формування висновку. Модель, розміщена на виділеному сервері, виконує обчислення ризику Rt та повертає результат по тому самому каналу.

```
{
 "timestamp": "2025-11-04T14:22:37Z",
 "user": "db_admin02",
 "query": "SELECT * FROM confidential_clients;",
 "risk_score": 0.91,
 "action": "execution_paused"
}
```

Альтернативний варіант реалізується через інтеграцію моделі безпосередньо у середовище СУБД. Якщо в СУБД наявна підтримка плагінів або функцій користувача, тоді існує можливість проведення інтеграції у вигляді вбудованого модуля на Python або Java. PostgreSQL забезпечує можливість реалізації логіки через PL/Python, а MySQL – через UDF (User Defined Functions). У цьому випадку, формування висновку відбувається локально в рамках транзакції, що дозволяє скоротити мережеві затримки. При реалізації у контейнері (Docker або Kubernetes pod), кожний екземпляр моделі може взаємодіяти з окремим пулом з'єднань з БД, що є підґрунтям для масштабування системи по мірі росту навантаження.

Відтак, узагальнена схема архітектури системи представлена на рисунку 1.

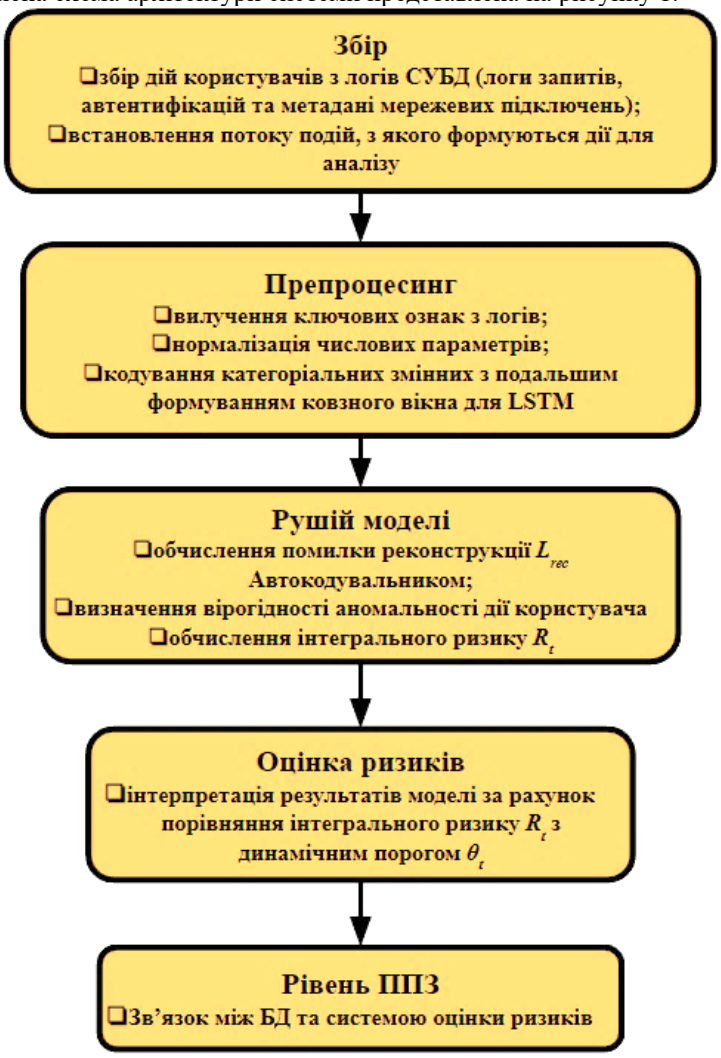

Рис. 1. Архітектура системи оцінювання поведінкових ризиків

Наступний етап полягає у проведенні валідації на даних, максимально наближених до реальних умов корпоративних систем. Для цієї мети була обрана орієнтована на планування ресурсів підприємства (enterprise resource planning system – ERP) інформаційна база SALT (Sales Autocompletion Linked Tables, тобто автодоповнення даних про продажі через зв'язані таблиці), що являє собою набір взаємопов'язаних таблиць, що моделюють типові бізнес-процеси та інформацію щодо підприємства, тобто замовлення, клієнтів, товари, платіжну та логістичну інформацію [11]. Цей набір даних характеризується високим ступенем структурної складності та репрезентативності для корпоративних БД, що робить його придатним середовищем для оцінки ефективності розробленої архітектури.

Були використані комплексні вибірки даних JoinedTables_train.parquet для навчання моделі та JoinedTables_test.parquet для валідації. Ці файли вже об'єднують транзакційні, клієнтські та адресні дані у єдину структуру, зберігаючи часові та семантичні взаємозв'язки, необхідні для послідовного моделювання. Вони були перетворені у структуру з 29 взаємопов'язаними ознаками та часовою міткою активності. Повний навчальний набір містив 1,916,685 записів, а тестовий – 402,855. Кожний екземпляр являє собою абстракцію SQL-операції користувача, включно з характеристиками транзакції SALESDOCUMENTTYPE, TRANSACTIONCURRENCY, PLANT, PRODUCT), контекстною інформацією щодо доступу (SALESORGANIZATION, SHIPTOPARTY, PAYERPARTY), а також часовими ознаками (CREATIONDATE, CREATIONTIME). Частка пропусків не перевищувала 1,2%, що дозволило не застосовувати складні методи реконструкції, тобто достатньою була проста лінійна інтерполяція за часом.

Оскільки поведінкова динаміка користувачів в корпоративних середовищах проявляється на рівні

послідовності операцій, в якості основних форматів вхідних даних застосовувалися часові вікна довжиною у m = 20 запитів, об'єднаних згідно з користувачем та впорядковані за часовими мітками. Для нормалізації числових ознак застосовувався MinMaxScaler, категоріальні атрибути кодувалися методом One-Hot Encoding, в результаті чого розмірність вхідного вектора становила n = 87 ознак. Кінцева матриця для навчання мала форму (95834 × 20 × 87).

Дворівневий симетричний Автокодувальник мав таку структуру: вхідний шар 87 нейронів, прихований шар 65, приховане представлення розмірності 32. У той же час, двошарова LSTM-підмережа мала 62 та 32 нейрони, відповідно. Мережа навчалася на нормальних транзакціях, та застосовувався коефіцієнт α=0.65. Порогове значення обчислювалося кожні 5000 ітерацій.

Навчання моделі відбувалося на GPU (Tesla T4, 16 ГБ), з оптимізатором Adam, причому швидкість навчання становила 1×10^{-3}, розмір міні-пакету – 128, а кількість епох – 50. Майже на 30-тій епосі функція втрат стабілізувалася на рівні MSE = 0.0028, а майже на фінальній – на 0.0023. Середня тривалість однієї епохи становила 72 с, а загальна тривалість – 1 год 2 хв. Вірогідність перенавчання не враховувалася з огляду на співпадіння кривих train/test на графіку реконструкційної помилки, як видно на рисунку 2.

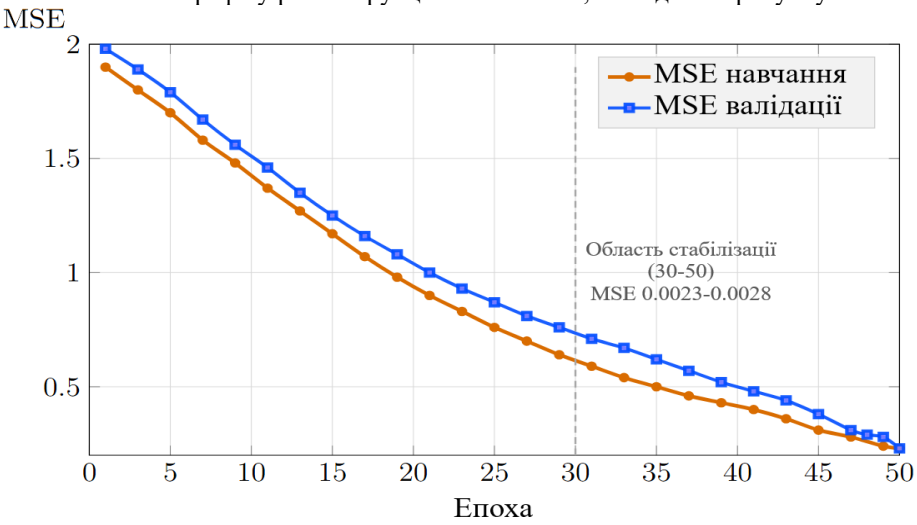

Рис. 2. Криві навчання та валідації моделі

При тестуванні моделі на JoinedTables_test, значення продуктивності становили: Влучність = 0.946, Повнота = 0.931, Оцінка F1 = 0.938, AUC (Area under the curve – ділянка під кривою) = 0.972.

Для інтерпретації результатів було виділено 2 класи аномалій:

– структурні – нетипові комбінації параметрів запиту (UPDATE таблиці clients при нещодавньому SELECT тим самим користувачем);

– поведінкові – зміни динаміки закономірностей активності (підвищення кількості запитів не під час робочого часу, або зміна геолокації).

Модель була здатна розділяти ці категорії: частка правильно виявлених структурних аномалій становила 94.1% та поведінкових – 92.7%.

Середня помилка реконструкції для нормальних запитів Lrecnorm=0.0017, а для аномальних – 0.0079, що означає більш ніж чотирьохкратну розбіжність між кластерами.

Для оцінки переваг моделі була проведена серія експериментів з еквівалентними архітектурами, як показано на таблиці 1.

Таблиця 1

Порівняння з альтернативними моделями

Модель	Влучність	Повнота	F1	MSE	Тривалість навчання
Автокодувальник (без LSTM)	0.873	0.861	0.867	0.0046	36 хв
LSTM (без Автокодувальника)	0.915	0.892	0.903	—	58 хв
Isolation Forest	0.842	0.811	0.826	—	9 хв
One-Class SVM	0.801	0.774	0.787	—	12 хв
Запропонована LSTM-AE	0.946	0.931	0.938	0.0023	62 хв

Відтак, гібридна архітектура характеризується приростом оцінки F1 на 7.1-15.1% відносно класичних алгоритмів та на 6.8% порівняно з ізольованою LSTM. Ізольовані моделі часто ініціювали хибні тривоги в періоди пікового трафіку, тоді як гібридна мережа, завдяки врахуванню контекста та структури, коректно інтерпретувала ці явища я нормальні.

Для перевірки стабільності системи перед шумами у даних проводилися експерименти зі штучно доданим джиттером часових міток (±30 с) та 5%-ою заміною значень категоріальних ознак на випадкові. Навіть за таких умов, F1 оцінка знижувалася не більш ніж до 0.924.

При впровадженні механізму динамічного порогу, частка хибних тривог зменшилася з 6.3% до 3.1%, що є ключовим фактором для промислової експлуатації. Якщо розглядати ефективність з практичної точки зору, модель виявляла такі типи явищ:

– різкі зміни динаміки активності – перехід від пасивного читання (SELECT) до масових операцій DELETE або UPDATE без проміжних транзакцій;

– контекстні невідповідності – операції, нехарактерні для ролі користувача (наприклад, SQL-запити адміністратора з IP-пулу клієнтського відділу);

– повторювані та майже ідентичні операції за короткий проміжок часу.

Графік розподілу помилок реконструкції, продемонстрований на рисунку 3 показує чітке розмежування нормальних та аномальних записів.

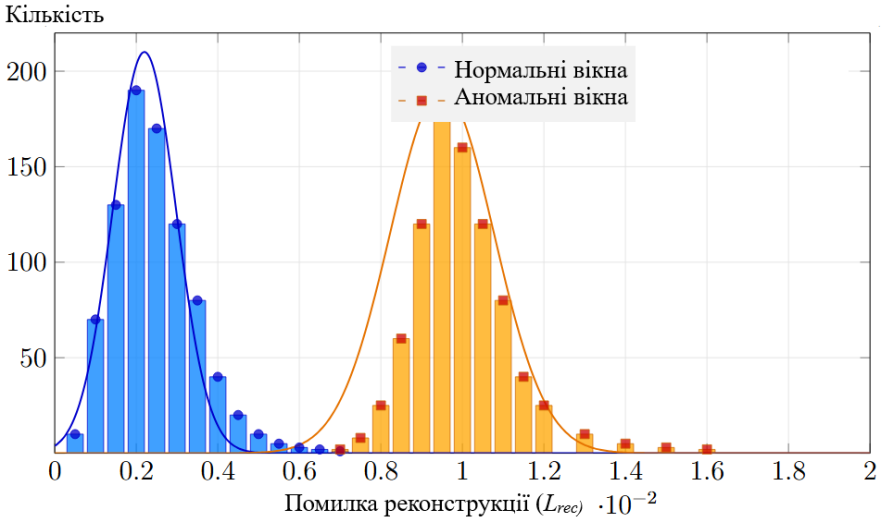

Рис. 3. Розподіл помилки реконструкції для нормальних та аномальних часових вікон (репрезентативна підвибірка даних)

Середня щільність $p(L_{rec})$ нормальних операцій зосереджена в діапазоні 0.001–0.003, у той час, як аномалії формують окремий пік близько 0.008–0.012, що підтверджує можливість порогового розмежування без додаткового навчання класифікатора.

Окрім аналізу розподілу помилки реконструкції, для наочності важливо оцінити її динаміку у часі для послідовностей нормальних та аномальних часових вікон, як показано на рисунку 4. Така візуалізація дозволяє простежити стабільність розмежування між класами на рівні послідовних спостережень.

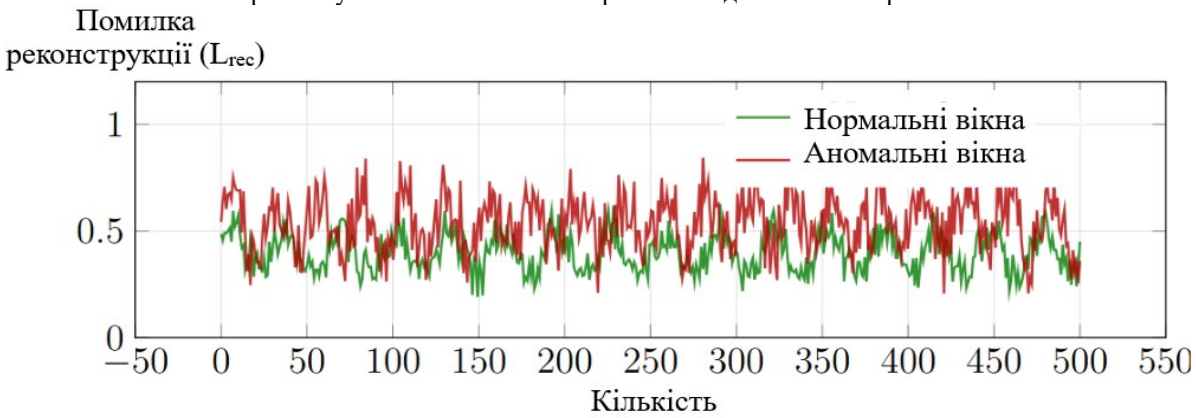

Рис. 4. Часова динаміка помилки реконструкції для нормальних та аномальних часових вікон (репрезентативна підвибірка даних)

Як видно з графіку, криві, що відповідають аномальним часовим вікнам, розташовуються систематично вище, аніж криві нормальних, що підтверджує стабільність виявлення відмінностей моделлю в контексті часової перспективи.

Висновки та перспективи подальших досліджень у даному напрямі

Експериментальні результати підтверджують здатність моделі відрізняти структурні та поведінкові аномалії, включаючи різкі переходи у типах запитів, нехарактерні дії для ролі користувача у корпоративній ієрархії та звернення з нетипових географічних позицій. Підсумовано, що запропонований метод забезпечує підґрунтя для реалізації інтелектуальної системи оцінки поведінкових ризиків корпоративних баз даних у режимі реального часу, здатної інтегруватися у наявні СУБД через інтерфейси проміжного програмного забезпечення без втрат продуктивності.

Перспективи подальших досліджень пов’язані з розширенням моделі за рахунок інтеграції додаткових джерел даних та застосування напівкерованих підходів машинного навчання для зменшення залежності від розмічених вибірок. Перспективним є також дослідження адаптації запропонованого підходу до потокової

обробки даних у режимі реального часу та оцінка його масштабованості в умовах високонавантажених корпоративних СУБД.

Література

1. Li S., Yin Q., Li G., Li Q., Liu Z., Zhu J. Unsupervised Contextual Anomaly Detection for Database Systems. Proceedings of the ACM SIGMOD International Conference on Management of Data. 2022. DOI: 10.1145/3514221.3517887

2. Naserinia V., Beremark M. Anomaly Detection in a SQL Database: A Retrospective Investigation (Master's Thesis, Halmstad University, Master's Programme in Network Forensics). 2022. Retrieved from: https://www.diva-portal.org/smash/get/diva2:1671461/FULLTEXT03

3. Wei Y., Jang-Jaccard J., Xu W., Sabrina F., Camtepe S., Boulic M. LSTM-Autoencoder Based Anomaly Detection for Indoor Air Quality Time Series Data. 2022. arXiv preprint arXiv:2204.06701. Retrieved from: https://arxiv.org/pdf/2204.06701

4. Zhang L., Jia T., Jia M., Li Y., Yang Y., Wu Z. Multivariate Log-based Anomaly Detection for Distributed Database. 2024. arXiv preprint arXiv:2406.07976. Retrieved from: https://arxiv.org/pdf/2406.07976

5. Lachekhab F., Benzaoui M., Tadjer S.A., Bensmaine A., Hamma H. LSTM-Autoencoder Deep Learning Model for Anomaly Detection in Electric Motor. Energies. 2024. Vol. 17. 2340. DOI: 10.3390/en17102340

6. Lee Y., Park C., Kim N., Ahn J., Jeong J. LSTM-Autoencoder Based Anomaly Detection Using Vibration Data of Wind Turbines. Sensors. 2024. Vol. 24. 2833. DOI: 10.3390/s24092833

7. Gong D., Liu L., Le V., Saha B., Mansour M.R., Venkatesh S., van den Hengel A. Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2019. P. 1705–1714. Retrieved from: https://openaccess.thecvf.com/content_ICCV_2019/papers/Gong_Memorizing_Normality_to_Detect_Anomaly_Memory-Augmented_Deep_Autoencoder_for_Unsupervised_ICCV_2019_paper.pdf

8. Barman D., Hasnat A., Nag R. An Introduction to Autoencoders. Computation, CSE-GCETTB. 2021-22. Vol. III. P. 14–23. Retrieved from: https://www.researchgate.net/publication/379542201_AN_INTRODUCTION_TO_AUTOENCODERS2022

9. Okut H. Deep Learning: Long-Short Term Memory. School of Medicine, University of Kansas Medical Center, Wichita, USA. 2021. Retrieved from: https://www.researchgate.net/publication/352383391_Deep_Learning_Long-Short_Term_Memory2021

10. Bolanowski M., Żak K., Paszkiewicz A., Ganzha M., Paprzycki M., Sowiński P., Lacalle Úbeda I., Palau C. Efficiency of REST and gRPC Realizing Communication Tasks in Microservice-Based Ecosystems. 2022. arXiv preprint arXiv:2208.00682. DOI: 10.48550/arXiv.2208.00682

11. Klein T., Biehl C., Costa M., Sres A., Kolk J., Hoffart J. SALT: Sales Autocompletion Linked Business Tables Dataset. 2024. NeurIPS 2024 Third Table Representation Learning Workshop. Retrieved from: https://arxiv.org/pdf/2501.03413v1

References

1. Li, S., Yin, Q., Li, G., Li, Q., Liu, Z., & Zhu, J. (2022). Unsupervised contextual anomaly detection for database systems. In Proceedings of the ACM SIGMOD International Conference on Management of Data. https://doi.org/10.1145/3514221.3517887

2. Naserinia, V., & Beremark, M. (2022). Anomaly detection in a SQL database: A retrospective investigation (Master's thesis, Halmstad University, Master's Programme in Network Forensics). https://www.diva-portal.org/smash/get/diva2:1671461/FULLTEXT03

3. Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., & Boulic, M. (2022). LSTM-Autoencoder based anomaly detection for indoor air quality time series data. arXiv preprint arXiv:2204.06701. https://arxiv.org/pdf/2204.06701

4. Zhang, L., Jia, T., Jia, M., Li, Y., Yang, Y., & Wu, Z. (2024). Multivariate log-based anomaly detection for distributed database. arXiv preprint arXiv:2406.07976. https://arxiv.org/pdf/2406.07976

5. Lachekhab, F., Benzaoui, M., Tadjer, S. A., Bensmaine, A., & Hamma, H. (2024). LSTM-Autoencoder deep learning model for anomaly detection in electric motor. Energies, 17, 2340. https://doi.org/10.3390/en17102340

6. Lee, Y., Park, C., Kim, N., Ahn, J., & Jeong, J. (2024). LSTM-Autoencoder based anomaly detection using vibration data of wind turbines. Sensors, 24, 2833. https://doi.org/10.3390/s24092833

7. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., & van den Hengel, A. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 1705–1714). https://openaccess.thecvf.com/content_ICCV_2019/papers/Gong_Memorizing_Normality_to_Detect_Anomaly_Memory-Augmented_Deep_Autoencoder_for_Unsupervised_ICCV_2019_paper.pdf

8. Barman, D., Hasnat, A., & Nag, R. (2021–22). An introduction to autoencoders. Computation, CSE-GCETTB, III, 14–23. https://www.researchgate.net/publication/379542201_AN_INTRODUCTION_TO_AUTOENCODERS2022

9. Okut, H. (2021). Deep learning: Long-short term memory. School of Medicine, University of Kansas Medical Center. https://www.researchgate.net/publication/352383391_Deep_Learning_Long-Short_Term_Memory2021

10. Bolanowski, M., Żak, K., Paszkiewicz, A., Ganzha, M., Paprzycki, M., Sowiński, P., Lacalle Úbeda, I., & Palau, C. (2022). Efficiency of REST and gRPC realizing communication tasks in microservice-based ecosystems. arXiv preprint arXiv:2208.00682. https://doi.org/10.48550/arXiv.2208.00682

11. Klein, T., Biehl, C., Costa, M., Sres, A., Kolk, J., & Hoffart, J. (2024). SALT: Sales autocompletion linked business tables dataset. In NeurIPS 2024 Third Table Representation Learning Workshop. https://arxiv.org/pdf/2501.03413v1