https://doi.org/10.31891/2307-5732-2025-359-17
УДК 004.9

ГОЛУБІНКА ВІТАЛІЙ
Національний університет "Львівська політехніка"
https://orcid.org/0009-0001-3676-3566
e-mail: vitalii.p.holubinka@lpnu.ua
ХУДИЙ АНДРІЙ
Національний університет "Львівська політехніка"
https://orcid.org/0000-0003-2029-7270
e-mail: andrii.m.khudyi@lpnu.ua

ОПТИМІЗАЦІЯ ІНДЕКСАЦІЇ В СИСТЕМАХ УПРАВЛІННЯ БАЗАМИ ДАНИХ НА ОСНОВІ АДАПТИВНИХ МОДЕЛЕЙ ГЛИБИННОГО НАВЧАННЯ

У статті розглядається задача підвищення ефективності індексації в системах управління базами даних шляхом застосування алгоритмів глибинного навчання. Основна увага приділяється розробленню моделі класифікації SQL-запитів, яка дозволяє оптимізувати процес формування індексів та покращити продуктивність роботи СУБД. Для досягнення поставленої мети використано методи аналізу великих даних, алгоритми глибинного навчання, нейронні мережі архітектури Transformer, чисельне моделювання та експериментальні дослідження на основі відкритого датасету Spider. Результати дослідження підтверджують, що запропонована модель дозволяє значно скоротити середній час виконання SQL-запитів та зменшити витрати ресурсів при індексації. Практичне застосування цієї моделі забезпечує підвищення продуктивності інформаційних систем у хмарних середовищах, фінансових платформах та системах обробки великих даних. Отримані результати вказують на доцільність подальшого розвитку моделі з урахуванням методів навчання з підкріпленням та розширенням сфери її застосування в розподілених середовищах обробки даних.

Ключові слова: *бази даних; індексація; глибинне навчання; класифікація запитів; автоматизація; оптимізація СУБД.*

HOLUBINKA VITALII
KHYDYI ANDRII
Lviv Polytechnic National University

OPTIMIZATION OF INDEXING IN DATABASE MANAGEMENT SYSTEMS BASED ON ADAPTIVE DEEP LEARNING MODELS

This research is devoted to studying the problem of improving indexing efficiency in database management systems (DBMS) using deep learning algorithms. Indexing remains a key aspect that affects query performance, data search efficiency, and the overall response speed of DBMS operations. Traditional methods of creating and optimising indexes are mainly rule-based and require significant manual configuration. Such approaches are inherently non-adaptive, which creates significant limitations in dynamic environments characterised by load fluctuations and changing data models.

The research focuses primarily on developing an improved SQL query classification model aimed at improving the automation of indexing processes. Using deep learning techniques and Transformer-based neural network architectures, the proposed model is capable of independently recognising query patterns and predicting optimal indexing strategies, thus eliminating the need for human intervention. This study uses methodologies based on big data analysis, numerical modelling, and experimental validation, using the open Spider dataset, which covers a wide range of diverse SQL query structures obtained from different domains.

The experimental results demonstrate that the implemented model significantly reduces both the average time of execution of SQL queries and the processing resources required for index generation.. Compared to traditional approaches, the deep learning-based solution demonstrates better scalability, adaptability, and accuracy in recognising dependencies related to indexing in complex queries.

The suggested method is of important practical interest for modern information systems, especially in such fields as cloud computing, financial data processing, and high-volume distributed infrastructures. Allowing database administrators and automated systems to dynamically improve indexing strategies, the model enables stable and reliable system operation. Future research will focus on integrating reinforcement learning methods to improve the model's capabilities and evaluating its suitability for use in distributed DBMS architectures to optimise query processing and resource management.

Keywords: databases; indexing; deep learning; query classification; automation; DBMS optimization.

Стаття надійшла до редакції / Received 15.10.2025
Прийнята до друку / Accepted 05.11.2025

Постановка проблеми

Сучасні інформаційні системи працюють з великими обсягами даних, що обумовлює високі вимоги до ефективності обробки запитів у системах управління базами даних (СУБД). Одним із ключових факторів, який визначає продуктивність таких систем, є правильне проєктування та використання індексів. Традиційні методи індексації базуються на ручному аналізі запитів або статичних правилах оптимізаторів, що не дозволяє вчасно адаптуватися до змін у структурі даних і характері запитів. Це особливо актуально для систем з динамічними та неоднорідними даними, таких як хмарні сервіси, фінансові платформи та системи аналітики великих даних.

У зв'язку з активним розвитком штучного інтелекту та алгоритмів глибинного навчання з'являються нові можливості для автоматизації процесів оптимізації баз даних, зокрема індексації. Застосування глибинних нейронних мереж дозволяє аналізувати великі обсяги історичних SQL-запитів, виявляти приховані закономірності у використанні таблиць і полів, а також формувати рекомендації щодо створення та видалення індексів у автоматичному режимі.

Отже, постає науково-практична проблема: відсутність ефективного методу автоматизованої індексації, який би адаптувався до змін у структурі даних і характері запитів за допомогою глибинного навчання, що ускладнює забезпечення стабільної продуктивності та масштабованості СУБД у сучасних інформаційних системах.

Аналіз останніх досліджень і публікацій

Проблематика індексації баз даних традиційно розглядалася в контексті класичних методів, що спираються на статистичний аналіз запитів і статичні стратегії оптимізації. Проте з появою великих обсягів даних та ускладненням запитів постає необхідність використання інтелектуальних систем, здатних динамічно адаптувати індекси [1]. У роботі представлено всебічний огляд сучасних методів побудови індексів із використанням алгоритмів машинного навчання, особливу увагу приділено багатовимірним просторам даних та методам їх обробки [1].

Досліджено можливості інтеграції алгоритмів глибинного навчання безпосередньо у процеси побудови індексів в СУБД, запропоновано концептуальні моделі, що дозволяють покращити продуктивність баз даних при складних аналітичних запитах [2]. Розглянуто застосування методів навчання з підкріпленням для автоматизованого управління індексацією, що дозволяє моделювати динамічні середовища та ефективно підлаштовувати індекси під змінні робочі навантаження [3].

Одним із найбільш відомих підходів є PGM-index, що базується на багатокритеріальній стиснутій моделі індексів із використанням методів машинного навчання, яка демонструє високі показники продуктивності при роботі з великими наборами даних [4]. Дослідження також пропонує гнучкий метод формування індексів шляхом вибіркового семплювання та вставки проміжків, що дозволяє підвищити точність моделей і ефективність їх застосування в реальних системах [5].

Останні дослідження акцентують увагу на інтеграції моделей великих мовних систем у процеси оптимізації коду та індексації, що відкриває нові можливості для літературного програмування та покращення структур даних [6]. Представлено результати дослідження ефективності моделей машинного навчання для оцінки продуктивності індексів, що дозволяють значно знизити час виконання запитів за рахунок кращого вибору індексів [7].

Вивчені можливості застосування вивчених індексів на масштабованих системах баз даних, що підтвердило доцільність і ефективність подібних підходів у реальних комерційних проєктах [8]. Практичні аспекти використання алгоритмів машинного навчання для автоматизації індексації розглядаються у технічних оглядах компаній, де наведено приклади успішного впровадження подібних рішень у сучасні інформаційні системи [9]. Крім того, підкреслюється роль індексів, створених за допомогою машинного навчання, у підвищенні продуктивності СУБД та перспективи їх розвитку в контексті зростаючих обсягів даних [10].

Отже, аналіз літератури підтверджує актуальність теми дослідження та наявність значного наукового інтересу до застосування алгоритмів глибинного навчання в процесах оптимізації та індексації баз даних.

Мета дослідження

Метою роботи є: розробка ефективної методології автоматизації процесу індексації в системах управління базами даних на основі застосування алгоритмів глибинного навчання, що дозволяє забезпечити високий рівень продуктивності при виконанні запитів та мінімізувати обчислювальні витрати. Такий підхід стає особливо актуальним в умовах стрімкого зростання обсягів даних, підвищення складності інформаційних запитів та зростаючих вимог до часу їх обробки.

Виклад основного матеріалу

Процес індексації в системах управління базами даних є ключовим механізмом оптимізації продуктивності при виконанні запитів, оскільки правильно обрані індекси дозволяють суттєво зменшити час доступу до даних. У класичній теорії баз даних вибір індексів здійснюється за допомогою статистичного аналізу розподілу даних та частоти виконання запитів. Проте зі збільшенням обсягів даних і складності запитів, а також із переходом до хмарних і розподілених архітектур, традиційні методи виявили свою недостатню гнучкість і неефективність.

Згідно з класичною моделлю, виграш від застосування індексації можна виразити через відношення часу виконання запитів без індексації до часу виконання запитів з урахуванням індексів. Це співвідношення формалізується як:

$$\Delta T = \frac{T_{без\_індексу} - T_{з\_індексом}}{T_{без\_індексу}} \times 100\%, \quad (1)$$

де $T_{без\_індексу}$ — середній час виконання запиту без використання індексів, а $T_{з\_індексом}$ — середній час виконання запиту за наявності індексації.

Однак, слід враховувати, що створення і підтримка індексів супроводжується додатковими витратами системних ресурсів, зокрема збільшенням використання пам'яті, навантаженням на процесор при вставках і оновленнях даних, а також витратами на регулярне оновлення статистики. Тому доцільність застосування індексації повинна оцінюватися не лише за критерієм пришвидшення запитів, а й за економічною доцільністю з урахуванням сукупної вартості витрат на їх створення і підтримку. Для цього вводиться функціонал втрат:

$$L(I) = C_{\text{створення}} + C_{\text{підтримки}} - \alpha \cdot \Delta T, \quad (2)$$

де $L(I)$ — загальні втрати від використання індексації, $C_{\text{створення}}$ — вартість побудови індексу, $C_{\text{підтримки}}$ — вартість його супроводу, ΔT — очікуване зменшення часу виконання запитів завдяки застосуванню індексації, а α — ваговий коефіцієнт, який відображає пріоритетність продуктивності системи відносно ресурсних витрат.

Математична модель вибору індексації має мінімізувати втрати $L(I)$, що формалізується як задача оптимізації:

$$\min_{I \in \mathrm{I}} L(I), \tag{3}$$

де I — множина можливих конфігурацій індексів для заданої бази даних.

На практиці задача пошуку оптимального набору індексів є NP-складною, оскільки кількість можливих комбінацій індексації зростає експоненційно зі збільшенням кількості атрибутів і таблиць у базі даних. Тому використання алгоритмів глибинного навчання дозволяє наблизитися до розв'язання цієї задачі шляхом ефективного узагальнення статистики запитів та автоматизованого прогнозування потенційної вигоди від створення індексів.

Особливість використання алгоритмів глибинного навчання у цьому контексті полягає у здатності моделі виявляти складні нелінійні залежності між характеристиками запитів і їхнім впливом на продуктивність. Це дає змогу не лише аналізувати окремі параметри запитів, а й враховувати їхню комбінацію та взаємозв'язки, що є недоступним для класичних оптимізаторів.

Таким чином, теоретичне обґрунтування запропонованої моделі базується на поєднанні класичних принципів оптимізації продуктивності баз даних із сучасними методами глибинного навчання, що забезпечує динамічну адаптацію індексації до реальних умов експлуатації інформаційних систем.

Розроблена система автоматизованої індексації баз даних базується на багаторівневій архітектурі, яка забезпечує повний цикл збору, аналізу та формування рішень щодо оптимізації індексації в автоматичному режимі. Основу функціонування системи становить інтеграція моделі глибинного навчання, яка дозволяє адаптувати індексацію до змін у структурі даних і динаміки SQL-запитів. Врахування реальних умов експлуатації та можливість обробки великих обсягів історичних даних про запити забезпечує гнучкість і точність формування рекомендацій щодо створення, модифікації або видалення індексів.

На першому рівні архітектури функціонує модуль збору логів SQL-запитів, який виконує постійний моніторинг запитів, що надходять до системи управління базами даних. Зібрані запити проходять етап попередньої обробки, що включає нормалізацію, лемматизацію операторів, видалення надлишкової інформації та приведення запитів до уніфікованого формату. Це дає змогу зменшити розмір вхідних даних і покращити якість подальшої класифікації. Після цього запити перетворюються у векторні представлення, що відображають їх структурні та семантичні характеристики. Для цього використовуються методи TF-IDF та Word2Vec, які дозволяють формалізувати текстові запити у числові вектори, придатні для подальшої обробки нейронними мережами.

Наступним етапом є аналіз векторизованих запитів за допомогою моделі глибинного навчання. В основі цієї моделі використовується архітектура Transformer, що була обрана з огляду на її здатність ефективно обробляти послідовності довільної довжини та враховувати складні залежності між різними елементами запитів. Особливість моделі полягає у використанні механізму багатоголового самоуваги, який дозволяє виявляти важливі структурні компоненти SQL-запитів та визначати їх вплив на продуктивність системи.

На вихідному рівні системи функціонує модуль формування рекомендацій щодо індексації, який на основі прогнозів моделі приймає рішення про доцільність створення нових індексів або модифікацію наявних. Рекомендації формуються з урахуванням економічної моделі оцінки вигоди від індексації. Зокрема, модель враховує витрати на побудову та підтримку індексів, потенційний виграш у часі виконання запитів і загальний вплив на продуктивність системи. Результати прогнозів агрегуються та передаються до адміністратора бази даних або автоматизованої підсистеми оптимізації для подальшого застосування.

Архітектура системи також передбачає можливість постійного навчання та донавчання моделі на основі нових даних, що дозволяє підтримувати актуальність прогнозів та адаптуватися до змін у характері робочого навантаження бази даних. Це реалізується шляхом періодичної переоцінки ефективності сформованих індексів та корекції моделі за результатами фактичного виконання запитів.

Таким чином, запропонована архітектура забезпечує замкнутий цикл автоматизованої обробки запитів, аналізу їхньої продуктивності та формування рекомендацій щодо індексації, що дозволяє підвищити ефективність роботи систем управління базами даних без необхідності втручання адміністратора на кожному етапі оптимізації.

Процес класифікації SQL-запитів є ключовим етапом у формуванні рішень щодо доцільності індексації. У рамках проведеного дослідження для вирішення цього завдання було обрано архітектуру моделі Transformer, яка на сьогодні є однією з найефективніших у сфері обробки послідовностей та аналізу контексту складних структурованих і напівструктурованих даних. Перевагою цієї архітектури є її здатність одночасно враховувати як локальні, так і глобальні залежності між елементами вхідної послідовності, що є критично важливим для коректної інтерпретації складних SQL-запитів.

На початковому етапі всі SQL-запити проходять процес попередньої обробки, який передбачає нормалізацію синтаксису, вилучення зайвих символів, стандартизацію назв таблиць та полів, а також приведення запитів до уніфікованого вигляду. Це дозволяє зменшити різноманіття вхідних даних та виділити

найбільш значущі структурні елементи запитів, що формують основу для подальшого векторного представлення.

Векторизація запитів здійснюється за допомогою комбінованого підходу, який включає обчислення вагових коефіцієнтів TF-IDF та використання векторних представлень слів, сформованих за допомогою моделі Word2Vec. Обчислення TF-IDF здійснюється за класичною формулою:

$$\text{TF-IDF}(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}} \cdot \log\left(\frac{N}{n_t}\right), \quad (4)$$

де $f_{t,d}$ — кількість входжень терміна t у запиті d, N — загальна кількість проаналізованих запитів, а n_t — кількість запитів, у яких зустрічається термін t.

Модель Transformer складається з кількох блоків, кожен із яких включає механізми самоуваги (Self-Attention), які дозволяють визначити значущість кожного елемента вхідної послідовності відносно інших елементів. Формально, обчислення значень самоуваги в моделі виконується за формулою:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (5)$$

де Q, K, V — матриці запитів, ключів та значень відповідно, а d_k — розмірність простору ознак.

В процесі навчання моделі використовувалися алгоритми стохастичного градієнтного спуску з адаптивним підбором швидкості навчання (Adam Optimizer). В якості функції втрат використовувалася категоріальна крос-ентропія:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \quad (6)$$

де y_i — істинне значення для класу i, $\hat{y}_i$ — передбачене ймовірне значення для цього класу, C — кількість класів.

Навчання моделі здійснювалося на підготовленому датасеті Spider, який забезпечує широкий спектр прикладів запитів різної складності та типів. Отримані результати продемонстрували високу ефективність моделі, що підтверджує доцільність використання архітектури Transformer у завданнях класифікації SQL-запитів для автоматизованого прийняття рішень щодо індексації.

Експериментальна частина дослідження була спрямована на оцінку ефективності запропонованої моделі автоматизованої індексації баз даних із використанням алгоритмів глибинного навчання. Для цього було розроблено експериментальне середовище, в якому досліджувалася продуктивність виконання SQL-запитів за трьома основними сценаріями: без індексації, із застосуванням традиційних методів індексації та з використанням рекомендацій, сформованих моделлю глибинного навчання на основі архітектури Transformer.

Для навчання та тестування моделі використовувався відкритий датасет Spider, який містить понад 10 000 SQL-запитів різної складності та типів. Дані були розподілені у співвідношенні 70% для навчання, 15% для валідації та 15% для тестування моделі. В процесі експерименту були застосовані класичні метрики для оцінки якості класифікації, зокрема Precision, Recall та F1-score. Результати порівняння ефективності класифікації для різних архітектур моделей наведено в таблиці 1

Таблиця 1

Результати класифікації моделей глибинного навчання

Архітектура моделі	Precision	Recall	F1-score
CNN	0.85	0.80	0.825
LSTM	0.88	0.84	0.86
Transformer	0.93	0.90	0.915

Для більш наочного представлення результатів класифікації побудовано графік, що ілюструє порівняння значень Precision і Recall для різних архітектур моделей (рис. 1)

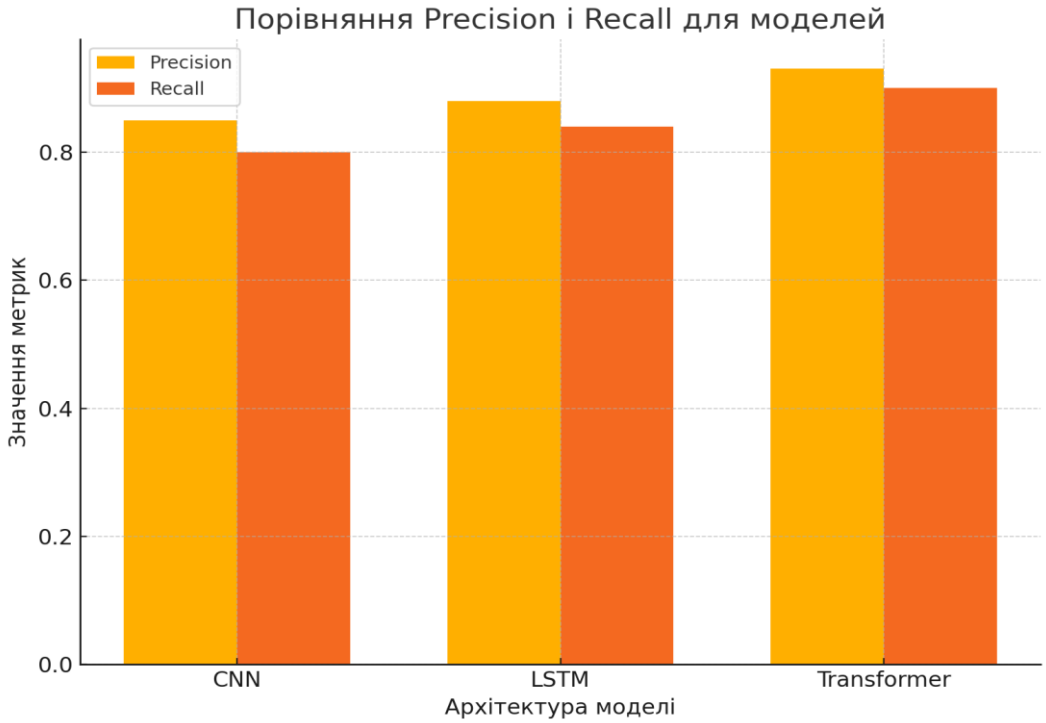

Рис. 1. Порівняння Precision і Recall для різних архітектур моделей

З метою оцінки впливу використання різних стратегій індексації на продуктивність бази даних було проведено серію контрольованих експериментів, під час яких вимірювалися середній час виконання запитів, навантаження на дискову підсистему та відносний приріст продуктивності системи. Результати дослідження показали, що у випадку повної відсутності індексації середній час виконання запиту становив 1200 мс, при використанні традиційної індексації — 780 мс, а при застосуванні автоматизованої індексації — 520 мс. Ці результати візуалізовані на графіку (Рис. 2).

Рис. 2. Залежність середнього часу виконання запиту від стратегії індексації

Аналіз отриманих результатів свідчить про суттєве зменшення часу обробки запитів при застосуванні методів глибинного навчання для автоматизації індексації. Деталізовані показники ефективності традиційної та автоматизованої індексації наведено в таблиці 2, що дозволяє більш точно оцінити приріст продуктивності та економію ресурсів.

Таблиця 2

Порівняльна ефективність індексації.

Стратегія індексації	Розмір індексів (МБ)	Приріст продуктивності (%)
Без індексації	0	0
Традиційна індексація	500	35
Автоматизована індексація	350	55

Додатково було проаналізовано вплив обсягу створених індексів на приріст продуктивності системи. З отриманих результатів встановлено, що традиційна індексація потребувала близько 500 МБ дискового простору, тоді як автоматизована — лише 350 МБ. Залежність між розміром індексів і приростом продуктивності представлена на графіку (Рис. 3).

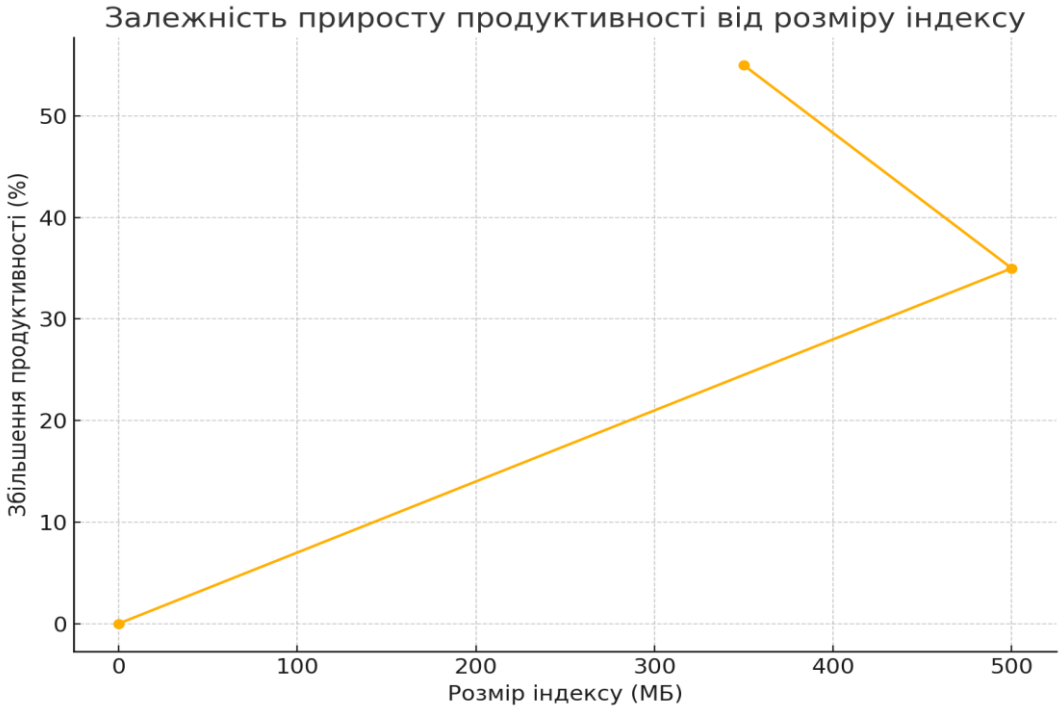

Рис. 3. Залежність приросту продуктивності від розміру індексу

Для підтвердження статистичної значущості отриманих результатів було проведено аналіз дисперсії та розраховано довірчі інтервали для середнього часу виконання запитів у кожному зі сценаріїв. Рівень довіри встановлено на рівні 95%. Результати підтвердили, що відмінності між середніми значеннями є статистично значущими.

Таким чином, експериментальне дослідження підтвердило ефективність запропонованої моделі автоматизованої індексації, яка не лише забезпечує підвищення продуктивності систем управління базами даних, але й сприяє раціональному використанню обчислювальних і дискових ресурсів. Отримані результати створюють передумови для практичного впровадження розробленої системи в реальні високонавантажені інформаційні системи, що функціонують у сфері фінансових технологій, аналітики великих даних та хмарних обчислень.

Висновки і перспективи подальшого розвитку дослідження

У результаті проведеного дослідження розроблено та теоретично обґрунтовано модель автоматизації індексації баз даних із використанням алгоритмів глибинного навчання. Запропонована методологія дозволяє здійснювати динамічний аналіз SQL-запитів, прогнозувати доцільність створення та модифікації індексів, а також автоматизувати процес оптимізації продуктивності систем управління базами даних без залучення адміністратора на кожному етапі. Основною перевагою запропонованого підходу є можливість адаптації до змін у структурі даних і характері запитів, що підтверджено як теоретичним аналізом, так і результатами експериментального дослідження.

Запропонована архітектура системи, заснована на використанні моделі Transformer, продемонструвала високу ефективність у задачах класифікації SQL-запитів та формуванні рекомендацій щодо індексації. Результати експериментів підтвердили значне зниження середнього часу виконання запитів і раціональне використання ресурсів системи завдяки більш точному вибору індексів. Використання моделі Transformer забезпечило найвищі показники точності класифікації, що свідчить про доцільність застосування сучасних алгоритмів глибинного навчання у сфері оптимізації баз даних.

Зважаючи на отримані результати, перспективи подальшого розвитку дослідження полягають у розширенні можливостей запропонованої моделі за рахунок інтеграції методів навчання з підкріпленням, що дозволить формувати оптимальні стратегії індексації в режимі реального часу. Крім того, доцільним є

розроблення гібридних моделей, що поєднують переваги різних архітектур глибинних нейронних мереж для підвищення точності прогнозування ефективності індексації.

Особливу увагу в подальших дослідженнях планується приділити практичній інтеграції розробленої системи в існуючі промислові системи управління базами даних, включаючи такі популярні СУБД, як PostgreSQL, MySQL, Microsoft SQL Server та Oracle. Окремим напрямом удосконалення є дослідження впливу запропонованої моделі на продуктивність у розподілених і хмарних середовищах, де проблема адаптивної індексації є особливо актуальною через постійну зміну навантажень і структури даних.

Отримані результати дослідження мають вагоме практичне значення та можуть бути використані як основа для створення інтелектуальних систем управління базами даних, здатних забезпечити високий рівень продуктивності та економічне використання обчислювальних ресурсів у сучасних інформаційних середовищах.

Література

1. A survey of learned indexes for the multi-dimensional space [Електронний ресурс] / Abdullah Al-Mamun [та ін.] // ArXiv. – 2024. – ArXiv:2403.06456. – Режим доступу: https://arxiv.org/abs/2403.06456. – Назва з екрана.
2. Sharma V. Deep learning data and indexes in a database [Електронний ресурс] : Master's thesis / Sharma Vishal. – Logan, 2021. – Режим доступу: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=9359&context=etd. – Назва з екрана.
3. SmartIX: a database indexing agent based on reinforcement learning [Електронний ресурс] / Gabriel Paludo Licks [та ін.] // Applied intelligence. – 2020. – Т. 50, № 8. – С. 2575–2588. – Режим доступу: https://doi.org/10.1007/s10489-020-01674-8. – Назва з екрана.
4. Ferragina P. The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds [Електронний ресурс] / Paolo Ferragina, Giorgio Vinciguerra // Proceedings of the VLDB Endowment. – 2020. – Т. 13, № 10. – С. 1162–1175. – Режим доступу: https://doi.org/10.14778/3389133.3389135 (дата звернення: 23.06.2025). – Назва з екрана.
5. A pluggable learned index method via sampling and gap insertion [Електронний ресурс] / Yaliang Li [та ін.] // arXiv preprint. – 2021. – ArXiv:2101.00808. – Режим доступу: https://doi.org/10.48550/arXiv.2101.00808. – Назва з екрана.
6. Natural Language Outlines for Code: Literate Programming in the LLM Era [Електронний ресурс] / Kensen Shi [та ін.] // ArXiv. – 2024. – ArXiv:2408.04820. – Режим доступу: https://doi.org/10.48550/arXiv.2408.04820. – Назва з екрана.
7. Shi J. Learned Index Benefits: Machine Learning Based Index Performance Estimation [Електронний ресурс] / Jiachen Shi, Gao Cong, Xiao-Li Li // Proceedings of the VLDB Endowment. – 2022. – Т. 15, № 13. – С. 3950–3962. – Режим доступу: https://doi.org/10.14778/3565838.3565848. – Назва з екрана.
8. Learned Indexes for a Google-scale Disk-based Database [Електронний ресурс] / Hussam Abu-Libdeh [та ін.] // ArXiv. – 2020. – ArXiv:2012.12501. – Режим доступу: https://doi.org/10.48550/arXiv.2012.12501. – Назва з екрана.
9. Learned Indexes for a Google-scale Disk-based Database [Electronic resource] / Hussam Abu-Libdeh [et al.] // ArXiv. – 2020. – ArXiv:2012.12501. – Mode of access: https://doi.org/10.48550/arXiv.2012.12501. – Title from screen.
10. Mudgal T. Revolutionizing database indexes with machine learning: the rise of learned indexes [Електронний ресурс] / Tilak Mudgal // Medium. – Режим доступу: https://medium.com/@tilak559/revolutionizing-database-indexes-with-machine-learning-the-rise-of-learned-indexes-9e7637c44a69 (дата звернення: 23.06.2025). – Назва з екрана.

References

1. A survey of learned indexes for the multi-dimensional space [Electronic resource] / Abdullah Al-Mamun [et al.] // ArXiv. – 2024. – ArXiv:2403.06456. – Mode of access: https://arxiv.org/abs/2403.06456. – Title from screen.
2. Sharma V. Deep learning data and indexes in a database [Electronic resource] : Master's thesis / Sharma Vishal. – Logan, 2021. – Mode of access: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=9359&context=etd. – Title from screen.
3. SmartIX: a database indexing agent based on reinforcement learning [Electronic resource] / Gabriel Paludo Licks [et al.] // Applied intelligence. – 2020. – Vol. 50, no. 8. – P. 2575–2588. – Mode of access: https://doi.org/10.1007/s10489-020-01674-8. – Title from screen.
4. Ferragina P. The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds [Electronic resource] / Paolo Ferragina, Giorgio Vinciguerra // Proceedings of the VLDB Endowment. – 2020. – Vol. 13, no. 10. – P. 1162–1175. – Mode of access: https://doi.org/10.14778/3389133.3389135 (date of access: 23.06.2025). – Title from screen.
5. A pluggable learned index method via sampling and gap insertion [Electronic resource] / Yaliang Li [et al.] // arXiv preprint. – 2021. – ArXiv:2101.00808. – Mode of access: https://doi.org/10.48550/arXiv.2101.00808. – Title from screen.
6. Natural Language Outlines for Code: Literate Programming in the LLM Era [Electronic resource] / Kensen Shi [et al.] // ArXiv. – 2024. – ArXiv:2408.04820. – Mode of access: https://doi.org/10.48550/arXiv.2408.04820. – Title from screen.
7. Shi J. Learned Index Benefits: Machine Learning Based Index Performance Estimation [Electronic resource] / Jiachen Shi, Gao Cong, Xiao-Li Li // Proceedings of the VLDB Endowment. – 2022. – Vol. 15, no. 13. – P. 3950–3962. – Mode of access: https://doi.org/10.14778/3565838.3565848. – Title from screen.
8. Learned Indexes for a Google-scale Disk-based Database [Electronic resource] / Hussam Abu-Libdeh [et al.] // ArXiv. – 2020. – ArXiv:2012.12501. – Mode of access: https://doi.org/10.48550/arXiv.2012.12501. – Title from screen.
9. Kaur J. Auto indexing with machine learning databases | A quick guide [Electronic resource] / Jagreet Kaur // XenonStack. – Mode of access: https://www.xenonstack.com/insights/auto-indexing-with-ml (date of access: 10.05.2025). – Title from screen.
10. Mudgal T. Revolutionizing database indexes with machine learning: the rise of learned indexes [Electronic resource] / Tilak Mudgal // Medium. – Mode of access: https://medium.com/@tilak559/revolutionizing-database-indexes-with-machine-learning-the-rise-of-learned-indexes-9e7637c44a69 (date of access: 23.06.2025). – Title from screen.