УДК 681.518.2 DOI: 10.18372/2073-4751.82.20364

Калашник М.О.,
e-mail: 294776@stud.kai.edu.ua

АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ І ТЕХНОЛОГІЙ ОБРОБКИ ВЕЛИКИХ ДАНИХ У РЕАЛЬНОМУ ЧАСІ В ІНФРАСТРУКТУРАХ ІНТЕРНЕТУ РЕЧЕЙ (IoT)

Державний університет «Київський авіаційний інститут»

Постановка проблеми

Зростання кількості пристроїв Інтернету речей (*IoT*) супроводжується експоненційним збільшенням обсягів даних, що генеруються в режимі реального часу. Сенсори, контролери, камери, мобільні платформи й інші розподілені джерела даних створюють потоки, які потребують негайної обробки для забезпечення коректної та безпечної роботи критичних систем. У галузях охорони здоров'я, розумного міського управління, промислової автоматизації й безпеки дані мають бути оброблені з мінімальною затримкою – часто в межах мілісекунд – що створює високі вимоги до обчислювальних інфраструктур.

Існуючі централізовані рішення обробки даних, зокрема хмарні обчислення, не завжди здатні забезпечити необхідний рівень швидкодії та надійності в умовах обмеженого пропускного каналу, нестабільного підключення або високої частоти подій. Це спонукає до використання гібридних підходів – зокрема *edge*- і *fog*-обчислень – що дозволяють виконувати попередню обробку даних ближче до джерела їхнього виникнення.

Незважаючи на різноманіття технологій потокової обробки, на практиці існує низка труднощів у виборі оптимальної комбінації платформи, архітектури та аналітичної моделі. Проблеми масштабованості, відмовостійкості, синхронізації подій та узгодженості даних залишаються відкритими, як і питання адаптації алгоритмів машинного навчання до вимог реального часу, особливо в умовах гетерогенних *IoT*-мереж. У цьому контексті особливо актуальним є проведення системного аналізу існуючих методів і технологій обробки великих даних у реальному часі з урахуванням специфіки *IoT*-середовищ, що дозволить ідентифікувати сильні та слабкі сторони сучасних рішень і надати орієнтири для вибору відповідних технологій залежно від прикладного контексту.

Формулювання мети дослідження

Метою даного дослідження є проведення системного аналітичного огляду сучасних методів і технологій обробки великих даних у реальному часі, що застосовуються в інфраструктурах Інтернету речей. У фокусі дослідження – порівняння архітектурних підходів (*Edge*, *Fog*, *Cloud*), аналіз можливостей провідних платформ потокової обробки (*Apache Kafka*, *Flink*, *Storm*, *Spark Streaming*), а також вивчення застосування моделей машинного навчання для задач виявлення аномалій та аналізу поведінки пристроїв. Дослідження спрямоване на виявлення сильних і слабких сторін існуючих рішень, визначення ключових критеріїв вибору технологій з урахуванням обмежень *IoT*-середовища (затримка, продуктивність, масштабованість, енергоспоживання).

Аналіз останніх досліджень і публікацій

Останні дослідження демонструють значний прогрес у розробці методів і технологій обробки великих даних у реальному часі в інфраструктурах Інтернету речей. Науковці активно працюють над створенням ефективних рішень для управління потоками даних *IoT*, покращенням архітектур розподілених обчислень, енергоефективністю та безпекою.

У роботі Андріуло та ін. [1] представлено систематичний огляд підходів до *Edge* та *Cloud Computing* в контексті *IoT*. Дослідники показали, що гібридні

архітектури забезпечують баланс між обчислювальною потужністю хмари та низькою затримкою периферійної обробки, що є критично важливим для застосунків у реальному часі.

Розширення цього напряму запропонували Ортіс та співавт. [2], які розробили контекстно-орієнтовану архітектуру «*Atmosphere*» для інтеграції *Edge*, *Fog* і *Cloud* обчислень. Запропоноване рішення дозволяє динамічно адаптувати поведінку *IoT*-систем до змін умов середовища, забезпечуючи гнучкість і масштабованість обробки даних.

Значну увагу в дослідженнях приділяють застосуванню машинного навчання для виявлення аномалій у потокових *IoT*-даних. Так, Аргубі та ін. [3] представили адаптивну систему виявлення вторгнень для *IoT*-мереж, засновану на динамічному налаштуванні моделей до змін в інформаційному середовищі.

Подібні підходи досліджували й автори статті в *Scientific Reports* [4], зосередившись на виявленні аномалій в *IoT*-системах охорони здоров'я. Їхні результати демонструють ефективність застосування глибоких нейронних мереж у реальному часі з точністю класифікації понад 90%.

На розподілену обробку даних без централізованого збору орієнтована робота Геленбе, Гюля та Накіпа [5], які запропонували самонавчальний алгоритм виявлення вторгнень. Його головна перевага – збереження приватності та зменшення навантаження на мережу, що є надзвичайно актуальним для сучасних розподілених *IoT*-інфраструктур.

У галузі енергоефективної обробки даних Дюш та ін. [6] представили енергозберігаюче рішення для аналізу біосигналів на носимих пристроях. Система виконує обробку даних безпосередньо на пристрої, що значно знижує енергоспоживання та забезпечує можливість тривалої автономної роботи.

Ще один перспективний напрям досліджень – інтелектуальне управління туманними обчисленнями. У роботі Фрьоліха та ін. [7] представлено підхід до управління *fog*-сервісами з використанням програмно-детермінованих мереж (*SDN*), який дозволяє оптимізувати як якість обслуговування (*QoS*), так і енергетичну ефективність систем у режимі реального часу.

Проведений аналіз останніх досліджень свідчить, що основними напрямками розвитку технологій обробки великих даних у реальному часі в інфраструктурах Інтернету речей є: оптимізація архітектур розподілених обчислень, розвиток *Edge* та *Fog Computing*, підвищення енергоефективності, забезпечення захисту та приватності даних, а також впровадження сучасних алгоритмів машинного навчання. Водночас актуальною залишається потреба у комплексному підході, який би поєднував переваги різних технологій для створення масштабованих, надійних та ефективних *IoT*-систем.

Основна частина

Інфраструктури Інтернету речей формують середовище постійної генерації гетерогенних даних, що надходять у вигляді неперервних потоків з різноманітних джерел: сенсорів температури, тиску, вологості, контролерів промислового обладнання, *GPS*-трекерів, камер відеоспостереження, дронів, *RFID*-зчитувачів, носимих пристроїв тощо. Ці потоки мають суттєві особливості порівняно з традиційними статичними наборами даних.

По-перше, потоки *IoT*-даних характеризуються високою частотою надходження – від декількох повідомлень на секунду у споживчій електроніці до мільйонів подій щохвилини у промислових та міських системах. Обсяг інформації, що передається, залежить як від типу джерела, так і від режиму його роботи (періодична передача, подієва, за розкладом тощо). Наприклад, потік даних з відеокамери формує мегабайти за секунду, тоді як датчик температури – лише кілька байтів щохвилини.

По-друге, ці дані часто мають неоднорідну структуру, що змінюється в залежності від контексту застосування. Наприклад, в аграрному *IoT* застосовуються прості сенсори з передбачуваною структурою

повідомлень, тоді як у розумних містах потоки включають комплексні *JSON*-повідомлення з вкладеними об'єктами.

Важливим є також вимір латентності: більшість прикладних сценаріїв (моніторинг аварій, детекція аномалій, реагування в медичних пристроях) вимагають обробки із затримкою менш ніж одна секунда [6]. Це накладає суттєві вимоги до обчислювальних інфраструктур: висока доступність, відмовостійкість, низька затримка, здатність до масштабування.

Однією з найсерйозніших проблем залишається синхронізація та консистентність подій, особливо коли йдеться про багатоджерельні потоки з часовими розривами або втратами пакетів. Наприклад, для виявлення аномалії у виробничій лінії часто необхідно зіставити події з кількох сенсорів, які мають різний часовий лаг та можуть надсилати некоректні або неповні дані. У таких випадках важливою стає наявність буферів, механізмів часової нормалізації та політик злиття даних.

Крім того, специфікою *IoT* є непередбачуваність появи нових джерел даних, що зумовлює необхідність гнучкості у масштабуванні каналів обробки та зберігання.

Побудова ефективної системи обробки *IoT*-даних в реальному часі вимагає ретельно спроектованої архітектури, яка повинна враховувати як фізичні характеристики пристроїв, так і логіку аналітичних задач. Залежно від цілей, використовують *edge*-, *fog*- та *cloud*-архітектури [1].

Edge-архітектура передбачає виконання частини обробки даних безпосередньо на пристроях або поблизу них – наприклад, на гейтах чи маршрутизаторах. Це дозволяє скоротити затримку, зменшити навантаження на мережу, забезпечити автономність у разі втрати зв'язку з центральним сервером. Проте, обчислювальні ресурси *edge*-пристроїв обмежені, що унеможливлює виконання складної аналітики або глибокого навчання.

Fog-архітектура реалізує проміжний рівень між хмарою та периферією – це можуть бути локальні сервери, які обслуговують групу пристроїв [2]. Такий підхід забезпечує баланс між затримкою та обчислювальною потужністю, а також дозволяє виконувати попередню обробку, фільтрацію, агрегування перед відправкою в хмару.

Хмарна архітектура надає найбільші можливості для глибокої аналітики, зберігання історичних даних та інтеграції з *ML*-платформами. Проте, її недоліками є вища затримка, залежність від стабільного з'єднання та більша вразливість до атак під час передавання даних.

Lambda-архітектура, що поєднує потокову обробку (*speed layer*) та пакетну обробку (*batch layer*), є класичним рішенням для задач, де необхідна як швидка реакція на події, так і ретроспективна аналітика. Вона дозволяє зберігати точність при довготривалому аналізі, але вимагає подвійного програмування – окремо для обох рівнів, що ускладнює підтримку системи.

Kappa-архітектура пропонує альтернативу, де всі обчислення реалізуються через обробку потоків. Це спрощує розробку і забезпечує меншу затримку, проте не підходить для задач, де критичним є ретроспективний аналіз повного набору даних.

Окрему нішу займає концепція *Streaming-as-a-Service*, яка передбачає використання керованих сервісів (наприклад, *Amazon Kinesis*, *Google Cloud Dataflow*, *Azure Stream Analytics*) для обробки потоків за підпискою. Це знижує вартість початкового впровадження та підвищує доступність технологій для невеликих компаній, хоча може викликати залежність від постачальника (*vendor lock-in*) і обмеження в налаштуваннях.

Загалом, вибір архітектури повинен ґрунтуватися на поєднанні вимог до затримки, складності обробки, географічного розташування пристроїв та обмежень щодо ресурсів і безпеки.

Ефективна реалізація аналітики потокових *IoT*-даних у реальному часі значною мірою залежить від обраної платформи. Найбільш поширені серед них – *Apache Kafka*, *Apache Flink*, *Apache Storm* та *Spark Streaming* – пропонують різні

рівні функціональності, масштабованості та затримки.

Apache Kafka є системою обміну повідомленнями, побудованою за принципом *publish-subscribe*. Вона виконує роль буфера між джерелами даних та обчислювальними модулями, забезпечуючи зберігання та впорядкування подій у реальному часі. *Kafka* має високу пропускну здатність (до декількох мільйонів повідомлень за секунду) та горизонтальну масштабованість, що робить її основою для більшості сучасних стримінгових архітектур.

Apache Flink – це платформа для розподіленої обробки потоків і пакетів даних, що підтримує складні типи подій, часові вікна, контроль збереження стану та *fault tolerance* через механізми *checkpointing*. *Flink* відзначається низькою латентністю (< 100 мс), що критично для застосувань у сфері охорони здоров'я або промислового моніторингу.

Apache Storm забезпечує обробку потоків з дуже низькою затримкою, але має обмежені можливості в роботі зі складними операціями, такими як злиття потоків чи агрегування за часовими вікнами. Через це *Storm* часто використовується у сценаріях, де домінує швидкість над складністю аналітики.

Spark Streaming базується на мікропакетній обробці (*micro-batch*) та інтегрується з екосистемою *Apache Spark*. Його перевага – у простоті використання та гнучкості, особливо у випадках, де потрібна інтеграція потоків із машинним навчанням або *SQL*-аналітикою.

На рис. 1 наведено порівняння ключових характеристик платформ.

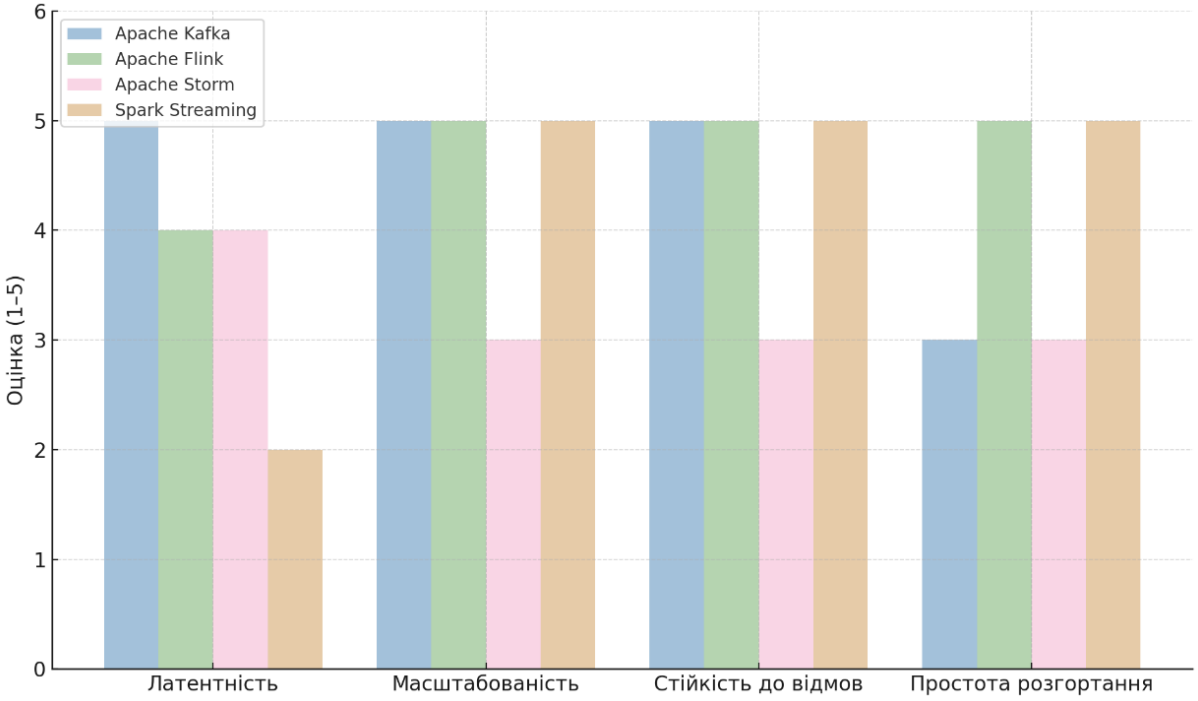

Рис. 1. Порівняння ключових характеристик платформ

Параметри оцінювання (шкала: 1 – низький рівень, 5 – найвищий рівень).

Потокова аналітика *IoT* вимагає не лише транспортної інфраструктури, а й ефективної обробки самих даних: фільтрації, нормалізації, агрегації, класифікації, виявлення аномалій тощо. На етапі передобробки застосовуються методи фільтрації за порогами, дедуплікації подій, хешування, розбиття на часові вікна (*tumbling*/*sliding windows*), а також алгоритми згладжування, що знижують шум. Наприклад, для сенсорів температури можуть використовуватись фільтри Калмана або експоненційне згладжування.

Для виявлення аномалій у потоках даних найчастіше застосовуються методи машинного навчання:

- *Isolation Forest* – для швидкого виявлення рідкісних подій;
- *LSTM*-рекурентні нейромережі – для обробки часових залежностей;
- *AutoEncoder* – для відновлення типових зразків і виявлення аномалій на основі похибки реконструкції;
- *Streaming k-means* – для динамічної кластеризації нових точок [3].

Важливу роль відіграє *Complex Event Processing* (*CEP*) − методологія, що дозволяє формалізувати складні шаблони подій. Наприклад, система може виявити закономірність "зниження напруги на 10% + перегрів > $70^{\circ}C$ → аварійний режим", і сигналізувати про потенційну несправність.

Актуальним є і адаптивне навчання (*online learning*), яке дозволяє моделі постійно оновлювати свої ваги без потреби у повному перенавчанні. Це особливо важливо для довготривалих *IoT*-систем зі змінним середовищем [5].

Застосування потокової обробки в *IoT* поширене у багатьох доменах, де необхідна миттєва реакція на зміну параметрів:

Розумні міста. У проектах з моніторингу трафіку, наприклад у Сінгапурі, використовується *Flink+Kafka* для виявлення заторів у реальному часі на основі *GPS*-даних з тисяч транспортних засобів. У системах моніторингу якості повітря платформи *Spark Streaming* аналізують показники вмісту *PM*2.5 з розподілених сенсорів для миттєвого виявлення забруднення.

Промисловий *IoT*. Заводи *Siemens* застосовують *edge-Flink* для діагностики обладнання: дані з сенсорів температури, вібрації та навантаження обробляються безпосередньо на гейтах, а виявлені аномалії передаються в хмару для подальшого аналізу.

Медичні системи. У проектах цифрового здоров'я (наприклад, телеметрія пацієнтів з кардіомоніторами) застосовуються моделі на основі *LSTM* та *AutoEncoder*, які виявляють відхилення у серцевому ритмі в реальному часі, що дозволяє швидко реагувати на аритмії.

Ці приклади демонструють здатність сучасних технологій забезпечувати масштабовану, низьколатентну та надійну обробку даних у критичних застосуваннях.

Проблема масштабування залишається ключовою у великих гетерогенних системах з десятками тисяч вузлів. Стримуючим фактором є як обмеженість каналів зв'язку, так і складність узгодження потоків від численних джерел.

Безпека потокових даних набуває особливого значення через постійну передачу незашифрованої інформації. Існує потреба в інтеграції легких протоколів шифрування (наприклад, *DTLS*) та багаторівневої аутентифікації пристроїв, які працюють у реальному часі, не втрачаючи продуктивності.

Енергоспоживання також стає критичним, особливо в *edge*-сценаріях, де пристрої працюють від батарей. Застосування важких моделей глибокого навчання вимагає оптимізації обчислень – як за рахунок квантованих моделей, так і через компресію архітектур.

Інтеграція з існуючими системами часто викликає труднощі, оскільки старіші протоколи або не стандартизовані формати не сумісні з сучасними платформами потокової обробки. У таких випадках необхідна розробка адаптерів, що знижує гнучкість.

Вирішення зазначених викликів є пріоритетом сучасних досліджень у галузі *IoT*-аналітики.

Висновки

У статті проведено аналітичний огляд сучасних методів і технологій обробки великих даних у реальному часі в інфраструктурах Інтернету речей. Розглянуто основні характеристики потокових даних *IoT*-систем, включно з високою частотою оновлення, мінливістю структури та вимогами до низької затримки та відмовостійкості. Проведено порівняльний аналіз архітектур *edge*-, *fog*- та *cloud*-комп'ютингу, а також підходів *Lambda* та *Kappa*, що дозволило визначити переваги й недоліки кожної з них у контексті реального часу.

Описано функціональні можливості провідних платформ потокової обробки,

таких як *Apache Kafka*, *Flink*, *Storm* та *Spark Streaming*, з урахуванням їхньої масштабованості, латентності та стійкості до відмов. Розглянуто сучасні методи обробки даних у потоці – фільтрацію, виявлення аномалій, *Complex Event Processing* та *online learning* – що забезпечують адаптивність і точність аналізу. Наведено приклади практичного застосування технологій у сферах *smart city*, промислового моніторингу та медичного *IoT*, що підтверджує їхню ефективність у вирішенні завдань оперативного реагування.

Узагальнення результатів дозволило виокремити ключові виклики, зокрема – масштабування, енергоспоживання, безпеку даних та інтеграцію з гетерогенними системами. Отримані висновки можуть стати основою для формування рекомендацій щодо вибору оптимальних архітектур і технологій обробки *IoT*-даних у реальному часі. Перспективними напрямами подальших досліджень є розробка гібридних підходів до обробки, удосконалення методів адаптивного навчання та впровадження легких криптографічних протоколів для захисту потокових даних.

Література

1. Andriulo F. C. et al. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics*. 2024. Vol. 11, no. 4. P. 71. DOI: 10.3390/informatics11040071.

2. Ortiz G. et al. Atmosphere: Context and Situational-Aware Collaborative IoT Architecture for Edge-Fog-Cloud Computing. arXiv preprint. 2024. URL: https://arxiv.org/abs/2401.14968.

3. Rguibi A. et al. Adaptive Machine Learning-Based Intrusion Detection Systems for IoT Networks: A Dynamic Approach. Lecture Notes in Networks and Systems. Vol. 1353. Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment ICAISE'2024, Volume 1 / ed. by Y. Farhaoui et al. Cham, 2025. P. 101–106. DOI: 10.1007/978-3-031-88304-0_14.

4. Khan M. M., Alkhathami M. Anomaly Detection in IoT-Based Healthcare: Machine Learning Approaches. *Scientific Reports*. 2023. Vol. 13. 56126. DOI: 10.1038/s41598-024-56126-x.

5. Gelenbe E., Gül B. C., Nakip M. DISFIDA: Distributed Self-Supervised Federated Intrusion Detection Algorithm with Online Learning for Health Internet of Things and Internet of Vehicles. *Internet of Things*. 2024. Vol. 28. 101340. DOI: 10.1016/j.iot.2024.101340.

6 Duch L. et al. HEAL-WEAR: An Ultra-Low Power Heterogeneous System for Bio-Signal Analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2017. Vol. 64, no. 9. P. 2448–2461. DOI: 10.1109/TCSI.2017.2701499.

7. Fröhlich P. et al. Smart SDN Management of Fog Services to Optimize QoS and Energy. *Sensors*. 2021. Vol. 21, no. 9. 3105. DOI: 10.3390/s21093105.

Калашник М.О.

АНАЛІТИЧНИЙ ОГЛЯД МЕТОДІВ І ТЕХНОЛОГІЙ ОБРОБКИ ВЕЛИКИХ ДАНИХ У РЕАЛЬНОМУ ЧАСІ В ІНФРАСТРУКТУРАХ ІНТЕРНЕТУ РЕЧЕЙ (IoT)

У статті проведено аналітичний огляд сучасних методів і технологій обробки великих даних у реальному часі в інфраструктурах Інтернету речей (IoT). Розглянуто джерела потокових даних, особливості їхньої структури, мінливість та вимоги до затримки обробки. Проаналізовано переваги та обмеження архітектур Edge, Fog, Cloud, а також підходів Lambda та Kappa для побудови високопродуктивних IoT-систем. Окрема увага приділена платформам потокової обробки даних – Apache Kafka, Flink, Storm, Spark Streaming – з оцінкою їхньої масштабованості, відмовостійкості та зручності впровадження. Висвітлено сучасні методи виявлення аномалій у потоках даних, зокрема

на основі AutoEncoder, LSTM, Isolation Forest, та застосування CEP для аналізу складних подій. Представлено приклади практичного застосування технологій у системах «розумного міста», промислової автоматизації та медицини. Узагальнено виклики та напрями подальших досліджень щодо покращення безпеки, адаптивності та ефективності реального часу в гетерогенних середовищах IoT.

Ключові слова: *Інтернет речей; великі дані; потокова аналітика; обробка в реальному часі; Apache Kafka; Flink; виявлення аномалій; Edge/Fog/Cloud; адаптивне навчання; Complex Event Processing.*

Kalashnyk M.O.

ANALYTICAL REVIEW OF METHODS AND TECHNOLOGIES FOR REAL-TIME BIG DATA PROCESSING IN IoT INFRASTRUCTURES

This paper presents an analytical review of modern methods and technologies for real-time big data processing in Internet of Things (IoT) infrastructures. It explores data stream sources, structural variability, and the key processing requirements such as latency, fault tolerance, and continuous availability. The advantages and limitations of Edge, Fog, and Cloud architectures are discussed, along with Lambda and Kappa approaches for building high-performance IoT systems. Special attention is given to stream processing platforms – Apache Kafka, Flink, Storm, and Spark Streaming – with an evaluation of their scalability, fault tolerance, and ease of deployment. The study highlights state-of-the-art anomaly detection methods in streaming data, including AutoEncoder, LSTM, and Isolation Forest, as well as the use of Complex Event Processing (CEP) for composite event analysis. Real-world applications in smart city systems, industrial automation, and healthcare are examined. The article summarizes current challenges and outlines directions for future research on improving security, adaptability, and efficiency in heterogeneous real-time IoT environments.

Keywords: *Internet of Things; big data; stream analytics; real-time processing; Apache Kafka; Flink; anomaly detection; Edge/Fog/Cloud; adaptive learning; Complex Event Processing.*