КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.32.1104
УДК 004.8:001.891

Лобода Юлія Геннадіївна
кандидат педагогічних наук, доцент, доцент кафедри інформаційних технологій,
Національний університет «Одеська юридична академія», м. Одеса, Україна,
ORCID: 0000-0001-7083-552X
loboda@onua.edu.ua

Трофименко Олена Григорівна
кандидат технічних наук, доцент, доцент кафедри інформаційних технологій,
Національний університет «Одеська юридична академія», м. Одеса, Україна,
ORCID: 0000-0001-7626-0886
trofymenko@onua.edu.ua

Михелєв Ігор Леонідович
кандидат технічних наук, доцент, доцент кафедри інформаційних управляючих систем та технологій,
Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв, Україна,
ORCID: 0000-0001-9579-6547
igor.michelev@nuos.edu.ua

Гайдаєнко Оксана Володимирівна
кандидат технічних наук, доцент, доцент кафедри інформаційних управляючих систем та технологій,Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв, Україна,
ORCID: 0000-0002-6614-5443
oksana.gaidaienko@nuos.edu.ua

Ворона Михайло Владиславович
доктор філософії, викладач кафедри інформаційних управляючих систем та технологій,
Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв, Україна,
ORCID: 0000-0003-4288-0096
mykhailo.vorona@nuos.edu.ua

МУЛЬТИАГЕНТНИЙ ПІДХІД ДО ЗАБЕЗПЕЧЕННЯ СТАБІЛЬНОСТІ КОНВЕЄРІВ ОБРОБКИ ДАНИХ З ВИКОРИСТАННЯМ РОЗПОДІЛЕНИХ СХОВИЩ

Анотація. Стаття присвячена аналізу підходів до забезпечення стабільності конвеєрів обробки даних в аналітичних системах, що є критично важливим аспектом сучасних рішень у сфері Big Data та машинного навчання. Актуальність дослідження зумовлена стрімким зростанням обсягів і різноманітності даних, ускладненням архітектур платформ та підвищенням вимог до надійності й відтворюваності результатів в умовах динамічних навантажень і розподілених обчислень. Показано, що нестабільність конвеєрів проявляється у вигляді каскадних збоїв, деградації продуктивності, порушення узгодженості даних і варіативності результатів, що особливо критично для середовищ Big Data. Проаналізовано обмеження централізованих підходів до оркестрації процесів, які формують єдині точки відмови та знижують адаптивність системи. Обґрунтовано доцільність переходу до мультиагентних механізмів керування, що забезпечують автономність компонентів, локалізацію збоїв і підвищення відмовостійкості. Особливу увагу приділено ролі розподілених сховищ даних, які виконують функції координаційного середовища для взаємодії агентів. Використання документо-орієнтованих, графових, об’єктних та потокових сховищ дозволяє підтримувати узгодженість станів, відтворюваність керуючих дій та асинхронну координацію процесів. Узагальнено підходи до організації аналітичних і прогностичних механізмів підтримки стабільності, включно з моніторингом стану конвеєра, прогнозуванням навантажень і ризиків збоїв, а також адаптацією параметрів обробки на основі накопиченого досвіду. Запропоновано концепцію безперервного циклу оцінювання – прогнозування – адаптації, що забезпечує проактивне керування стабільністю аналітичних

процесів. Практичний приклад у сфері електронної комерції підтвердив ефективність підходу, зокрема скорочення часу відновлення після збоїв, підвищення рівня автоматизації реагування та адаптацію конвеєра до змін навантажень і якості даних. Отримані результати можуть бути використані при проєктуванні та модернізації систем Big Data та машинного навчання, орієнтованих на підвищення стабільності й надійності аналітичних процесів.

Ключові слова: аналітичні системи; конвеєри обробки даних; стабільність; мультиагентні системи; розподілені сховища; Big Data; NoSQL; машинне навчання.

ВСТУП

Постановка проблеми. У сучасних аналітичних системах ключову роль відіграють конвеєри обробки даних (data pipelines), які забезпечують збір, трансформацію та підготовку даних для подальшого аналізу й застосування методів машинного навчання. Зростання обсягів даних, їх різноманітності та швидкості надходження, а також ускладнення архітектур аналітичних платформ зумовлюють підвищені вимоги до стабільності функціонування конвеєрів обробки даних.

Нестабільність конвеєрів обробки даних проявляється у вигляді каскадних збоїв, деградації продуктивності та зниження відтворюваності аналітичних результатів. Некоректні типи даних, проблеми інтеграції та узгодженості інформації формують критичні точки відмови в розподілених аналітичних системах. Крім того, нестабільність обчислювальних процесів у межах аналітичних конвеєрів може призводити до суттєвої варіативності результатів обробки навіть за незначних змін вхідних даних або обчислювального середовища, що ускладнює інтерпретацію та валідацію отриманих результатів.

Актуальність досліджуваної проблеми зумовлена низкою факторів. По-перше, експонентне зростання потоків даних у режимі реального часу від IoT-пристроїв, вебсервісів і мобільних застосунків створює безпрецедентне навантаження на сучасні системи обробки даних. По-друге, сучасні аналітичні системи характеризуються високою архітектурною складністю, що пов’язано з використанням розподілених платформ потокової та пакетної обробки (зокрема Apache Kafka, Apache Spark, Apache Flink), гетерогенних сховищ даних (реляційних і NoSQL-систем, озер даних), а також мікросервісних архітектур. По-третє, критично важливі прикладні системи вимагають мінімальних затримок обробки для підтримки процесів ухвалення рішень у реальному часі.

Наслідки нестабільності конвеєрів обробки даних є суттєвими для функціонування аналітичних систем. Порушення узгодженості даних і каскадні збої призводять до невиконання угод про рівень обслуговування, фінансових втрат унаслідок помилкових аналітичних висновків та зниження ефективності процесів ухвалення рішень. Пошкодження даних у виробничих середовищах створює ризики для бізнес-процесів, а нестабільність аналітичних обчислень обмежує відтворюваність результатів і ускладнює їх подальшу перевірку та використання. Зазначене зумовлює потребу комплексного аналізу механізмів керування конвеєрами обробки даних, здатних забезпечити стабільну роботу аналітичних систем в умовах динамічних змін даних і навантажень на всіх етапах життєвого циклу аналітичного конвеєра: від збору та інтеграції даних до їх аналітичної обробки й використання в системах підтримки ухвалення рішень.

Аналіз останніх досліджень та публікацій. Проблеми стабільності конвеєрів обробки даних є предметом активних досліджень науковців. При цьому основними

джерелами нестабільності виступають порушення якості та узгодженості даних, складність інтеграції гетерогенних джерел, а також динамічні зміни обчислювального середовища. Зокрема, некоректні типи даних і проблеми інтеграції формують найбільш критичні категорії ризиків для стабільності аналітичних конвеєрів [1]. Суттєво, що надійність має забезпечуватися на кожному етапі конвеєра, оскільки проблеми на будь-якому рівні здатні поширюватися по всьому ланцюгу обробки та впливати на кінцеві результати [2]. Порушення узгодженості даних у розподілених середовищах призводить до втрати порядку подій, дублювання результатів і зниження відтворюваності аналітичних обчислень [3], [4]. Окрім інженерних та організаційних чинників, окремою категорією проблем у сучасних аналітичних системах є чисельна нестійкість аналітичних процесів, яка проявляється у варіативності результатів навіть за незначних змін вхідних даних або параметрів обчислювального середовища [5]. Обмеження традиційних підходів до керування конвеєрами значною мірою пов’язані зі статичністю їх структури та централізованими механізмами оркестрації. Орієнтація на наперед задані робочі процеси ускладнює адаптацію до змін навантаження, структури даних або доступності ресурсів [6], [7]. Для підвищення надійності потокових систем досліджено методи динамічної оптимізації контрольних точок (checkpointing), які дозволяють балансувати між відмовостійкістю та продуктивністю [8]. Водночас централізовані оркестратори формують єдину точку відмови та обмежують масштабованість аналітичних платформ, що є критичним для застосувань із високими вимогами до надійності та часу реагування [9], [10]. Наявні підходи до забезпечення стабільності конвеєрів обробки даних умовно поділяють на механізми відмовостійкості, забезпечення узгодженості та контролю якості даних. Дослідження [9], [11] зосереджені на збереженні проміжних станів обробки та можливості відновлення після збоїв, що зменшує втрати даних і часу виконання. Узгодженість у розподілених конвеєрах підтримується через формалізацію подій, використання версій схем і детермінованих операцій агрегації [4], [12]. У статті [13] проаналізовано компроміси між якістю даних і вимогами до їх захисту в конвеєрах на основі оптимізації вибору та композиції сервісів. Такі підходи знижують ризик розбіжностей між компонентами системи, однак потребують чіткої координації між етапами обробки та узгодження правил взаємодії. Контроль якості даних розглядається як важливий фактор стабільності, що охоплює валідацію схем, виявлення аномалій і моніторинг зсувів розподілів [14], [15]. Дослідження [16] пропонує поєднання аналітичних метрик із методами машинного навчання для раннього виявлення деградації якості та підтримки ухвалення рішень, однак такі рішення зазвичай впроваджуються ізольовано та недостатньо узгоджуються з механізмами оркестрації й реконфігурації конвеєра. Еволюція архітектур аналітичних систем від монолітних до розподілених і мікросервісних моделей забезпечила незалежне масштабування компонентів [12], [17], проте водночас ускладнила координацію процесів і підвищила вимоги до керування стабільністю [18]. Використання гетерогенних сховищ даних зумовлене побребою підтримки різних типів даних і патернів доступу [19]. Документо-орієнтовані (зокрема MongoDB), графові (Neo4j) та об’єктні сховища (MinIO) застосовуються для зберігання метаданих, залежностей між задачами та артефактів обробки [20]. Проте такі сховища переважно розглядаються як інфраструктурні компоненти, тоді як їх потенціал як активного координаційного середовища для керування стабільністю розкритий недостатньо. Перспективним напрямом є децентралізовані підходи до керування, зокрема мультиагентні системи, в яких автономні компоненти виконують функції моніторингу, планування та оптимізації. Так, дослідження [21] демонструє

ефективність агентних систем для керування Big Data, де агенти реалізують спеціалізовані ролі моніторингу, планування й оптимізації. Chakraborty описує концепцію самовідновлювальних (self-healing) конвеєрів на основі AI-агентів, здатних автоматично виявляти та усувати проблеми [22]. Для масштабних автономних середовищ розроблено архітектури великомасштабної розподіленої оркестрації, адаптовані до вимог агентних систем [23]. Разом із тим питання інтеграції агентних механізмів з архітектурними рішеннями, а також із гетерогенними розподіленими сховищами даних залишається недостатньо опрацьованим. Застосування прогнозних моделей і методів машинного навчання забезпечує перехід від реактивного до проактивного керування стабільністю конвеєрів обробки даних, зокрема через прогнозування навантаження, виявлення аномалій та оцінювання ризиків збоїв [14], [24-26]. У межах адаптивних підходів реалізовано динамічну зміну параметрів обробки, рівня паралелізму та розподілу ресурсів, у тому числі із використанням інтелектуальних агентів для автономного керування розподіленими середовищами [27-29]. Узагальнення сучасних наукових публікацій свідчить, що проблематика стабільності конвеєрів обробки даних переважно розглядається фрагментарно, з акцентом на окремі технологічні, інфраструктурні або методичні аспекти. Водночас недостатньо систематизованими лишаються питання узгодженого поєднання архітектурних принципів, мультиагентних механізмів керування та ролі розподілених сховищ як координаційного середовища, що зумовлює потребу в інтегрованому підході до забезпечення стабільності аналітичних конвеєрів.

Метою статті є узагальнення та обґрунтування підходів до забезпечення стабільного функціонування конвеєрів обробки даних в аналітичних системах шляхом виокремлення архітектурних патернів, організаційних підходів та інтелектуальних методів керування процесами обробки даних.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

1. Архітектурні аспекти стабільності конвеєрів обробки даних.

Архітектура сучасних аналітичних систем складається з кількох функціональних рівнів, кожен з яких відповідає за окремий аспект обробки даних та вносить специфічний внесок у загальну стабільність системи [17]. У межах даного дослідження було виокремлено чотири взаємопов’язані рівні:

1) рівень даних (Data Layer) забезпечує збір інформації з гетерогенних джерел, її первинну валідацію та зберігання;

2) рівень обробки (Processing Layer) реалізує трансформації, агрегації та аналітичні операції над даними;

3) рівень оркестрації (Orchestration Layer) координує виконання задач, управляє залежностями між ними та контролює потоки даних між компонентами аналітичної системи [11];

4) рівень керування (Control Layer) відповідає за моніторинг стану конвеєра, планування та адаптацію конфігурації системи до змінних умов функціонування.

Точки виникнення нестабільності розподіляються по всіх архітектурних рівнях:

→ на рівні даних критичними є проблеми узгодженості при інтеграції множини джерел, порушення схем та відмови систем зберігання [4];

→ на рівні обробки – каскадні збої при трансформації, чисельна нестійкість процесів та неконтрольоване споживання ресурсів [9];

→ на рівні оркестрації – невідповідність фактичної та очікуваної структури залежностей, перевантаження координуючих компонентів і втрати синхронізації [11];

→ на рівні керування – неповнота інформації про стан системи, запізніле реагування та конфлікти між стратегіями оптимізації [24].

Архітектурні вимоги до забезпечення стабільності охоплюють базові принципи:

→ декомпозиція – розділення функціональності на слабко зв'язані компоненти для локалізації збоїв [18];

→ ізоляція відмов – розмежування ресурсів та обмеження поширення помилок між компонентами конвеєра [13];

→ керованість стану – відстеження змін і можливість відновлення системи до контрольної точки виконання [8];

→ спостережуваність – збір метрик, журналізація подій та реконструкція послідовності операцій для аналізу причин нестабільності [14], [27].

Порівняльний аналіз централізованих та мультиагентних підходів до керування стабільністю конвеєрів обробки даних дозволив виокремити ключові відмінності за основними характеристиками функціонування (табл. 1).

Таблиця 1

Порівняльна характеристика підходів до керування стабільністю конвеєрів обробки даних

Характеристика	Централізовані системи	Мультиагентний підхід
Точка відмови	Єдина (центральний оркестратор)	Розподілена між агентами, відсутність єдиної точки відмови
Адаптивність до змін	Статична конфігурація, потребує перезапуску системи	Динамічна реконфігурація, адаптація без зупинки
Масштабованість	Обмежена пропускною здатністю центрального оркестратора	Горизонтальна, практично необмежена
Складність координації	Низька (централізоване ухвалення рішень)	Висока (потребує механізмів узгодження через сховища)
Час відновлення (MTTR)	10-20 хвилин (залежить від планувальника)	2-5 хвилин (локальне реагування агентів)
Відтворюваність станів	Висока (детерміновані процеси)	Забезпечується через версіонування в розподілених сховищах
Типова доступність системи	95-98%	98-99.5%
Обсяг даних	До 100 GB/день	Понад 1 TB/день
Складність впровадження	Низька	Висока (потребує експертизи у NoSQL та розподілених системах)

Наведені у табл. 1 показники є усередненими значеннями, отриманими шляхом узагальнення даних з публікацій [8], [9], [11], [21-23] та власних експериментальних вимірів у тестовому середовищі. Вони відображають типові сценарії навантаження для виробничих систем. Мультиагентний підхід забезпечує вищу відмовостійкість і масштабованість порівняно з централізованими системами завдяки відсутності єдиної точки відмови та можливості горизонтального масштабування. Водночас такий підхід характеризується підвищеною складністю координації між компонентами, що потребує використання розподілених сховищ як координаційного середовища. Вибір між централізованим і децентралізованим підходом має ґрунтуватися на специфіці прикладного сценарію, обсягах даних, вимогах до доступності та ресурсах інфраструктури.

2. Мультиагентні механізми керування стабільністю аналітичних процесів. Мультиагентний підхід до керування стабільністю аналітичних конвеєрів ґрунтується на делегуванні функцій моніторингу, аналізу та адаптації автономним агентам, які взаємодіють між собою для досягнення спільної мети – безперебійного функціонування системи [22]. На відміну від централізованих підходів, де рішення ухвалює єдиний керуючий компонент, мультиагентна архітектура розподіляє відповідальність між спеціалізованими агентами з обмеженою автономією [23], [29].

У межах такої архітектури доцільно виокремити кілька функціональних ролей:

→ агенти моніторингу здійснюють неперервне спостереження за станом компонентів конвеєра, збір метрик продуктивності та виявлення відхилень від нормального режиму роботи;

→ агенти планування аналізують поточний стан системи, оцінюють потенційні ризики й формують стратегії превентивних дій на основі метаданих конвеєра та інформації про залежності між задачами;

→ агенти реконфігурації реалізують адаптивні зміни у конфігурації конвеєра, виконують відновлення після збоїв та фіксують результати втручань;

→ агенти аналітичної підтримки накопичують знання про історичні інциденти, узагальнюють досвід попередніх втручань і формують рекомендації для підвищення стабільності системи.

Стабільність конвеєра обробки даних досягається не за рахунок централізованого керування, а через спільні інформаційні середовища (рис. 1). Розподілені сховища в даній моделі виконують роль координаційного шару, який забезпечує узгодженість стану, відтворюваність ухвалених керуючих дій та асинхронну взаємодію між агентами.

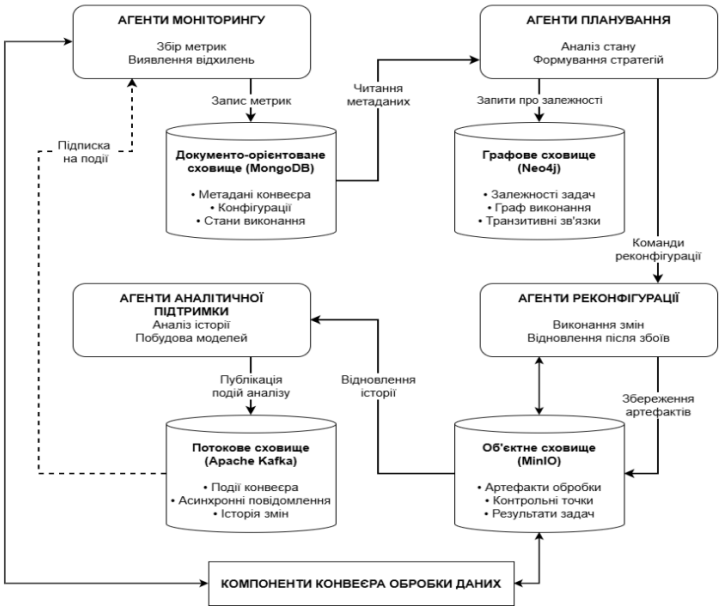

Рис. 1. Механізми мультиагентного керування з використанням розподілених сховищ

Ключова особливість мультиагентної архітектури полягає у використанні розподілених сховищ не лише як засобів зберігання даних, а й як механізмів координації між агентами [19]. Різні типи сховищ обслуговують специфічні інформаційні потреби:

→ документо-орієнтовані бази даних (MongoDB) зберігають метадані конвеєра, конфігурації компонентів та стани виконання задач, що дозволяє агентам планування отримувати актуальну інформацію для ухвалення рішень;

→ графові бази даних (Neo4j) подають залежності між задачами у вигляді графа виконання, що дає змогу ефективно виконувати запити про транзитивні зв'язки та виявляти критичні шляхи;

→ об'єктні сховища (MinIO) забезпечують надійне зберігання артефактів обробки, контрольних точок та результатів задач, необхідних для відновлення стану після збоїв і аналізу історичних даних [30];

→ потокові платформи (Apache Kafka) реєструють події конвеєра, забезпечують асинхронну доставку повідомлень між агентами та зберігають історію змін стану системи [9], [11].

Загальний цикл координації складається з таких етапів: агенти моніторингу фіксують стан системи та передають його у спільні сховища; агенти планування аналізують залежності й формують керуючі дії; агенти реконфігурації виконують зміни та зберігають артефакти результатів; агенти аналітичної підтримки узагальнюють історію виконань і забезпечують інформаційну підтримку наступних рішень. Такий підхід усуває єдину точку відмови та підвищує стійкість системи до локальних збоїв

3. Аналітичні та прогностичні механізми підтримки стабільності. Аналітичні механізми забезпечують кількісну оцінку стану конвеєра обробки даних та виявлення відхилень від очікуваної поведінки. До ключових показників належать метрики продуктивності (час виконання задач, пропускна здатність, затримки між етапами), показники використання ресурсів, а також індикатори якості даних, що фіксують порушення схем, появу аномальних значень і статистичні зсуви [26], [29]. Сукупний аналіз цих показників дозволяє сформувати актуальну картину стану системи та своєчасно виявляти передумови нестабільності (табл. 2).

Таблиця 2

Метрики оцінювання стабільності конвеєрів обробки даних

Метрика	Опис	Цільові значення для систем середнього масштабу	Критичні пороги
MTTR (Mean Time To Recovery)	Середній час відновлення після збою	Централізовані: 10-20 хв; Мультиагентні: 2-5 хв	Понад 20 хвилин
Затримка обробки даних	Час від надходження події до завершення обробки	1-10 секунд	Понад 30 секунд
Точність прогнозування навантаження	Відсоток коректних прогнозів пікових навантажень	70-85%	Нижче 60%
Ступінь автоматизації відновлення	Частка збоїв, усунутих автоматично без втручання операторів	Централізовані: 40-60%; Мультиагентні: 60-80%	Нижче 30%
Доступність системи (Availability)	Відсоток часу безперебійної роботи протягом періоду	Цільове значення: понад 98%	Нижче 95%
Відтворюваність результатів	Ідентичність результатів при повторній обробці тих самих даних	Цільове значення: понад 98%	Нижче 95%
Час затримки ухвалення рішень агентами	Латентність від виявлення проблеми до ініціювання дії	100-500 мілісекунд	Понад 2 секунди
Відсоток каскадних збоїв	Частка інцидентів, що поширилися на суміжні компоненти	Мультиагентні: 5-10%; Централізовані: 20-30%	Понад 30%

Наведені значення отримані шляхом узагальнення даних з публікацій [7], [13], [23], [26], [27], [29] та власних експериментів у тестовому середовищі – стенді інтернет-магазину середнього масштабу (~10,000 замовлень/день). Вони відображають типові сценарії для систем середнього масштабу й використовуються як контрольні порогові показники під час моніторингу.

Прогностичні механізми використовують накопичену історію виконань і результати поточного оцінювання для передбачення майбутніх проблем. Прогнозування навантаження дозволяє завчасно виявляти періоди потенційного перевантаження та ініціювати відповідні дії з управління ресурсами [7]. Оцінка ризиків збоїв базується на аналізі патернів, що передували попереднім інцидентам, і дає змогу формувати ймовірнісні оцінки відмов окремих компонентів [13]. Прогнозування деградації якості спрямоване на виявлення тенденцій погіршення результатів обробки ще до досягнення критичних порогів.

Метрики дозволяють здійснювати як оперативний моніторинг поточного стану конвеєра, так і аналіз ефективності механізмів стабілізації. Інтегральна оцінка стабільності може формуватися на основі зваженої комбінації цих метрик з урахуванням специфіки конкретної аналітичної системи та критичності окремих показників для бізнес-процесів [14], [26], [27].

Механізми адаптації реалізують неперервний цикл оцінювання – прогнозування – адаптації, у межах якого система використовує результати аналізу для підтримання стабільності [7], [26]. Превентивні дії ініціюються на основі прогнозів і спрямовані на запобігання проблемам, тоді як реактивні механізми забезпечують швидке відновлення після фактичних збоїв. Адаптивна оптимізація стосується коригування параметрів конвеєра на основі результатів попередніх втручань, що забезпечує поступове вдосконалення стратегій керування.

На рис. 2 показано узагальнений цикл оцінювання – прогнозування – адаптації у механізмах забезпечення стабільності конвеєра обробки даних.

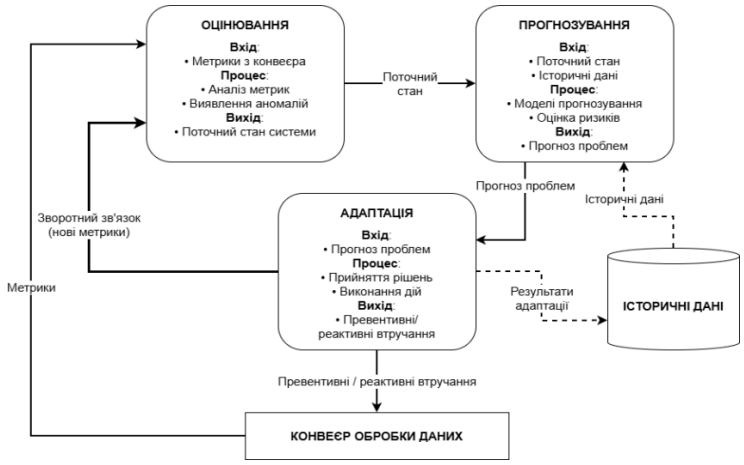

Рис. 2. Цикл оцінювання – прогнозування – адаптації в механізмах забезпечення стабільності

Запропонований цикл оцінювання – прогнозування – адаптації відображає принцип безперервного керування стабільністю аналітичного конвеєра. На відміну від статичних підходів, в яких реакція на збої здійснюється постфактум, дана модель передбачає проактивне використання результатів аналізу та прогнозування для

ініціювання превентивних дій. Замикання циклу за рахунок зворотного зв'язку забезпечує накопичення знань про поведінку системи та поступове вдосконалення стратегій керування, що є ключовим фактором підтримки стабільності в умовах динамічних навантажень і змін якості даних.

4. Приклад застосування: конвеєр аналітики електронної комерції. Яскравим прикладом практичного застосування запропонованого мультиагентного підходу є конвеєр обробки даних системи персоналізованих рекомендацій інтернет-магазину середнього масштабу. Типова архітектура такої системи характеризується обробкою різнорідних потоків даних: подій користувачів (перегляди товарів, додавання до кошика, покупки), оновлень каталогу товарів, інформації про наявність і ціни, а також зовнішніх даних про тренди та сезонні коливання попиту.

Характеристики конвеєра. Система обробляє приблизно 5000 подій на секунду у штатному режимі з піковим навантаженням до 15000 подій/сек під час маркетингових кампаній або розпродажів. Загальний обсяг даних становить близько 500 GB на добу. Цільовий показник доступності системи визначено на рівні 98.5%, а допустима затримка обробки подій для формування рекомендацій у реальному часі не має перевищувати 2-5 секунд.

Архітектурна організація. На рівні даних (Data Layer) використовується Apache Kafka для потокової обробки подій користувачів та MinIO для зберігання артефактів моделей машинного навчання. Рівень обробки (Processing Layer) реалізовано на базі Apache Spark для пакетного переобчислення моделей та Apache Flink для потокової обробки в реальному часі. Рівень оркестрації (Orchestration Layer) координує залежності між задачами через графове сховище Neo4j, яке містить повний граф виконання конвеєра. Рівень керування (Control Layer) реалізований як система мультиагентів, що взаємодіють через MongoDB (метадані та стани виконання) та Apache Kafka (події та повідомлення між агентами).

Типові сценарії нестабільності та механізми реагування. Розглянемо декілька характерних ситуацій порушення стабільності та відповідні механізми їх виявлення й усунення.

Сценарій 1. Раптовий сплеск навантаження. Під час щотижневого розпродажу трафік збільшується у 10 разів протягом перших 15 хвилин після початку акції. Агент моніторингу фіксує зростання затримки (lag) у Kafka-топіках з 100 мілісекунд до 5 секунд. Агент планування отримує сповіщення про відхилення та аналізує поточну конфігурацію Spark-задач через MongoDB, виявляючи критичну задачу генерації рекомендацій у реальному часі. На основі історичних даних про аналогічні інциденти агент планування формує рішення про збільшення рівня паралелізму. Агент реконфігурації виконує масштабування кількості Spark executors з 10 до 30 одиниць та фіксує контрольну точку виконання в MinIO. Протягом 3-4 хвилин затримка знижується до прийнятних 500 мілісекунд. Агент аналітичної підтримки реєструє інцидент у MongoDB разом із параметрами успішного втручання для використання у майбутніх ситуаціях.

Сценарій 2. Зміна схеми даних каталогу товарів. Зовнішня система керування асортиментом оновлює структуру даних про товари, додаючи нові атрибути. Агент моніторингу виявляє зростання кількості помилок валідації на рівні даних та сповіщає агента планування. Агент планування аналізує залежності у Neo4j і визначає, що зміна впливає на задачу конструювання ознак для моделей рекомендацій. Оскільки задача має складні залежності від інших компонентів конвеєра, агент планування формує рішення про тимчасове використання попередньої версії схеми даних із буферної копії

схеми у MongoDB та паралельне оновлення коду трансформацій. Агент реконфігурації розгортає оновлену версію задачі у тестовому середовищі, перевіряє її коректність та поступово переводить обробку на нову версію без зупинки конвеєра. Завдяки використанню версіонування схем даних у MongoDB система підтримує сумісність із обома версіями даних протягом перехідного періоду.

Сценарій 3. Деградація якості моделей рекомендацій. Агент моніторингу відстежує метрики якості рекомендацій (click-through rate, conversion rate) та виявляє їх поступове зниження протягом тижня. Агент аналітичної підтримки аналізує історичні дані у MongoDB та виявляє кореляцію зі зміною розподілу категорій товарів у каталозі (сезонні товари витіснили базовий асортимент). Агент планування ініціює переобчислення моделей на актуальних даних та формує пріоритетну задачу для Spark-кластера. Агент реконфігурації організовує виконання перенавчання моделей у позапікові години роботи системи, зберігає нові версії моделей у MinIO та поступово переводить систему рекомендацій на оновлені моделі. Протягом доби якість рекомендацій відновлюється до цільових показників.

У таблиці 3 наведено результати порівняння ефективності централізованої та мультиагентної архітектури на прикладі конвеєра аналітики електронної комерції.

Таблиця 3

Порівняльні показники ефективності систем

Метрика	До (централізована)	Після (мультиагентна)	Покращення
MTTR	15-20 хв	3-5 хв	зменшення на 70%
Доступність	96.2%	98.7%	+2.5 процентних пунктів
Автоматизація відновлення	45%	70%	+25 п.п. (≈+55%)
Відсоток каскадних збоїв	25%	8%	зменшення на 68%
CPU overhead	н/з	+7%	нова метрика
Латентність координації	н/з	200-400 мс	нова метрика

Показники отримані під час моделювання роботи конвеєра аналітики електронної комерції середнього масштабу, що обробляє 5000-15000 подій/сек із загальним обсягом даних близько 500 GB на добу. Тестове середовище, наближене до виробничого, включало Apache Kafka, Apache Spark, Apache Flink, MongoDB, Neo4j та MinIO, що забезпечувало відтворення типових сценаріїв навантаження. Метрики MTTR, доступності та автоматизації відновлення визначалися на основі журналів симульованих інцидентів, тоді як накладні витрати (CPU overhead) та латентність координації вимірювалися за допомогою системних моніторингових інструментів.

Аналіз показує, що мультиагентна архітектура забезпечує суттєве скорочення часу відновлення (MTTR), зниження частки каскадних збоїв та підвищення рівня автоматизації відновлення. Водночас спостерігаються додаткові накладні витрати та латентність координації, що є типовими для агентних систем і компенсуються загальним зростанням доступності та стійкості конвеєра.

Узагальнення досвіду застосування. Отримані результати підтверджують, що мультиагентна архітектура забезпечує локалізацію збоїв, скорочення часу відновлення (3-5 хв проти 15-20 хв у централізованій системі) та підвищення рівня автоматизації втручань (70% проти 45%). Використання розподілених сховищ (MongoDB, Neo4j, MinIO, Apache Kafka) як координаційного середовища додатково гарантує узгодженість станів між агентами та відтворюваність керуючих дій.

Обмеження підходу. Водночас, реалізація мультиагентної архітектури характеризується підвищеною складністю налаштування та потребує експертизи у

NoSQL-системах. Додаткові накладні витрати від координації між агентами становлять приблизно 5-10% обчислювальних ресурсів порівняно з централізованими рішеннями. Крім того, забезпечення узгодженості між розподіленими сховищами потребує ретельного проєктування механізмів синхронізації та розв'язання конфліктів, що є предметом подальших досліджень.

Отримані результати демонструють, що поєднання архітектурної декомпозиції аналітичних конвеєрів, мультиагентних механізмів керування та використання розподілених сховищ як координаційного середовища створює передумови для підвищення стабільності, адаптивності та відтворюваності аналітичних процесів. Запропонована організація керування дозволяє перейти від реактивного усунення збоїв до проактивного підтримання стабільності конвеєрів обробки даних у розподілених середовищах.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Проаналізовано підходи до забезпечення стабільності конвеєрів обробки даних в аналітичних системах за умов зростання обсягів і різноманітності даних, високої динаміки навантажень та архітектурної складності сучасних платформ Big Data і машинного навчання. Показано, що стабільність аналітичних конвеєрів є інтегральною властивістю всієї системи, яка формується на перетині архітектурних, організаційних та керуючих механізмів.

Виокремлено ключові архітектурні рівні аналітичного конвеєра та систематизовано типові джерела нестабільності на кожному з них. Порівняльний аналіз підтвердив обмеження централізованих підходів до оркестрації аналітичних процесів, зокрема наявність єдиних точок відмови та низьку адаптивність до змін навантажень і структури даних. Обґрунтовано доцільність використання мультиагентних механізмів керування, які забезпечують децентралізацію ухвалення рішень і підвищують відмовостійкість системи.

Особливу увагу приділено ролі розподілених сховищ даних як координаційного середовища взаємодії агентів. Показано, що документо-орієнтовані, графові, об’єктні та потокові сховища можуть виконувати активну функцію підтримки узгодженості станів, відтворюваності керуючих дій та асинхронної координації процесів. Систематизовано набір кількісних метрик оцінювання стабільності, що дозволяє здійснювати як оперативний моніторинг, так і ретроспективний аналіз ефективності механізмів стабілізації. Приклад застосування у задачі аналітики електронної комерції підтвердив ефективність запропонованого підходу, його здатність скорочувати час відновлення після збоїв, підвищувати рівень автоматизації реагування та забезпечувати адаптацію конвеєра до динамічних змін навантажень і якості даних.

Перспективи подальших досліджень можуть бути пов’язані з формалізацією механізмів взаємодії та координації між агентами, розробленням методів розв’язання конфліктів у мультиагентних середовищах і побудовою інтегральних показників стабільності для різних класів аналітичних систем. Окремий інтерес становить експериментальна валідація запропонованих підходів у промислових середовищах. Перспективним напрямом є також дослідження можливостей використання методів машинного навчання та великих мовних моделей для автоматизації ухвалення керуючих рішень у складних розподілених аналітичних конвеєрах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Foidl, H., Golendukhina, V., Ramler, R., & Felderer, M. (2023). Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. *Journal of Systems and Software, 207*, 111855. https://doi.org/10.1016/j.jss.2023.111855
2. Moskovitch, Y., & Jagadish, H. V. (2022). Reliability at multiple stages in a data analysis pipeline. *Communications of the ACM, 65*(7), 118–128. https://doi.org/10.1145/3500923
3. Anderson, W., Bhatnagar, R., Scollick, K., Schito, M., Walls, R., & Podichetty, J. T. (2024). Real-world evidence in the cloud: Tutorial on developing an end-to-end data and analytics pipeline using Amazon Web Services resources. *Clinical and Translational Science, 17*, e70078. https://doi.org/10.1111/cts.70078
4. Terletska, K. (2025). Data consistency in distributed multi-stage event processing pipelines. *The American Journal of Engineering and Technology, 7*(6), 127–134. https://doi.org/10.37547/tajet/volume07issue06-14
5. Kiar, G., Chatelain, Y., de Oliveira Castro, P., Petit, É., Rokem, A., Varoquaux, G., et al. (2021). Numerical uncertainty in analytical pipelines lead to impactful variability in brain networks. *PLOS ONE, 16*(11), e0250755. https://doi.org/10.1371/journal.pone.0250755
6. Залеський, В., Івановський, П., & Федорченко, В. (2024). Сучасні інструменти оркестрації даних для побудови конвеєрів автоматичної обробки даних. *Системи управління, навігації та зв'язку, 2*(76), 95–98. https://doi.org/10.26906/SUNZ.2024.2.095
7. Albers, T., Lazovik, E., & Yousef, M. H. N. (2021). Adaptive on-the-fly changes in distributed processing pipelines. *Frontiers in Big Data, 4*, 666174. https://doi.org/10.3389/fdata.2021.666174
8. Geldenhuys, M., Pfister, B., Scheinert, D., Kao, O., & Thamsen, L. (2022). Khaos: Dynamically optimizing checkpointing for dependable distributed stream processing. In *Proceedings of the 17th Conference on Computer Science and Intelligence Systems (FedCSIS)* (pp. 553–561).
9. Uzoagu, U. U. (2025). Designing resilient, low-latency data pipelines for streaming big data analytics using Apache Kafka and Spark ecosystems. *World Journal of Advanced Research and Reviews, 27*(3), 1856–1873. https://doi.org/10.30574/wjarr.2025.27.3.3369
10. Omolayo, O., Ugboko, R., Oyeyemi, D. O., Oloruntoba, O., & Fakunle, S. O. (2025). Optimizing data pipelines for real-time healthcare analytics in distributed systems: Architectural strategies, performance trade-offs, and emerging paradigms. *International Journal of Scientific and Management Research, 8*(7), 89–99. https://doi.org/10.37502/ijsmr.2025.8708
11. Deepthi, B. G., Rani, K. S., Krishna, P. V., Obaidat, M. S., & Ramalakshmi, K. (2024). An efficient architecture for processing real-time traffic data streams using Apache Flink. *Multimedia Tools and Applications, 83*, 37369–37385. https://doi.org/10.1007/s11042-023-17151-6
12. Singh, N., Singh, D. P., Pant, B., Seth, A., & Elhoseny, M. (2021). BIGMSA – Microservice-based model for big data knowledge discovery: Thinking beyond the monoliths. *Wireless Personal Communications, 116*, 2819–2833. https://doi.org/10.1007/s11277-020-07822-0
13. Polimeno, A., Braghin, C., Anisetti, M., & Ardagna, C. (2025). Maximizing data quality while ensuring data protection in service-based data pipelines. *Journal of Big Data, 12*, 62. https://doi.org/10.1186/s40537-025-01118-5
14. Dadeboe, A., Mansourifard, F., & Sugrim, S. (2025). Uncertainty quantification and data provenance for data pipeline security analysis. In *Proceedings of the 7th Workshop on Design Automation for CPS and IoT* (pp. 1–6). https://doi.org/10.1145/3722573.3727831
15. Anisha, S., & Thiyagarajan, S. (2025). An explainable deep learning based data mining framework for automated data loading optimization and pipeline evaluation using sentiment analysis. *International Journal of Environmental Sciences, 11*(7), 584–602. https://doi.org/10.64252/yx8fzb84
16. Трофименко, О. Г., Лобода, Ю. Г., Гура, В. І., Дика, А. І., & Стрілець, М. І. (2024). Інструменти штучного інтелекту для системного аналізу. *Вісник Херсонського національного технічного університету, 4*, 349–357. https://doi.org/10.35546/kntu2078-4481.2024.4.46
17. Janev, V. (2020). Ecosystem of big data. In *Knowledge graphs and big data processing* (Lecture Notes in Computer Science, Vol. 12072). Springer. https://doi.org/10.1007/978-3-030-53199-7_1
18. Demchenko, Y., de Laat, C., & Membrey, P. (2014). Defining architecture components of the big data ecosystem. In *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS)* (pp. 104–112). https://doi.org/10.1109/CTS.2014.6867550
19. Khan, W., Kumar, T., Zhang, C., Raj, K., & Roy, A. (2023). SQL and NoSQL database software architecture performance analysis and assessments: A systematic literature review. *Big Data and Cognitive Computing, 7*(2), 97. https://doi.org/10.3390/bdcc7020097

20. Лобода, Ю., & Кричун, О. (2025). Інтелектуальні агенти як засіб адаптивного керування конвеєрами даних у середовищах Big Data. In *Інформаційні технології: моделі, алгоритми, системи (ITMAS-2025): VI Міжнародна науково-практична інтернет-конференція* (pp. 456–458).
21. Simamora, R., Ulwi, K., & M. A. H. (2024). Implementation of agent systems in big data management: Integrating artificial intelligence for data mining optimization. *Journal of Computer Science Advancements, 2*(1), 33–47. https://doi.org/10.70177/jsca.v2i1.1210
22. Chakraborty, S. (2025). Beyond ETL: How AI agents are building self-healing data pipelines. *Journal of Computer Science and Technology Studies, 7*(3), 741–756. https://doi.org/10.32996/jcsts.2025.7.3.81
23. Kodi, D. (2024). Designing real-time data pipelines for predictive analytics in large-scale systems. *Transactions on Sustainable Computing Systems, 2*(4), 178–188. https://doi.org/10.69888/ftscs.2024.000294
24. Alva, L. (2025). Generative AI for self-optimizing and autonomous data pipelines. *World Journal of Advanced Research and Reviews, 26*(2), 1071–1079. https://doi.org/10.30574/wjarr.2025.26.2.1667
25. Trirat, P., Jeong, W., & Hwang, S. J. (2024). AutoML-Agent: A multi-agent LLM framework for full-pipeline AutoML. *arXiv*. https://doi.org/10.48550/arXiv.2410.02958
26. Gandhi, H., & Solanki, S. (2025). Advanced CI/CD pipelines for testing big data job orchestrators. *Journal of Quantum Science and Technology, 2*(1), 131–149. https://doi.org/10.63345/jqst.v2i1.155
27. Sinha, R. K. (2025). Architecting resilient data pipelines: A framework for enterprise analytics in cloud environments. *World Journal of Advanced Engineering Technology and Sciences, 15*(3), 1099–1105. https://doi.org/10.30574/wjaets.2025.15.3.0942
28. Li, Y., Wu, B., Huang, Y., Luan, S., & Zhang, L. (2024). Developing trustworthy artificial intelligence: Insights from research on interpersonal, human-automation, and human-AI trust. *Frontiers in Psychology, 15*. https://doi.org/10.3389/fpsyg.2024.1382693
29. Zhang, L., Zhai, Y., Tong, J., Huang, X., Duan, C., & Li, Y. (2025). AgentFM: Role-aware failure management for distributed databases with LLM-driven multi-agents. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25)* (pp. 525–529). https://doi.org/10.1145/3696630.3728492
30. Malhotra, S., Yashu, F., Saqib, M., Mehta, D., Jangid, J., & Dixit, S. (2025). Evaluating fault tolerance and scalability in distributed file systems: A case study of GFS, HDFS, and MinIO. *arXiv*. https://arxiv.org/pdf/2502.01981

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023

Yuliia Loboda
PhD, Associate Professor, Associate Professor at the Department of Information Technologies of the National University "Odessa Law Academy", Odesa, Ukraine,
ORCID: 0000-0001-7083-552X
loboda@onua.edu.ua

Olena Trofymenko
PhD, Associate Professor, Associate Professor at the Department of Information Technologies of the National University "Odessa Law Academy", Odesa, Ukraine,
ORCID: 0000-0001-7626-0886
trofymenko@onua.edu.ua

Ihor Mykheliev
Candidate of Sciences in Technology, Associate Professor at the Department of Information Control Systems and Technology of the Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine,
ORCID: 0000-0001-9579-6547
igor.michelev@nuos.edu.ua

Oksana Haidaienko
Candidate of Sciences in Technology, Associate Professor at the Department of Information Control Systems and Technology of the Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine,
ORCID: 0000-0002-6614-5443
oksana.gaidaienko@nuos.edu.ua

Mykhaylo Vorona
PhD, lecturer at the Department of Information Control Systems and Technology of the Admiral Makarov National University of Shipbuilding, Mykolaiv, Ukraine,
ORCID: 0000-0003-4288-0096
mykhailo.vorona@nuos.edu.ua

MULTI-AGENT APPROACH TO DATA PIPELINE STABILITY USING DISTRIBUTED STORAGE COORDINATION

Abstract. The article analyzes approaches to ensuring the stability of data processing pipelines in analytical systems, which is a critical aspect of modern Big Data and machine learning solutions. The relevance of the study is driven by the rapid growth of data volumes and heterogeneity, increasing architectural complexity of analytical platforms, and rising requirements for reliability and reproducibility under dynamic workloads and distributed computing. Instability in pipelines manifests through cascading failures, performance degradation, data inconsistency, and variability of results, which is particularly critical in Big Data environments. The limitations of centralized orchestration approaches are examined, highlighting their reduced adaptability and single points of failure. The paper substantiates the transition to decentralized multi-agent control mechanisms that provide component autonomy, fault localization, and improved resilience. Special attention is given to the role of distributed storage systems, which serve not only as repositories but also as coordination environments for agent interaction. The use of document-oriented, graph, object, and streaming storage systems enables consistent state management, reproducibility of control actions, and asynchronous coordination of processing workflows. The study summarizes approaches to organizing analytical and predictive mechanisms for stability support, including pipeline monitoring, workload and failure risk forecasting, and adaptive adjustment of processing parameters based on accumulated experience. A concept of a continuous evaluation–forecasting–adaptation cycle is proposed, enabling proactive stability management. A case study in e-commerce confirms the effectiveness of the approach, demonstrating reduced recovery time, increased automation of interventions, and improved adaptability of pipelines to workload and data quality changes. The results can be applied in the design and modernization of Big Data and machine learning systems aimed at enhancing stability, reliability, and reproducibility of analytical processes.

Keywords: analytical systems; data processing pipelines; stability; multi-agent systems; distributed storage; Big Data; NoSQL; machine learning.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Foidl, H., Golendukhina, V., Ramler, R., & Felderer, M. (2023). Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. *Journal of Systems and Software, 207*, Article 111855. https://doi.org/10.1016/j.jss.2023.111855
2. Moskovitch, Y., & Jagadish, H. V. (2022). Reliability at multiple stages in a data analysis pipeline. *Communications of the ACM, 65*(7), 118–128. https://doi.org/10.1145/3500923
3. Anderson, W., Bhatnagar, R., Scollick, K., Schito, M., Walls, R., & Podichetty, J. T. (2024). Real-world evidence in the cloud: Tutorial on developing an end-to-end data and analytics pipeline using Amazon Web Services resources. *Clinical and Translational Science, 17*, e70078. https://doi.org/10.1111/cts.70078
4. Terletska, K. (2025). Data consistency in distributed multi-stage event processing pipelines. *The American Journal of Engineering and Technology, 7*(6), 127–134. https://doi.org/10.37547/tajet/volume07issue06-14
5. Kiar, G., Chatelain, Y., de Oliveira Castro, P., Petit, É., Rokem, A., Varoquaux, G., et al. (2021). Numerical uncertainty in analytical pipelines lead to impactful variability in brain networks. *PLOS ONE, 16*(11), Article e0250755. https://doi.org/10.1371/journal.pone.0250755
6. Zaleskyi, V., Ivanovskyi, P., & Fedorchenko, V. (2024). Modern data orchestration tools for building big data processing pipelines. *Control, Navigation and Communication Systems, 2*(76), 95–98. https://doi.org/10.26906/SUNZ.2024.2.095
7. Albers, T., Lazovik, E., & Yousef, M. H. N. (2021). Adaptive on-the-fly changes in distributed processing pipelines. *Frontiers in Big Data, 4*, Article 666174. https://doi.org/10.3389/fdata.2021.666174
8. Geldenhuys, M., Pfister, B., Scheinert, D., Kao, O., & Thamsen, L. (2022). Khaos: Dynamically optimizing checkpointing for dependable distributed stream processing. In *Proceedings of the 17th Conference on Computer Science and Intelligence Systems (FedCSIS)* (pp. 553–561). IEEE.
9. Uzoagu, U. U. (2025). Designing resilient, low-latency data pipelines for streaming big data analytics using Apache Kafka and Spark ecosystems. *World Journal of Advanced Research and Reviews, 27*(3), 1856–1873. https://doi.org/10.30574/wjarr.2025.27.3.3369
10. Omolayo, O., Ugboko, R., Oyeyemi, D. O., Oloruntoba, O., & Fakunle, S. O. (2025). Optimizing data pipelines for real-time healthcare analytics in distributed systems: Architectural strategies, performance trade-offs, and emerging paradigms. *International Journal of Scientific and Management Research, 8*(7), 89–99. https://doi.org/10.37502/ijsmr.2025.8708
11. Deepthi, B. G., Rani, K. S., Krishna, P. V., Obaidat, M. S., & Ramalakshmi, K. (2024). An efficient architecture for processing real-time traffic data streams using Apache Flink. *Multimedia Tools and Applications, 83*, 37369–37385. https://doi.org/10.1007/s11042-023-17151-6
12. Singh, N., Singh, D. P., Pant, B., Seth, A., & Elhoseny, M. (2021). BIGMSA: Microservice-based model for big data knowledge discovery: Thinking beyond the monoliths. *Wireless Personal Communications, 116*, 2819–2833. https://doi.org/10.1007/s11277-020-07822-0
13. Polimeno, A., Braghin, C., Anisetti, M., & Ardagna, C. (2025). Maximizing data quality while ensuring data protection in service-based data pipelines. *Journal of Big Data, 12*, Article 62. https://doi.org/10.1186/s40537-025-01118-5
14. Dadeboe, A., Mansourifard, F., & Sugrim, S. (2025). Uncertainty quantification and data provenance for data pipeline security analysis. In *Proceedings of the 7th Workshop on Design Automation for CPS and IoT* (pp. 1–6). ACM. https://doi.org/10.1145/3722573.3727831
15. Anisha, S., & Thiyagarajan, S. (2025). An explainable deep learning-based data mining framework for automated data loading optimization and pipeline evaluation using sentiment analysis. *International Journal of Environmental Sciences, 11*(7), 584–602. https://doi.org/10.64252/yx8fzb84
16. Trofymenko, O. G., Loboda, Y. G., Hura, V. I., Dyka, A. I., & Strilets, M. I. (2024). Artificial intelligence tools for system analysis. *Visnyk Khersonskoho Natsionalnoho Tekhnichnoho Universytetu, 4*(91), 349–357. https://doi.org/10.35546/kntu2078-4481.2024.4.46
17. Janev, V. (2020). Ecosystem of big data. In V. Janev, D. Graux, H. Jabeen, & E. Sallinger (Eds.), *Knowledge graphs and big data processing* (Lecture Notes in Computer Science, Vol. 12072, pp. 1–23). Springer. https://doi.org/10.1007/978-3-030-53199-7_1
18. Demchenko, Y., de Laat, C., & Membrey, P. (2014). Defining architecture components of the big data ecosystem. In *2014 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 104–112). IEEE. https://doi.org/10.1109/CTS.2014.6867550

19. Khan, W., Kumar, T., Zhang, C., Raj, K., & Roy, A. (2023). SQL and NoSQL database software architecture performance analysis and assessments: A systematic literature review. *Big Data and Cognitive Computing*, *7*(2), Article 97. https://doi.org/10.3390/bdcc7020097
20. Loboda, Y., & Krychun, O. (2025). Intelligent agents as a means of adaptive control of data pipelines in big data environments. In *Information Technologies: Models, Algorithms, Systems (ITMAS-2025): VI International Scientific and Practical Internet Conference* (pp. 456–458).
21. M., A. H., Simamora, R., & Ulwi, K. (2024). Implementation of agent systems in big data management: Integrating artificial intelligence for data mining optimization. *Journal of Computer Science Advancements*, *2*(1), 33–47. https://doi.org/10.70177/jsca.v2i1.1210
22. Chakraborty, S. (2025). Beyond ETL: How AI agents are building self-healing data pipelines. *Journal of Computer Science and Technology Studies*, *7*(3), 741–756. https://doi.org/10.32996/jcsts.2025.7.3.81
23. Kodi, D. (2024). Designing real-time data pipelines for predictive analytics in large-scale systems. *Transactions on Sustainable Computing Systems*, *2*(4), 178–188. https://doi.org/10.69888/ftscs.2024.000294
24. Alva, L. (2025). Generative AI for self-optimizing and autonomous data pipelines. *World Journal of Advanced Research and Reviews*, *26*(2), 1071–1079. https://doi.org/10.30574/wjarr.2025.26.2.1667
25. Trirat, P., Jeong, W., & Hwang, S. J. (2024). AutoML-Agent: A multi-agent LLM framework for full-pipeline AutoML. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2410.02958
26. Gandhi, H., & Solanki, S. (2025). Advanced CI/CD pipelines for testing big data job orchestrators. *Journal of Quantum Science and Technology*, *2*(1), 131–149. https://doi.org/10.63345/jqst.v2i1.155
27. Sinha, R. K. (2025). Architecting resilient data pipelines: A framework for enterprise analytics in cloud environments. *World Journal of Advanced Engineering Technology and Sciences*, *15*(3), 1099–1105. https://doi.org/10.30574/wjaets.2025.15.3.0942
28. Li, Y., Wu, B., Huang, Y., & Luan, S. (2024). Developing trustworthy artificial intelligence: Insights from research on interpersonal, human-automation, and human-AI trust. *Frontiers in Psychology*, *15*. https://doi.org/10.3389/fpsyg.2024.1382693
29. Zhang, L., Zhai, Y., Tong, J., Huang, X., Duan, C., & Li, Y. (2025). AgentFM: Role-aware failure management for distributed databases with LLM-driven multi-agents. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25)* (pp. 525–529). ACM. https://doi.org/10.1145/3696630.3728492
30. Malhotra, S., Yashu, F., Saqib, M., Mehta, D., Jangid, J., & Dixit, S. (2025). Evaluating fault tolerance and scalability in distributed file systems: A case study of GFS, HDFS, and MinIO. *arXiv Preprint*. https://arxiv.org/pdf/2502.01981

Отримано редакцією журналу / Received: 14.01.26
Прорецензовано / Revised: 30.01.26
Схвалено до друку / Accepted: 26.03.26