

АНАЛІЗ ПОТОКОВИХ ДАНИХ У РЕАЛЬНОМУ ЧАСІ (STREAMING DATA ANALYSIS)

Вінницький національний технічний університет

Анотація

У тезах розглянуто актуальні підходи до аналізу поточкових даних у реальному часі, що набувають значного поширення у зв'язку з розвитком Інтернету речей, фінансових технологій, телекомунікацій та систем моніторингу. Проаналізовано ключові інструменти для обробки поточкових даних (Apache Kafka, Apache Flink, Spark Streaming), а також методи машинного навчання, орієнтовані на роботу з безперервними потоками інформації. Особливу увагу приділено викликам масштабованості, забезпеченню низької затримки та гарантуванню цілісності даних. Робота демонструє перспективи впровадження поточкового аналізу у сфері кібербезпеки, прогнозу аналітики та інтелектуальних міських систем.

Ключові слова: поточкові дані, реальний час, Big Data, машинне навчання, Apache Kafka, Spark Streaming, IoT.

Abstracts

The abstract discusses current approaches to real-time stream data analysis, which are becoming increasingly widespread due to the development of the Internet of Things, financial technologies, telecommunications, and monitoring systems. Key tools for processing streaming data (Apache Kafka, Apache Flink, Spark Streaming) are analyzed, as well as machine learning methods focused on working with continuous streams of information. Particular attention is paid to scalability challenges, ensuring low latency, and guaranteeing data integrity. The work demonstrates the prospects for implementing streaming analysis in the fields of cybersecurity, predictive analytics, and smart city systems.

Keywords: streaming data, real time, Big Data, machine learning, Apache Kafka, Spark Streaming, IoT.

Вступ

Сучасне суспільство генерує колосальні обсяги даних, значна частина яких надходить у вигляді безперервних потоків. Це можуть бути дані з датчиків Інтернету речей (IoT), фінансові транзакції, записи з логів серверів, потоки кліків у вебдодатках, телеметрія у транспортних системах чи повідомлення в соціальних мережах. Традиційні методи зберігання та пакетної обробки даних (batch processing) виявляються недостатніми у випадках, коли необхідно реагувати миттєво – наприклад, виявляти шахрайські операції у фінансових транзакціях, вчасно виявляти відмови обладнання чи забезпечувати кібербезпеку.

Тому виникає потреба в streaming data analysis – технологіях, які дозволяють обробляти, аналізувати та приймати рішення на основі даних у момент їхнього надходження. Сучасні платформи, такі як Apache Kafka, Apache Flink, Spark Streaming, пропонують гнучкі інструменти для масштабованої та стійкої обробки потоків [1].

Додатковий виклик полягає у застосуванні машинного навчання у реальному часі: класичні моделі не завжди підходять, оскільки дані постійно змінюються, і система має вміти навчатися динамічно (online learning).

Завдяки своїй універсальності поточковий аналіз застосовується у багатьох сферах: фінансовий сектор, кібербезпека, інтелектуальні міста (Smart City), охорона здоров'я, маркетинг і соціальні мережі.

Таким чином, аналіз поточкових даних стає ключовою складовою сучасної Big Data аналітики, відкриваючи нові можливості для наукових досліджень та прикладних рішень.

Результати дослідження

У світі, де кількість джерел даних стрімко зростає, дедалі більше завдань вимагає не відкладеної пакетної обробки, а миттєвої реакції на події, що відбуваються «тут і зараз». Поточковий аналіз даних – це підхід, у якому інформація обробляється відразу при надходженні, а рішення (виявлення аномалії, тригер алерта, оновлення прогнозу) приймаються у реальному часі або з мінімальною затримкою. Для України, яка сьогодні інтенсивно розвиває цифрові сервіси, захищає критичну інфраструктуру та

масштабує бізнес-процеси в умовах підвищених кіберризиків, ці технології набувають особливої значущості: у банківській сфері, в роботі державних систем моніторингу, у Smart City ініціативах та агротех-проектах, де швидка реакція може зберегти ресурси або життя людей. Базові принципи потокового аналізу та суміжні архітектурні патерни нині широко застосовують у промисловості та наукових дослідженнях, а перелік інструментів охоплює як open-source платформи, так і керовані хмарні сервіси [1].

Потокова обробка оперує поняттями вікон (windows) та часу, які визначають, як агрегуються події. Існує кілька концептуально різних підходів: ковзні (sliding) вікна дозволяють створювати перекриті часові інтервали, завдяки чому система може «гладко» реагувати на зміни й виявляти тренди, що формуються поступово; розмір вікна й крок слайду прямо впливають на баланс між чутливістю до миттєвих флуктуацій і стабільністю оцінок. Micro-batch підхід, який зробив популярним Spark Streaming, акумулює події у невеликі пакети (мікробатчі) і обробляє їх серіями – це спрощує реалізацію паралелізму й забезпечує високу пропускну здатність, але вводить обмеження на найменшу досяжну затримку [2]. Навпаки, системи «строкового» (true streaming) виконання, як правило представлені Flink або Beam/Google Dataflow, орієнтуються на обробку кожної події або дуже малих груп подій із мінімальними паузами. Ключове поняття при цьому – обробка за часом події (event-time): вона віддає перевагу часовим міткам генерації події, а не часу її надходження до системи. Це критично важливо у сценаріях, де дані приходять із затримками або в неправильному порядку – наприклад, у мережах датчиків віддалених агрополітів чи у телекомунікаційних логах. Для коректної роботи event-time підходів використовують механізм watermark (водяний знак), який дає змогу визначати «точку» в часі, після якої запізнілі події вважаються занадто пізніми або обробляються окремо. Вибір між sliding windows, micro-batch і event-time обробкою часто диктується природою даних і вимогами до затримки: якщо потрібна максимальна реактивність і точність при наявності запізнілих подій – вибір схиляється до event-time і систем із низькою латентністю; якщо пріоритет – масштабування й простота розгортання – micro-batch може виявитися прийнятним компромісом [3].

У технічному стеку потокових рішень часто можна побачити кілька класичних компонентів: протокол/шина для передачі подій, система обробки потоків і інструменти для збереження стану і його відновлення. Apache Kafka виступає не просто як «черга повідомлень», а скоріше як розподілений журнал подій (commit log) – централізована шина, яка дозволяє надійно інгестувати, буферизувати та реплікувати великі потоки подій із високою пропускну здатністю. Kafka має екосистему, що включає Kafka Streams для вбудованої обробки стрімів у Java/Scala та ksqlDB для SQL-подібної обробки потоків [4]. Apache Spark спочатку ввів концепцію micro-batch через Spark Streaming, а згодом еволюціонував у Structured Streaming, який надає більш уніфіковану модель обробки даних і інтеграцію з MLlib для побудови аналітичних конвеєрів. Apache Flink позиціонується як фреймворк «справжнього» стрімінгу з сильною підтримкою event-time, stateful processing і гарантіями «exactly-once» для стану та виводів, що робить його підходящим для складних реальних застосунків із надчутливими транзакційними вимогами [5]. Хмарні сервіси, такі як Google Dataflow, реалізують модель Apache Beam – єдиний програмний модельний шар, що дозволяє писати одноразову програму, яку можна запускати в batch або streaming режимі й переносити між середовищами. Для українських проектів ці інструменти дають можливість як використовувати open-source стеки на власних дата-центрах, так і запускати керовані сервіси у хмарі залежно від вимог до безпеки й суверенітету даних [3].

Машинне навчання в режимі потокової обробки: онлайн-алгоритми класифікації та кластеризації, адаптація до концептуального дрейфу. Інтеграція ML у поточкові конвеєри – одна з найцікавіших і найскладніших практичних задач. Класичні підходи передбачають періодичне навчання моделей на накопичених батчах, але для сценаріїв із високою швидкістю подій або змінною поведінкою джерел це не підходить: тут потрібні онлайн-алгоритми, які оновлюють параметри по мірі надходження нових прикладів. До прикладів таких методів належать стохастичний градієнтний спуск (SGD) у інкрементальній версії, деревоподібні алгоритми на основі Hoeffding tree, потокові варіанти k-means (streaming k-means), а також спеціалізовані потокові алгоритми для кластеризації, такі як CluStream або DenStream. Окремим викликом є concept drift – зміна розподілу даних у часі: у цьому випадку системи повинні вміти виявляти зміни й автоматично адаптувати моделі (через sliding windows для навчальних даних, за ваговими коефіцієнтами з віддачею ваги новішим прикладам, або через ансамблеві підходи з періодичною заміною слабких моделей). Для практичної реалізації потокового ML існують бібліотеки й платформи, орієнтовані на стрімінг: Apache SAMOA, MOA (Massive Online Analysis) для Java-середовищ і сучасні python-бібліотеки на кшталт River, які дозволяють будувати інкрементальні моделі, виконувати преюнкційне тестування та відслідковувати метрики у поточковому режимі [6]. У контексті України такі підходи важливі для банківських систем реального часу (моніторинг транзакцій

і виявлення шахрайства), для кібербезпеки (детекція аномалій у мережевому трафіку) й для агротехніки, де дані з датчиків можуть змінюватися в залежності від сезонності та погодних умов.

Технічні виклики: масштабованість, fault tolerance, низька затримка і безпека даних. Масштабованість поточкових систем – це не лише питання горизонтального масштабування компонентів, але й правильного проєктування розподілу даних (partitioning), мінімізації «гарячих точок» та ефективного керування станом [1]. Потоки із сотнями тисяч подій в секунду потребують балансування навантаження між нодами, використання партицій у Kafka та контролю backpressure у системі обробки, щоб запобігти падінню продуктивності. Fault tolerance досягається за допомогою механізмів контрольних точок (checkpoints), запису офсетів і реплікації стану; реалізація «exactly-once» семантики у складних трансформаціях часто критична для фінансових застосунків і вимагатиме правильно налаштованої взаємодії між системою доставляння подій (наприклад Kafka) і системою обробки (наприклад Flink). Низька затримка вимагає оптимізації на рівнях мережі, сховища стану та логіки обробки – наприклад, вибір на користь in-memory state backend, зменшення копіювань повідомлень та використання асинхронних інтерфейсів [4]. Безпека даних охоплює шифрування даних у транзиті й у стані, контроль доступу (RBAC), аудит, захист від DDoS й інших атак; для українських державних і комерційних сервісів питання суверенітету даних і відповідності національному законодавству про захист персональних даних – важливий фактор при виборі архітектури (локальне розміщення або сертифіковані хмарні зони). Практики моніторингу продуктивності та безпеки – Prometheus/Grafana для метрик і алертингу, логування з використанням централізованих систем і SIEM – є стандартом для забезпечення надійності та швидкого реагування. Ключовим є те, що кожен шар системи (інгест, обробка, збереження стану, ML-сервіси) повинен мати відповідні SLA та прозору стратегію відновлення після збоїв [7].

Прикладні сценарії: фінанси, IoT, телеком, кібербезпека, Smart City. У фінансовому секторі поточковий аналіз використовується для виявлення шахрайських транзакцій в реальному часі, для антифрод-систем і для моніторингу ризиків; українські банки активно інвестують у системи моніторингу подій і поведінкової аналітики, що дозволяє оперативно блокувати підозрілі операції й мінімізувати втрати. У сфері Інтернету речей поточковий аналіз дозволяє обробляти телеметрію з тисяч датчиків у промисловості й агросекторі. Телекомунікаційні оператори застосовують стрімінговий аналіз для моніторингу якості мережі, виявлення проблем з QoS і оптимізації маршрутизації трафіку; з появою 5G обсяги телеметрії зростають і вимоги до латентності стають жорсткішими. У кібербезпеці поточкові системи лежать в основі SIEM і детекторів аномалій: збір й аналіз подій у реальному часі дозволяє швидко ідентифікувати атакуючі патерни та запускати автоматизовані відповіді; в Україні відповідні ініціативи посилюються у зв'язку з підвищеною кіберзагрозою, і CERT-UA публікує рекомендації щодо виявлення й реагування на інциденти [8].

Архітектурні вибори й практичні поради для реалізації. При проєктуванні поточкових систем важливо вирішити, чи обирати підхід «Lambda» з окремими шляхами batch і streaming або «Kappa», де базу даних подій (event log) обробляється тільки поточною; останнім часом тренд рухається в бік Kappa-архітектури завдяки уніфікації конвеєра й зменшенню складності підтримки двох різних режимів обробки. Практичні рекомендації включають правильний вибір часу вікон (trade-off між узгодженістю й швидкістю реакції), налаштування watermark і lateness, стратегії для стейту (eviction policies, snapshots), організацію CI/CD для поточкових конвеєрів (тести на детермінованість трансформацій), інтеграцію з системами моніторингу і обов'язкове забезпечення можливості відстеження та пояснюваності рішень ML у реальному часі. Для оновлення моделей у продукційному середовищі доцільно розділяти ланцюг inference (низька латентність, готові збережені моделі) і learning (процес оновлення з контрольованою перевіркою), щоб уникнути непередбачуваних деградацій якості внаслідок drift [1].

Регуляторні та етичні аспекти. Будь-яка система, що обробляє персональні дані у режимі реального часу, повинна врахувати вимоги місцевого законодавства та міжнародні норми приватності, зокрема принципи мінімізації даних і прозорості обробки. Для українських організацій важливо дотримуватися національних правил захисту персональних даних, а при роботі з даними європейських громадян – враховувати вимоги GDPR. В аспекті кібербезпеки на перший план виходять питання доступу, шифрування, збереження логів для аудиту та відповідальності за інциденти. Етичні міркування стосуються запобігання алгоритмічним упередженням, прозорості рішень і уникнення надмірної автоматизації, яка може призводити до хибних блокувань або несправедливого обслуговування користувачів [9].

Висновки

Аналіз поточкових даних у реальному часі є одним із ключових напрямів сучасної аналітики, що поєднує досягнення Big Data, розподілених систем та машинного навчання. Розглянуті методи – sliding windows, micro-batch та event-time processing – формують теоретичну основу для побудови ефективних підходів до обробки даних у режимі безперервного надходження. Фреймворки й платформи на кшталт Apache Kafka, Spark Streaming, Apache Flink та Google Dataflow стали базовими інструментами, що дозволяють реалізовувати масштабовані та стійкі конвейери потокової обробки із мінімальною затримкою. Важливим складником розвитку є інтеграція алгоритмів машинного навчання, зокрема онлайн-класифікації та кластеризації, які забезпечують адаптацію до постійно змінних умов і явища концептуального дрейфу.

Таким чином, потокова обробка даних у реальному часі є не просто технологічним трендом, а необхідністю у світі, де дані генеруються з небаченою швидкістю. Подальший розвиток цієї галузі пов'язаний із поглибленою інтеграцією штучного інтелекту, розвитком edge computing та хмарних екосистем, а також з посиленням уваги до питань безпеки й етики. Для України освоєння й розвиток поточкових технологій може стати одним із рушіїв стійкого економічного зростання, підвищення кіберзахисту та побудови сучасних цифрових сервісів для громадян.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Що таке потокова (streaming) база даних? URL: <https://data-life-ua.com/db/shcho-take-potokova-baza-danykh/> (дата звернення: 27.09.2025).
2. Spark Streaming. URL: <https://spark.apache.org/docs/latest/streaming/index.html> (дата звернення: 29.09.2025).
3. Google Dataflow. URL: <https://cloud.google.com/products/dataflow> (дата звернення: 01.10.2025).
4. Apache Kafka. URL: <https://kafka.apache.org/documentation/> (дата звернення: 01.10.2025).
5. Apache Flink. URL: <https://flink.apache.org/what-is-flink/flink-architecture/> (дата звернення: 03.10.2025).
6. Доказова база з потокового ML і concept drift. URL: <https://www.evidentlyai.com/ml-in-production/concept-drift> (дата звернення: 03.10.2025).
7. Prometheus. URL: <https://prometheus.io/docs/introduction/overview/> (дата звернення: 03.10.2025).
8. CERT-UA. URL: <https://cert.gov.ua/> (дата звернення: 04.10.2025).
9. Загальні міркування щодо приватності. URL: <https://gdpr.eu/> (дата звернення: 04.10.2025).

Магденко Анастасія Романівна – студентка групи 1KITC-226, Факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м. Вінниця, e-mail: anastasiamahdenko@gmail.com

Науковий керівник: **Грицак Анатолій Васильович** – кандидат технічних наук, доцент, доцент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, м. Вінниця, e-mail: grytsak.a.v@gmail.com

Mahdenko Anastasiia R. – student of group 1KITS-22b, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: anastasiamahdenko@gmail.com

Supervisor: **Hrytsak Anatoly V.** – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Management and Security of Information Systems Vinnytsia National Technical University, Vinnytsia, e-mail: grytsak.a.v@gmail.com