

УДК 004.04

DOI: 10.31673/2786-8362.2026.011628

Ткаченко О.М., д.т.н; Шевченко А.Т.

АРХІТЕКТУРНИЙ ПІДХІД ПРОГРАМНОЇ ОБРОБКИ СИГНАЛІВ В ПОТОКОВИХ СИСТЕМАХ

Tkachenko O.M., Shevchenko A.T. Architectural approach to real-time processing in streaming systems. Internet of Things systems are characterized by heterogeneous large-scale data streams that arrive in real-time from distributed source and require their reliable and operational processing. This paper considers an architectural approach focused on real-time data processing that treats scalability and reliability as system-level properties of the data infrastructure. The proposed approach treats real-time signal processing as an integral component of the system's streaming architecture, functionality linked to event flow control, time semantic, and resource constraints execution environment. Accordingly, data flow management and signal modeling are integrated within a unified data platform that supports streaming, quality control, scalable processing, and storage. The approach formulates system requirements for models in streaming environments, considering the stochastic event nature, irregular time intervals, data quality, and resource constraints of a multi-tiered architecture. The behavior of the model is interred as an architecturally determined performance characteristic. The concept forms a methodological basis for the design of analytical components of real time IoT systems.

Keywords: Internet of Things, real-time signal processing, streaming data architectures, signal modeling in streaming systems, distributed analytics systems

Ткаченко О.М., Шевченко А.Т. Архітектурний підхід програмної обробки сигналів в потокових системах. У статті запропоновано архітектурний підхід до інтеграції аналізу сигналів у платформу обробки потокових даних, де моделювання розглядається як компонент, пов'язаний із керуванням подіями, часовою семантикою та ресурсними обмеженнями виконання для забезпечення стійкості на рівні системи. Сформульовано системні вимоги до моделей у потокових середовищах та обґрунтовано їх архітектурну зумовленість як основу проєктування аналітичних компонентів у системах IoT.

Ключові слова: Інтернет речей, обробка сигналів у реальному часі, архітектури потокової обробки даних, моделювання сигналів у потокових системах, розподілені аналітичні системи

Вступ

Зростаюча кількість розгортань систем Інтернету речей призводить до створення масштабних, високошвидкісних та неоднорідних потоків даних, тим самим позиціонуючи дані, згенеровані такими системами у парадигмі аналітики великих даних, демонструючи їх визначальні характеристики включаючи великий обсяг, швидкість, різноманітність та мінливість [1]. Сучасні системи інтернету речей створюють гетерогенні потоки даних, що надходять від пристроїв сприйняття, що актуалізує питання якості та придатності таких даних для аналізу [2]. Розгортання систем Інтернету речей у таких областях як промисловий та екологічний моніторинг, розумні міста та інтелектуальні транспортні системи [3] збільшує попит на масштабовані рішення обробки даних у реальному часі. На відміну від традиційної пакетної обробки середовища IoT дедалі частіше вимагають безперервної обробки генерованих сигнальних даних з обмеженою затримкою для підтримки своєчасного прийняття рішень.

Потоки даних Інтернету речей характеризуються стохастичною мінливістю, невизначеністю вимірювання, відсутніми спостереженнями та затримками. Однак, хоча алгоритмічні вдосконалення в аналітиці даних Інтернету речей продовжують вивчатися, проте менше уваги приділяється архітектурним та системним умовам, необхідним для їх надійного розгортання, що створює розрив між теоретичними моделями та їхньою реалізацією в реальних умовах. Проте варто зауважити, що також спостерігаються тенденції до впровадження розподілених технологій притаманних до аналітики в реальному часі для великих даних [4]. Тому дане дослідження має на меті сформулювати базові принципи архітектурного підходу до інтеграції моделей аналізу сигналів у системах обробки потокових даних, що узгоджений з властивостями розгортання потокових систем обробки даних.

На високому рівні архітектуру IoT можна представити як системи, що складаються з сенсорних пристроїв, комунікаційних рівнів та хмарних платформ [5]. Хоча така модель добре пояснює схему підключення та розгортання вона не розкриває архітектурну структуру та логіку функціонування платформи даних. У масштабних розгортаннях систем Інтернету речей рівень платформи даних є розподіленою потоковою інфраструктурою відповідальною за прийом, управління, обчислення та збереження всіх даних, забезпечуючи семантику часу подій, масштабованість та стійкість до відмов.

Пристрої інтернету речей зазвичай генерують неоднорідні потоки даних, як з точки зору форматів, так і протоколів зв'язку [6]. Забезпечення структурної та семантичної узгодженості між розподіленими джерелами вимагає спеціалізованих механізмів попередньої обробки, включаючи процедури синхронізації, перевірки та нормалізації. Окрім структурної неоднорідності, дані датчиків часто недосконалі: вимірювання можуть містити відсутні значення, викиди, аномальні спостереження та шум, спричинений перешкодами навколишнього середовища, деградацією обладнання або помилками передачі. Якщо такі недоліки не враховувати безпосередньо, вони спотворюють аналітичні результати та порушують стабільність моделі. Тому системи реального часу повинні підтримувати розподілене виконання, масштабовану обробку та часово угоджену інтеграцію неоднорідних даних. Аналітика не може підтримуватися виключно поточними обчисленнями. Для забезпечення збереження історії, відтворюваності та керованості необхідна структурована платформа даних. У сучасних архітектурах дані організуються у концептуальні шари, що включають зберігання необроблених подій, структуровані аналітичні представлення та оптимізовані для стабільного споживання сховища даних. Також поєднання цих підходів знаходить своє відображення у Lakehouse-архітектурах [7], зокрема у вигляді багаторівневих моделей обробки, наприклад у вигляді медальйон архітектури [8]. Отже, правильне проектування платформи даних стає критичним фактором у визначенні того чи може аналіз сигналів в реальному часі працювати надійно та послідовно на всіх рівнях розгортання.

Оскільки системи Інтернету речей в реальному часі функціонують в умовах стохастичного надходження подій, нерегулярних інтервалів вимірювання, затримок, дублювання та невпорядкованої доставки в розподілених асинхронних умовах, на відміну від класичної цифрової обробки сигналів, яка передбачає регулярну вибірку та детерміноване виконання, аналіз сигналів на основі потокової передачі слід розглядати як інтегрований компонент рівня обробки даних, поведінка якого залежить як від властивостей вхідного сигналу до системи, так і від архітектурних характеристик системи.

Метою статті є розробка архітектурного підходу, що передбачає структуровану платформу даних, що враховує аналіз та обробку сигналів, для забезпечення високоякісної аналітики в режимі реального часу в системах Інтернету речей. Замість розгляду IoT лише як платформи зв'язку між пристроями, пропонується розглядати систему як цілісне інтегроване середовище життєвого циклу даних, де гарантії потокової передачі безпосередньо підтримують обробку сигналів у режимі реального часу та дозволяють подальшу імплементацію розширення аналітики на основі історичних даних та подальше впровадження для аналізу таких даних рішення на основі машинного та глибокого навчання.

Виклад основного матеріалу дослідження

Моделювання сигналів у реальному часі з обмеженнями архітектури. З огляду на те, що аналіз сигналів у поточних середовищах доцільно розглядати не як ізольований алгоритмічний модуль, а як інтегрований компонент поточного шару обробки даних, його виконання повинно бути узгоджене з механізмами надходження подій, часової інтерпретації, буферизації та розгортання в розподіленому середовищі. У такій постановці задачі властивості алгоритму аналізу повинні зберігатися за варіацій інтенсивності потоку та конфігурацій розгортання, за умови дотримання визначених системних обмежень. Інтегруючи механізми валідації, синхронізації та контролю якості безпосередньо в поточну інфраструктуру, система

визначає робочі умови за яких адаптивна фільтрація, рекурсивна оцінка виявлення аномалій можуть надійно функціонувати в недосконалих реальних середовищах. Така інтеграція підвищує стійкість систем Інтернету речей, орієнтованих на прийняття рішень та узгоджує моделювання сигналів з архітектурними характеристиками базової платформи даних.

Формалізовано можемо визначити в найпростішому вигляді потокове середовище як систему, що працює з послідовністю подій:

$$\mathcal{E} = \{e_k\}, e_k = (d_k, \tau_k, a_k, y_k),$$

Де d_k – ідентифікатор джерела, τ_k – час виникнення події, a_k – час її надходження до обчислювальної інфраструктури, y_k – вимірне значення. Варто зазначити, що

$$a_k = \tau_k + \delta_k,$$

Де $\delta_k \geq 0$ – затримка доставки. Порядок подій за processing time може не збігатись з порядком за event-time. Потоковий конвеєр можемо розглядати як оператор:

$$\mathcal{P}: \mathcal{E} \rightarrow \mathcal{I},$$

що відображає множину вхідних подій \mathcal{E} у множину аналітичних представлень \mathcal{I} . Також можемо передбачати, що обчислення можуть використовувати певні внутрішні стани, що визначаються відповідно до заданих правил оновлень:

$$s_{k+1} = \mathcal{F}(s_k, E_t),$$

де E_t – деяка підмножина подій \mathcal{E} , що обробляються згідно з політикою виконання в конкретний момент часу, s_{k+1} та $s_k \in \mathcal{S}$ – внутрішні стани. Таким чином, аналіз сигналів у потоковому середовищі, розглядається як частина оператора \mathcal{P} , а його поведінка визначається не лише значеннями y_k , але і особливістю надходження подій.

Як вже було зазначено, дані, що надходять від пристроїв інтернету речей характеризуються структурною неоднорідністю, якість даних створює додатковий виклик. Сенсорні вимірювання можуть містити пропуски, викиди, аномальні спостереження та шум, спричинені впливом середовища, апаратними обмеженнями, деградацією обладнання або помилками передачі в мережі. Тож можемо говорити, що у середовищах інтернету речей недосконалості даних є невід'ємними властивостями, які варто брати до уваги при моделюванні сигналів. Таким чином, доцільно розглядати підхід, де замість стандартної інтеграції логіки жорсткої перевірки обробки, якість даних слід моделювати як структуровані метадані, або як певного роду метрики, де кожна подія може мати свої маркери якості, наприклад у найпростішому випадку:

$$e_k = (d_k, t_k, y_k, q_k),$$

де d_k – ідентифікатор джерела, t_k – мітка часу події, y_k – вимірне значення, q_k – набір маркерів якості, що потенційно може включати ознаки запізнення, порушення порядку ступінь не типовості вимірювання тощо. Узагальнено q_k може бути представлений як опис стану довіри до конкретної події. У системах Інтернету речей, орієнтованих на прийняття рішень, важливо враховувати якість даних та відповідно розглядати чинники, які є невід'ємними від явища спостереження та включати їх явно до моделювання.

Інтеграція моделювання в потокове середовище. Подолання наявних викликів вимагає інтегрованого архітектурного підходу в якому поєднуються механізми інженерії даних та моделі програмного аналізу сигналів у реальному часі. Пропонується розглядати обробку сигналів не як окремий ізольований та незалежний від інфраструктури крок, а як частина потокового рівня, що забезпечує часову когерентність, обробку обмеженої затримки та контрольоване керування управлінням неповними або пошкодженими вимірюваннями.

Таким чином відповідно до представленої моделі з припущенням, що модель може включати внутрішні стани \mathcal{S} для імплементації правил оновлень, потоковий конвеєр є оператором з внутрішнім станом:

$$\mathcal{P}: (\mathcal{E}, s_0) \rightarrow (\mathcal{L}, s_k).$$

Варто зауважити, що оскільки попередньо введена змінна E_t , що є підмножиною подій \mathcal{E} , доступних у момент часу t , та є визначеною політикою виконання, E_t є залежною не від природи сигналу, а від політики виконання обчислень платформою, що тісно інтегрує моделювання обробки сигналу з середовищем виконання демонструючи архітектурну залежність моделі.

Нехай алгоритмічна модель представлена як:

$$M: (s_k, y_k) \rightarrow x_k,$$

тоді в потоковому середовищі, відповідно

$$x_k = M(s_k, E_t),$$

де $x \in \mathcal{X}$ – множина можливих вихідних оцінок моделі.

Таким чином модель обробки сигналів працює не з окремим значенням y_k , а з множиною подій, доступних відповідно до семантики виконання. Поведінка моделі визначається не лише розподілом вхідних значень y_k , а і розподілом δ_k , політикою формування E_t та механізмами оновлення S_k .

У такій постановці модель сигналу функціонує всередині архітектурного середовища, взаємодіючи з механізмами доставки подій, управління станом включаючи часову семантику. Послідовність цієї взаємодії між джерелом подій потоковою платформою та моделлю сигналу схематично зображена на рисунку 1. На рисунку показано, як генерація подій, стохастичні затримки, часова семантика та управління станом взаємодіють у потоковій платформі, формуючи контекст виконання, в якому інтегровані моделі сигналів працюють як компонент платформи.

З огляду на викладені особливості поточкових платформ, стохастичну природу надходження подій та вплив архітектурних рішень на поведінку системи пропонується підхід, що буде враховувати як особливості програмного аналізу так і архітектурних обмежень. Тож таким чином визначається підхід, у межах якого проектування, аналіз та оцінка моделей виконуються з явним врахуванням операційних характеристик потокової системи, в якій вони розгортаються. За такого підходу поведінка моделі розглядається не як ізольована алгоритмічна задача, а як характеристика системного рівня, що формується в межах системи аналізу даних.

Системні характеристики моделювання сигналів у реальному часі. В запропонованому архітектурному підході моделі програмного аналізу сигналів працюють в межах конкретної потокової системи, тому їх поведінка також обумовлена набором системних характеристик, які визначають операційний контекст виконання, які не є деталями реалізації, а являють собою структурні властивості поточкових середовищ Інтернету речей, які в свою чергу безпосередньо впливають на припущення моделювання. Таким чином, з боку моделювання це накладає вимоги на впроваджувану модель, щоб залишатися коректною за змін основних системних характеристик, що схематично зображено на рисунку 2.

На відміну від класичної цифрової обробки сигналів, де часто передбачається рівномірна сітка дискретизації, потокові середовища піддаються стохастичному процесу надходження. Інтервали між надходженнями можуть змінюватися, можуть виникати сплески подій, а вимірювання можуть затримуватися, перевпорядковуватися, дублюватися, або бути відсутніми. Тож в результаті модель не повинна спиратися на фіксовані інтервали дискретизації як структурне припущення, а часова мінливість стає частиною контексту моделювання.

Обробка в реальному часі в системах Інтернету речей зазвичай виконується в розподілених та асинхронних середовищах. Обчислення можуть відбуватися на кількох вузлах зі змінною

затримкою та можливою зміною конфігурацій під час виконання. Динаміка стану моделі може залежати від прийнятих стратегій буферизації, логіки розподілу обчислень та механізмів відновлення після збоїв. У наслідок чого ідентичні події можуть оброблюватися по-різному. Таким чином, еволюцію стану моделі можна розглядати не лише як функцію послідовних подій але й архітектурних параметрів виконання. До того ж, потокові платформи використовують явну семантику часу, таку як моделі часу подій (event-time model) – коли обробка базується на часовій мітці, або часу обробки (processing-time order) – коли використовується час фактичного надходження події до системи, концепція схематично представлена на рисунку 3.

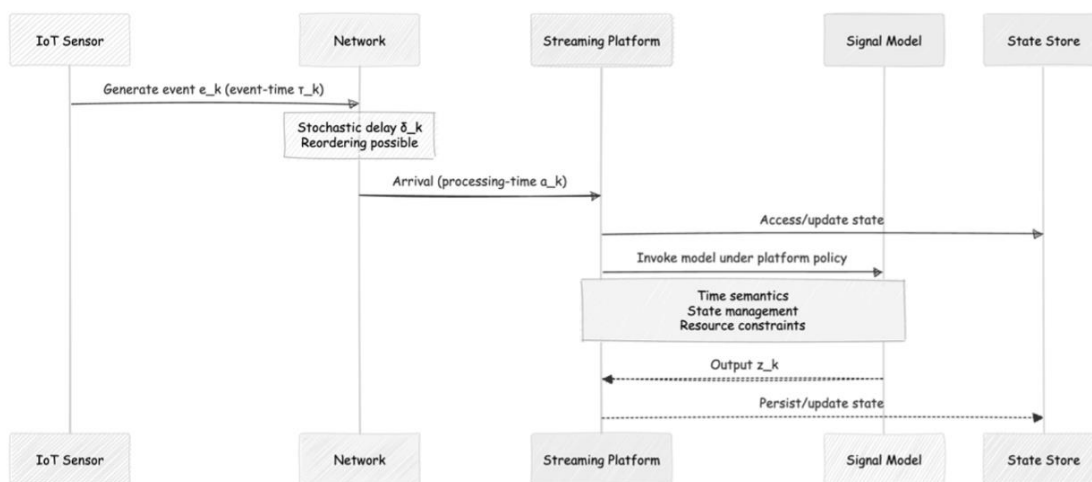


Рис. 1. Моделювання сигналів у реальному часі

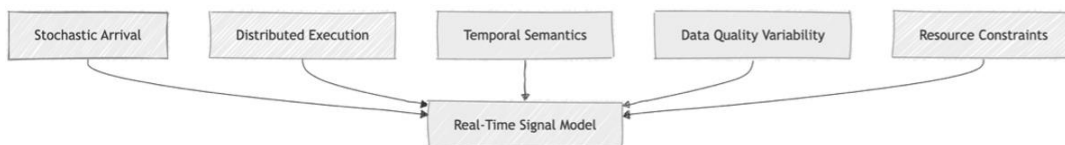


Рис. 2. Основні характеристики за змін яких, розроблена модель має залишатися коректною та стабільною

Опрацювання пізніх подій визначає політику реагування на події які прибувають після завершення певного часового вікна. Стратегія водяних знаків (watermark strategy), для визначення моменту після якого система вважає події достатньо повними, для їх агрегування [9]. Політики буферизації визначають правила тимчасового зберігання подій перед їх обробкою. Всі вище описані механізми безпосередньо впливають на еволюцію внутрішнього стану потокової системи та на те, як дані інтерпретуються та агрегуються. Таким чином, часова семантика є важливою частиною контексту моделювання. До того ж дані можуть бути пошкодженими, відсутніми взагалі в певних часових інтервалах та мати аномальні значення, тож доцільно розглядати при моделюванні і особливості якості вхідного потоку.

Системи інтернету речей часто функціонують у багаторівневих середовищах розгортання, що можуть включати периферійні (edge), туманні (fog) та хмарні (cloud) рівні. Кожен з зазначених рівнів відрізняються доступними обчислювальними ресурсами, обсягами пам'яті, пропускну здатністю мережі та різними допустимими затримками. Частина попередньої не складної обробки даних може бути виконана на периферійних вузлах, тоді як більш

ресурсоємні аналітичні процедури доцільно розгортати у централізованих середовищах, тобто на рівні хмари. Відповідно, складність і структура моделей сигнальної обробки повинні бути узгодженими з доступними ресурсами у межах конкретного рівня розгортання.

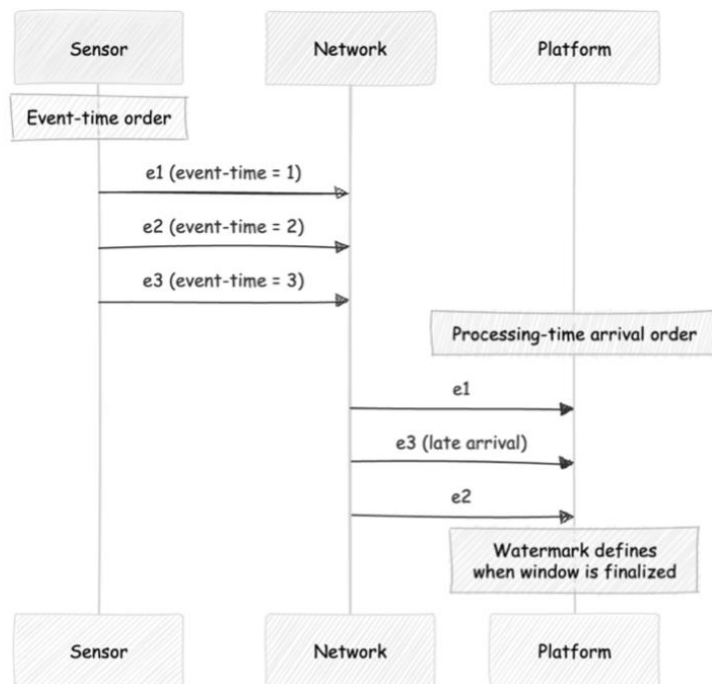


Рис. 3. Ілюстрація семантики часу подій та обробки на потоковій платформі, де події можуть надходити не в порядку виникнення, тоді як водяні знаки визначають, коли система починає агрегації

Підсумовуючи можемо виділити наступні вимоги до системно інтегрованої моделі в архітектуру платформи обробки:

- Модель повинна враховувати змінність інтенсивності потоку та нерегулярні часові інтервали, не порушуючи своїх основних властивостей.
- Поведінка моделі не повинна залежати від конкретного часу виконання або конфігурації вузла, натомість її основні властивості повинні залишатися збереженими в межах операційних змін розподіленої системи.
- Врахування часової семантики систем потокової обробки є важливою частиною контексту проектування та моделювання. Модель повинна бути узгодженою з часовою інтерпретацією.
- Характеристики якості даних можуть бути включені до моделювання задля підвищення інтерпретованості та надійності.
- Складність моделі має бути суміжною з обмеженням ресурсів та вимогами до операційної затримки.

У сукупності наведені характеристики та вимоги визначають операційні межі моделювання з обмеженнями архітектури. Модель сигналів у реальному часі можемо вважати як узгоджену з архітектурою, якщо її основні властивості зберігаються при змінах цих характеристик, за умови виконання окреслених системних обмежень.

Таким чином, на основі виділених системних характеристик, можемо також розглядати внутрішній стан \mathcal{S} потокового оператора \mathcal{P} саме як компоненти, які забезпечують виконання вимог до моделі. У такій постановці, внутрішній стан потокового оператора може бути визначений як множина змінних, що зберігають мінімально достатню інформацію про параметри моделі обробки сигналу, часовий контекст виконання та механізми консистентності, необхідні для забезпечення стабільної та коректної обробки подій у реальному часі.

Підсумовуючи, моделі програмного аналізу сигналів у контексті систем Інтернету речей розглядаються як інтегрований компонент потокової інфраструктури, а не як ізольована

алгоритмічна процедура. Запропонована концептуальна рамка дозволяє формалізувати вимоги до моделей, що функціонують у розподілених середовищах Інтернету речей і також створює основу для їх коректного проектування та ефективного впровадження в умови реального розгортання.

Висновки

В роботі представлено формальну модель потокової обробки сигнальних даних, що є тісно інтегрованою з контекстом її застосування; висвітлено архітектурні передумови для сталої та масштабованої інтеграції адаптивних моделей програмного аналізу сигналів у платформи даних у контексті Інтернету речей у реальному часі. Пропонується підхід, що визначає обробку сигналів у реальному часі у системах IoT, як системно інтегровану функцію, стабільність якої залежить не лише від алгоритмічних моделей а від організації процесів управління даними. Запропонований підхід є орієнтованим на інтеграції моделей програмного аналізу сигналів, в тому числі і гібридних моделей включаючи поєднання з алгоритмами машинного та глибокого навчання. Пропонується підхід орієнтований на розгортання аналітики у реальному часі, розділяючи рівні прийому, обробки, зберігання та обслуговування, орієнтований на усування розриву між теоретичним моделюванням сигналів та їх практичною імплементацію в реальні, розподілені, потокові системи.

Запропонований підхід передбачає поєднання різних принципів проектування систем узгоджуючи логічну структуру IoT-рівнів, моделей розгортання та організацію всього життєвого циклу даних у межах платформи даних задля досягнення ефективної та надійної обробки сенсорних даних у режимі реального часу. Варто зауважити, що запропонований підхід не передбачає використання конкретної моделі програмного аналізу сигналів, натомість він окреслює аспекти інфраструктури за яких адаптивні моделі можуть працювати ефективно та передбачувано. Сформовано концептуальну рамку, що дозволяє формалізувати вимоги, яким мають відповідати моделі які інтегруються у систему потокової обробки, та описані основні вимоги до таких моделей. Очікується, що за такого підходу система залишатиметься стійкою та відкритою до впровадження нових моделей програмного аналізу без зміни основної архітектури прийому або зберігання даних.

Подальші дослідження плануються спрямувати на поглиблення забезпечення теоретичної обґрунтованості та операційної стабільності запропонованого підходу в гетерогенних та недосконалих середовищах обробки даних у реальному часі. Передбачається подальша розробка та системна оцінка адаптивних методів моделювання сигналів у межах запропонованого підходу, а також подальший аналіз компромісів розгортання між затримкою, гарантіями узгодженості, обчислювальною складністю та масштабованістю моделей. Також плануються і подальші дослідження концепції внутрішнього стану системи. Очікується, що розвиток моделей у межах описаного архітектурного підходу забезпечить їх узгоджену інтеграцію в цілісну систему та операційну стабільність в реальних системах Інтернету речей. Окремий вектор подальших досліджень плануються зосередити на формалізації запропонованого підходу до рівня архітектурного фреймворку, що охоплює чітко визначені принципи проектування, управління якістю даних між рівнями та формалізовані інваріанти розгортання.

Список використаних джерел:

1. Real-Time Medical Data Analytics in Internet of Things-based Smart Healthcare Systems. American Journal of Medical Research. 2020. Vol. 7, no. 1. P. 61. URL: <https://doi.org/10.22381/ajmr7120209>.
2. Zhang L., Jeong D., Lee S. Data Quality Management in the Internet of Things. Sensors. 2021. Vol. 21, no. 17. P. 5834. URL: <https://doi.org/10.3390/s21175834>.
3. Choudhary A. Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things. 2024. Vol. 4, no. 1. URL: <https://doi.org/10.1007/s43926-024-00084-3>.

4. Vinny S., Brij M. G. Mitigating Security Threats in IoT Networks Using Big Data Analytics and On-Device Modeling. International Journal of Engineering and Management Research. 2025. Vol. 15, no. 1. P. 103–112. URL: <https://doi.org/10.5281/zenodo.15037481>.
5. Kumar N. Internet of Things-IoT: Definition, Characteristics, Architecture, Enabling Technologies, Application and Future Challenges. Independently Published, 2021.
6. Kamal R. Internet of Things: Architecture and Design Principles. Chennai, India: Mc Graw Hill India, 2017.
7. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics / M. Armbrust et al. 2021.
8. Ravindran G. S. Next-Generation Data Lakes: Innovations in Real-Time Analytics. Journal of Computer Science and Technology Studies. 2025. Vol. 7, no. 5. P. 803–809. URL: <https://doi.org/10.32996/jcsts.2025.7.5.90>.
9. Akidau T., others. Watermarks in stream processing systems: semantics and comparative analysis of Apache Flink and Google cloud dataflow. Proc. VLDB Endow. 2021. P. 3135–3147. URL: <https://doi.org/10.14778/3476311.3476389>.

Автори статті

Ткаченко Ольга – доктор технічних наук, професор, Київський національний університет імені Тараса Шевченка, Київ, Україна.

ORCID: 0000-0001-7983-9033

Шевченко Аліна – аспірант, Київський національний університет імені Тараса Шевченка, Київ, Україна.

ORCID: 0009-0003-9927-8706

Authors of the article

Tkachenko Olga – Doctor of Sciences (technical), Professor, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine.

ORCID: 0000-0001-7983-9033

Shevchenko Alina – postgraduate, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine.

ORCID: 0009-0003-9927-8706

Надійшла до редакції: 17.02.2026

Прийнята до друку: 13.03.2026

Опубліковано: 25.05.2026

© 2026 Ткаченко О.М., Шевченко А.Т.

Цей матеріал ліцензовано за умовами CC BY 4.0. <https://creativecommons.org/licenses/by/4.0>