

ЗАСТОСУВАННЯ БЕЗСЕРВЕРНИХ СИСТЕМ ДЛЯ ОБРОБКИ МЕТРОЛОГІЧНИХ МЕТАДАНИХ В ІОТ

*Оксана Гонсьор, к.т.н., доц., Олександр Демідов, аспірант,
Національний університет «Львівська політехніка», Україна;
e-mails: oksana.y.honsor@lpnu.ua, oleksandr.s.demidov@lpnu.ua*

<https://doi.org/10.23939/istcmtm2024.04.025>

Анотація

У статті досліджується потенціал безсерверних архітектур для ефективної обробки великомасштабних метаданих, що генеруються сенсорами Інтернету речей (IoT). Оскільки системи Інтернету речей стають дедалі складнішими, проблеми, пов'язані з обробкою великих обсягів даних у розподілених середовищах, стають все більш помітними. Ключовими питаннями є забезпечення точності даних, підтримка масштабованості та зменшення операційних витрат на інфраструктуру обробки даних. У статті запропоновано безсерверні обчислення як високоадаптивне рішення цих проблем, зосереджуючи увагу на їхній здатності до обробки в режимі реального часу, динамічного масштабування та безперешкодної інтеграції з сучасними хмарними платформами. Дослідження підкреслює важливість динамічного калібрування сенсорів Інтернету речей для забезпечення точності та надійності зібраних даних. Динамічне калібрування вирішує такі проблеми, як вплив змін навколишнього середовища і деградації сенсорів, використовуючи безсерверні системи для автоматизації повторного калібрування на основі аналізу даних в режимі реального часу. Автори пропонують архітектуру на основі Amazon Web Services (AWS) для демонстрації практичного застосування безсерверних принципів. Ця архітектура включає AWS Lambda для обчислювальних задач, SQS для розподілу робочого навантаження і S3 для масштабованого зберігання даних.

У статті підкреслюються переваги безсерверних систем, включаючи економічність, оптимізацію ресурсів і масштабованість, а також визнаються такі виклики, як безпечна інтеграція приватних даних і потенційні помилки в автоматизованих системах. Автори стверджують, що при правильному впровадженні безсерверні архітектури можуть забезпечити надійні рішення для обробки метаданих Інтернету речей, що дозволить підвищити продуктивність, надійність і економічну ефективність сучасних екосистем Інтернету речей.

Розглядаючи як теоретичні, так і практичні аспекти, дослідження пропонує цінну інформацію для дослідників і практиків, які прагнуть використовувати можливості безсерверних систем для додатків IoT. Результати дослідження підкреслюють трансформаційний потенціал хмарних безсерверних інфраструктур у досягненні ефективного і масштабованого управління даними для галузей, що використовують Інтернет речей.

Keywords

Інтернет речей, сенсорна мережа, обробка даних, безсерверні системи, цифрове калібрування.

1. Вступ

Обробка даних в системах Інтернету речей (IoT) пов'язана з низкою специфічних проблем, які виникають через розподілену структуру, велику кількість підключених пристроїв та обмеженість обчислювальних ресурсів. Інтернет речей генерує величезні обсяги метаданих від сенсорів, які містять інформацію про місцезнаходження, час, стан і виміряні параметри навколишнього середовища [1-3].

Динамічне калібрування сенсорів в системах Інтернету речей (IoT) є важливим аспектом забезпечення точності та надійності отриманих даних. Це особливо актуально для систем, що працюють в складних або змінних умовах, де точність вимірювань може коливатися з часом через фізичне зношування сенсорів, зміни в навколишньому середовищі, зміни температури або інших впливних факторів.

Безсерверні системи добре підходять для обробки таких даних завдяки своїй здатності до автоматичного масштабування, обробки подій в режимі реального часу і низьким витратам на інфраструктуру. Розгортання безсерверних систем для обробки метаданих сенсорів в контексті Інтернету речей (IoT) є перспективним напрямком досліджень, оскільки дозволяє розробити ефективну, гнучку і економічно ефективну інфраструктуру для обробки великих обсягів даних.

2. Недоліки

Для того, щоб дані, зібрані сенсорами Інтернету речей, були корисними в контексті автоматизованої системи, вони повинні пройти кілька етапів обробки. На етапі постобробки необхідна система, яка виконуватиме низку операцій, зокрема зберігання, перетворення в уніфікований формат, аналітику, генерацію результатів і передачу стороннім сервісам для подальшого використання (наприклад, моделям машинного навчання). Реалізація цих функцій може бути складною при роботі з великою кількістю вхідних даних, що вимагає

значних ресурсів. Основними вимогами до сенсорних даних в IoT є точність і надійність, які забезпечуються динамічним калібруванням сенсорів [4].

Динамічне калібрування сенсора - це процес автоматичного налаштування і корекції налаштувань сенсора в режимі реального часу або за певним графіком, що усуває необхідність ручного втручання. Такий підхід сприяє підвищенню точності вимірювань, навіть у контексті мінливих операційних обставин. Значна кількість систем динамічного калібрування включає механізми зворотного зв'язку, за допомогою яких система аналізує зібрані дані, виявляє будь-які невідповідності або відхилення, а потім автоматично коригує налаштування. Динамічне калібрування можна проводити за допомогою хмарних сервісів, які збирають дані з розрізнених сенсорів, аналізують їх і згодом видають директиви для калібрування. Хмара дозволяє зберігати великі обсяги даних і виконувати складні калібрувальні розрахунки, які може бути складно виконати на рівні окремого сенсора. Хмарні сервіси і безсерверні системи тісно пов'язані між собою, оскільки безсерверні системи являють собою один з видів послуг, які можна надавати через хмару. Їх об'єднує те, що обидві концепції використовують хмарну інфраструктуру для обробки даних, обчислень і управління додатками, але відрізняються з точки зору основних принципів роботи і підходів до управління ресурсами [5].

Існує багато моделей та інструментів для розробки такої системи, але оптимальним вибором є безсерверна архітектура. Цей підхід є найбільш придатним для обробки даних в системах IoT завдяки своїм відмінним характеристикам, зокрема: масштабованість, висока надійність, економічність, завадостійкість та можливість інтеграції з різноманітними сервісами, що підвищують продуктивність системи (логування, машинне навчання, сховища даних тощо) [6].

3. Мета

Мета статті - дослідити та проаналізувати потенційні переваги та виклики використання безсерверних архітектур для ефективної обробки метрологічних метаданих. Ця стаття має на меті надати уявлення про те, як безсерверні системи можуть підтримувати масштабованість, гнучкість і економічну ефективність, необхідні для обробки великомасштабних метрологічних даних, що є важливим для моніторингу в реальному часі, стандартизації даних та інтероперабельності в різних промислових і наукових застосуваннях. Розглядаючи конкретні випадки використання, найкращі практики та обмеження, у статті сформовані практичні рекомендації щодо впровадження безсерверних рішень, пристосованих до унікальних вимог обробки метрологічних даних в середовищі Інтернету речей (IoT).

4. Інтегрування з безсерверною системою обробки, модифікації та аналізу отриманих даних

4.1 Властивості без серверних систем в IoT

Безсерверні обчислення - це модель обчислень, при якій розробники можуть писати і виконувати код, не турбуючись про управління серверами, конфігурацію інфраструктури або масштабування ресурсів. Підхід безсерверних обчислень передбачає автоматичне розгортання і виконання функцій або додатків в інфраструктурі хмарного провайдера, при цьому ресурси автоматично масштабуються відповідно до необхідного навантаження.

Типові системи IoT складаються з трьох основних рівнів: збір, передача та аналітика даних, як показано на рис. 1 [7].

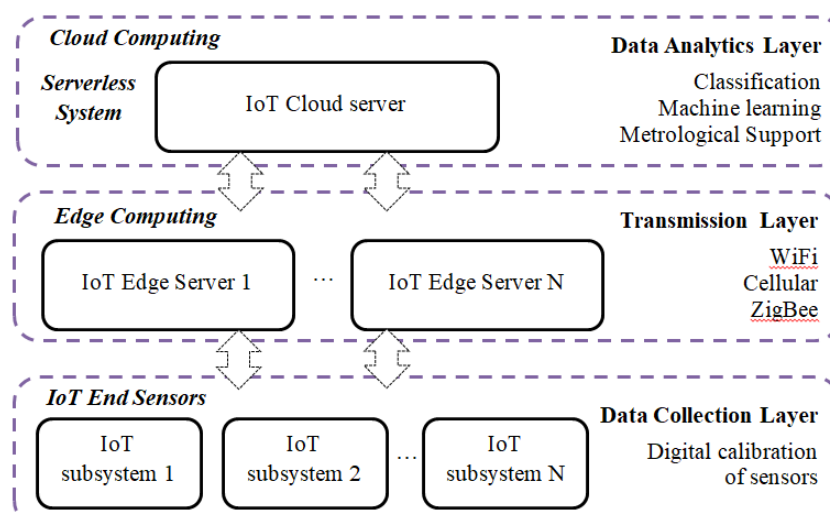


Fig. 1. An overview of IoT data analytics architecture

Рівень збору даних складається з кінцевих пристроїв Інтернету речей, які використовуються для виявлення, збору та зберігання даних з сенсорів. Пристрої Інтернету речей можуть утворювати підсистеми, такі як розумні будинки та інтелектуальні транспортні системи (ІТС). Кінцеві пристрої та підсистеми IoT є фундаментальними і основними компонентами систем IoT, які безпосередньо взаємодіють зі своїм фізичним середовищем IoT за допомогою сенсорів і виконавчих механізмів. Рівень передачі даних забезпечується шлюзами для передачі даних між кінцевими пристроями IoT та периферійними/хмарними серверами. Поширені стратегії передачі включають стільникові мережі, Wireless Fidelity (WiFi), Bluetooth, Zigbee тощо. Рівень аналітики відповідає за обробку та аналіз даних з пристроїв IoT, який може виконуватися як на хмарних, так і на периферійних серверах.

Безсерверні системи працюють на основі хмарних сервісів, і численні хмарні провайдери, включаючи Amazon Web Services (AWS), Microsoft Azure і Google Cloud Platform (GCP), пропонують послуги для розробки безсерверних додатків.

Безсерверні системи можуть бути використані для обробки метаданих Інтернету речей (IoT) в наступних ключових областях:

1. Аналіз даних у реальному часі. Безсерверні функції полегшують виявлення тенденцій або аномалій у даних в режимі реального часу. Це вигідно для систем моніторингу, які вимагають швидкої реакції на порогові значення або інші значущі події, такі як зміни температури або вологості.
2. Агрегація та фільтрація даних з сенсорів. Сенсори Інтернету речей (IoT) здатні генерувати високочастотні дані, які часто потребують фільтрації або зведення до більш узагальнених показників. Безсерверні функції можуть агрегувати ці дані і передавати в базу даних лише необхідну інформацію, наприклад, середні значення або відфільтровані події.
3. Динамічне калібрування сенсорів. Безсерверні функції можуть використовуватися для динамічного налаштування калібрування сенсорів на основі історичних даних або змін параметрів навколишнього середовища. Ці функції можна використовувати для запуску процедур перекалібрування, реєстрації змін та попереджень про необхідність технічного обслуговування.
4. Керування зберіганням метаданих та доступом до них. Безсерверні функції здатні класифікувати, фільтрувати та зберігати метадані у визначених структурах баз даних або сховищах даних. Це дозволяє ефективно організувати доступ до даних для подальшого аналізу або архівування.
5. Гарантія безпеки та відповідності вимогам. Вкрай важливо, щоб дані IoT оброблялися відповідно до встановлених стандартів безпеки та конфіденційності. Безсерверні функції можуть перевіряти відповідність потоків метаданих політикам безпеки, шифрувати конфіденційні дані та автоматично реагувати на підозрілі події. [8, 9].

4.2 Розроблення безсерверної архітектури системи обробки інформації від датчика IoT

Використання хмарних ресурсів AWS пропонує безліч функцій та сервісів, які полегшують виконання багатьох системних операцій. Запропонований набір сервісів і ресурсів, що використовуються для побудови безсерверної архітектури, а також пов'язані з ними з'єднання, передача даних і обґрунтування, окреслено на рис. 2.

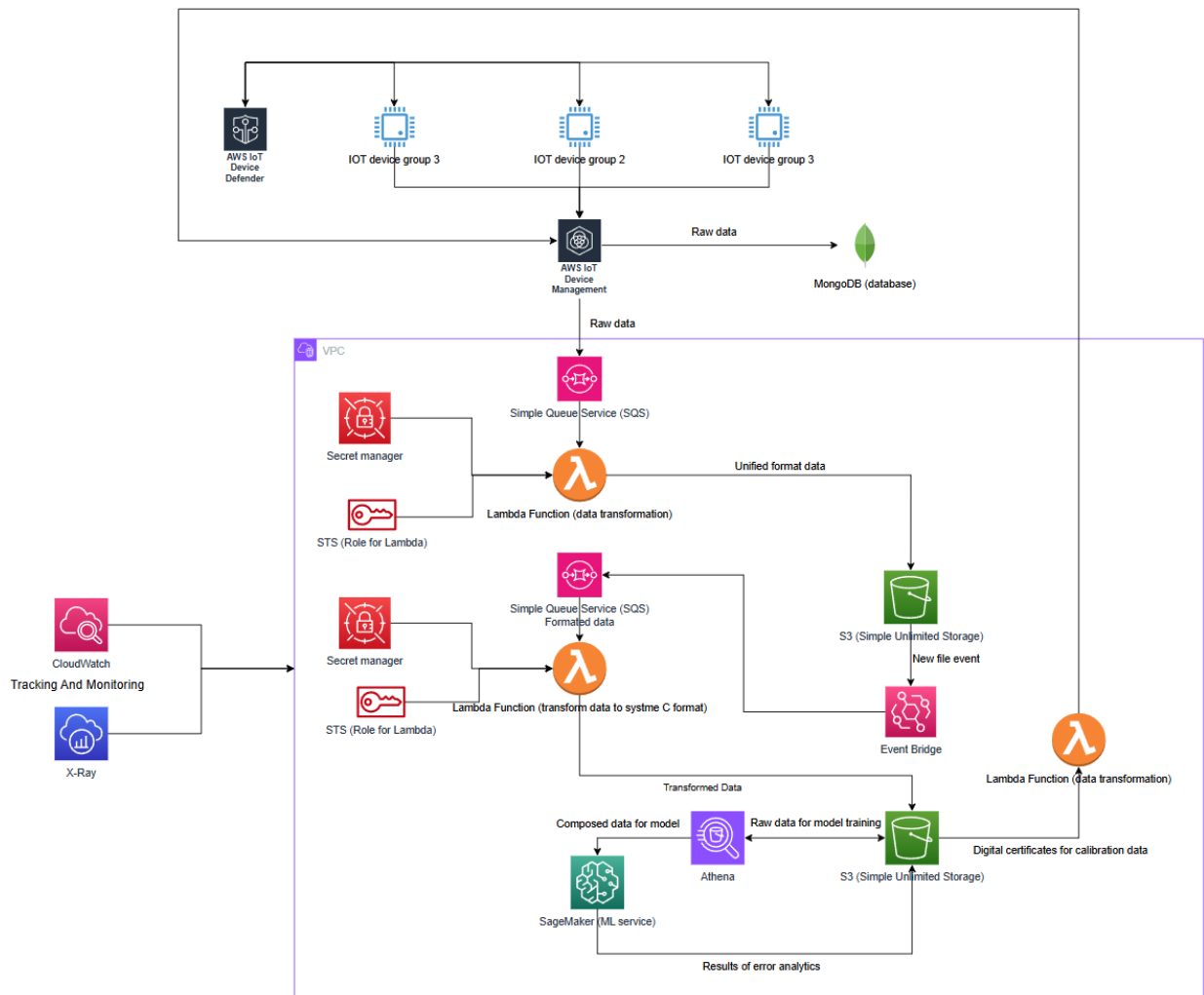


Fig. 2. Proposed architecture of a serverless system for processing information from IoT sensors

Нижче приведена характеристика кожної компоненти запропонованого архітектурного рішення:

1. У верхній частині системи відображаються три групи IoT-пристроїв. Однак можлива будь-яка кількість вибраних груп, оскільки ця архітектурна схема призначена для обходу обмежень, пов'язаних зі збільшенням кількості пристроїв та інформації, що підлягає аналізу.

2. Сервіс AWS IoT Device Defender підключений до вищезгаданих груп IoT-пристроїв, що дозволяє здійснювати моніторинг, реєстрацію та оцінку безпеки IoT-пристроїв. Це полегшує виявлення потенційних загроз і перевіряє операційну цілісність пристроїв. Крім того, сервіс AWS IoT Device Management, який підключений до всіх груп IoT, дозволяє дистанційно керувати групами IoT-пристроїв, паралельно виконувати певні команди та організовувати зв'язок із зовнішніми сервісами [10].

3. MongoDB - це нереляційна база даних, яка буде записувати всі чисті метадані перед обробкою. Це необхідно для збереження вихідних даних перед їх перетворенням, а також для можливої верифікації шляхом порівняння результату з початковим станом. Вибір нереляційної бази даних був зумовлений необхідністю записувати дані, які не мають єдиної схеми, структури чи формату.

4. Коли дані передаються з пристроїв Інтернету речей (IoT) до хмари Amazon Web Services (AWS), вони спочатку потрапляють до черги Simple Queue Service (SQS) у вигляді повідомлень. Це робиться для того, щоб розподілити навантаження і розтягнути його в часі. У разі одночасної передачі великого обсягу даних, повідомлення вилучаються з черги і обробляються, як тільки з'являється достатньо ресурсів. Крім того, існує спеціальна черга «мертвих листів» для обробки повідомлень, які не можуть бути оброблені негайно через певну помилку через короткий проміжок часу. Потім ці повідомлення передаються до лямбда-функції, яка відображає місцезнаходження наших обчислювальних потужностей. Лямбда-функція активна виключно під час обробки конкретного повідомлення і припиняє свою роботу, коли вона є не потрібною. Отже, початкова Лямбда-функція виконує початкове перетворення даних, перевірку та перетворення до одного типу.

5. Оброблені дані передаються до S3, хмарної системи зберігання даних, яка пропонує високий рівень ємності та інтегрована з низкою інших сервісів. Наприклад, як тільки дані потрапляють в S3, сервіс Event Bridge відстежує зміни в певній папці і повідомляє про це наступний етап, який, у свою чергу, викликає

функцію Lambda. Однак це робиться не для обробки даних, а для аналізу даних і створення з них наборів даних для моделі машинного навчання. На цьому етапі дані будуть передані в систему С. Варто зазначити, що кожна лямбда-функція має Secret Manager і STS (роль для лямбди). Це дозволяє передавати секретні ключі (наприклад, для написання бази даних або зв'язку зі сторонніми аналітичними ресурсами) і роль, яка визначає сервіси, до яких функція матиме доступ.

6. Після аналізу даних і перетворення їх у мережу передачі даних, вони повертаються до S3, де їх можна зібрати за допомогою Athena. Цей інструмент дозволяє виконувати SQL-запити до файлів S3 так, ніби вони є реляційною базою даних, за умови наявності схеми. Згодом мережеві дані передаються до моделі машинного навчання Sagemaker, яка слугує контейнером для обчислень моделі. Цей крок є необов'язковим, оскільки система може не використовувати машинне навчання, а використовувати аналіз даних за певним алгоритмом.

7. Після отримання результатів машинного навчання кінцеві дані записуються в S3. Після цього вони передаються до завершальної лямбда-функції, яка лише записує дані до різноманітних залежних систем. Це може спричинити створення документації або калібрування сенсорів, залежно від отриманої інформації або помилки.

4.3. Переваги та недоліки запропонованого архітектурного рішення

Запропонований архітектурне рішення має низку суттєвих переваг, які будуть розглянуті надалі. Загалом, недоліки є більш специфічними для конкретних обставин кожної окремої ситуації. Однак основним недоліком є інтеграція хмарних ресурсів. Це тягне за собою потенційну можливість порушень безпеки при надсиланні та перевірці вхідних даних для калібрування. Для того, щоб такий архітектурний підхід був практичним, важливо мати чітке розуміння типів даних, які будуть включені в аналітичну фазу, і забезпечити безпечну доставку цих даних, якщо вони є приватними. Цього можна досягти за допомогою ряду методів, включаючи використання приватної мережі, віртуальної приватної хмари без інтернет-шлюзу, що працює на виході, та інших [11-13]. Крім того, автоматизоване калібрування сенсорів на основі аналітичних даних за своєю суттю є невизначеним через потенційну можливість помилок як на етапі обробки, так і в аналітичній моделі або моделі машинного навчання. Для повної автоматизації цього процесу необхідно розробити систему виявлення перешкод та верифікації результатів (потенційно шляхом повторної обробки даних).

Тим не менш, переваги, які можна отримати від належного впровадження, є багатогранними. З точки зору безпеки, інформація, що зберігається в хмарі, і обробка цієї інформації будуть захищені багатьма показниками безпеки. До них відносяться надання приватних ключів, використання окремої віртуальної приватної хмари (VPC), система моніторингу та виявлення помилок, а також система ролей і контролю доступу. Крім того, є значний потенціал для розширення та підвищення ефективності. Крім того, система дозволяє виділяти необмежену кількість ресурсів для зберігання даних і обчислень.

І навпаки, ці ресурси працюють лише тоді, коли вводяться нові дані, забезпечуючи тим самим економічну ефективність системи. Функціонал оркестрування процесів дозволяє чітко розмежовувати етапи роботи, полегшуючи моніторинг та аналіз цих етапів, а також виявлення потенційних помилок або сфер для вдосконалення. Крім того, AWS пропонує безперешкодну інтеграцію з пристроями Інтернету речей, надаючи послуги, забезпечуючи безпеку і контроль над групами пристроїв Інтернету речей. До них відносяться AWS IoT Device Management і AWS IoT Device Defender.

4.4. Перспективи подальших досліджень

Враховуючи зазначені вище переваги та недоліки, а також наявність початкового архітектурного прототипу, є сенс продовжувати дослідження на основі практичного застосування. Хмарна інтеграція такої системи не є тривіальним завданням і включає багато змінних і факторів, які можуть змінити архітектурний підхід під час реалізації або на основі результатів практичних досліджень. Основними напрямками подальших досліджень будуть:

1. Вивчення можливих методів передачі даних між системою IoT і хмарою. Цей крок охоплює теоретичне дослідження найкращого рішення та його впровадження, а також отримання висновків за допомогою стрес-тестування та хакерського злому системи.
2. Пошук оптимального методу калібрування сенсора та алгоритму запобігання випадковим похибкам. Цей крок є ключовим, оскільки впливає на загальну надійність системи. Механізми перевірки даних перед калібруванням мають бути надійними. Цього можна досягти різними способами, включаючи впровадження моделі машинного навчання для перевірки отриманих даних і виявлення помилок.
3. Оцінка ефективності розробленої системи. Щоб оцінити ефективність запропонованого рішення, його необхідно порівняти з аналогічними конкурентами, іншими архітектурними підходами та альтернативними варіантами реалізації. Це передбачає визначення сильних і слабких сторін кожного рішення.

5. Висновки

Використання безсерверної архітектури для обробки метаданих сенсорів у системах Інтернету речей (IoT) дозволяє вирішити проблеми масштабованості, економічної ефективності та надійності. Запропоноване рішення, що базується на хмарних ресурсах AWS, забезпечує динамічне калібрування сенсорів, оперативну

аналітику в режимі реального часу, а також ефективну організацію зберігання та обробки великих обсягів даних. Незважаючи на певні проблеми, пов'язані з безпекою та інтеграцією приватних даних, ця архітектура демонструє високу гнучкість, ресурсоефективність та масштабованість, що робить її перспективним рішенням для сучасних IoT-рішень. Завдяки можливостям автоматизації, оркестрування та моніторингу ця архітектура може стати основою для надійної системи управління та обробки метрологічних метаданих в середовищах, що вимагають високої точності та надійності.

6. Подяка

The authors thank the Editorial Board of the Scientific journal "Measuring Equipment and Metrology" for their support.

7. Конфлікт інтересів

The authors state that there are no financial or other potential conflicts regarding this work.

Література

- [1] Eichstädt, S. (2020, June). From dynamic measurement uncertainty to the Internet of Things and Industry 4.0. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, pp. 632-635. IEEE. DOI:<https://doi.org/10.1109/MetroInd4.0IoT48571.2020.9138250>.
- [2] Schneider, T., Helwig, N., & Schütze, A. (2018). Industrial condition monitoring with smart sensors using automated feature extraction and selection. *Measurement Science and Technology*, 29(9), 094002. DOI:<https://doi.org/10.1088/1361-6501/aad1d4>
- [3] S. Hackel, F. Härtig, T. Schrader, A. Scheibner, J. Loewe, L. Doering, B. Gloger, J. Jagieniak, D. Hutzschenreuter, and G. Söylev-Öktem. The fundamental architecture of the DCC. *Measurement: Sensors*, 18:100354, 2021. DOI:<https://doi.org/10.1016/j.measen.2021.100354>.
- [4] Gu, J., Liu, C., Zhuang, Y., Du, X., Zhuang, F., Ying, H., ... & Guizani, M. (2020). Dynamic measurement and data calibration for aerial mobile IoT. *IEEE Internet of Things Journal*, 7(6), pp. 5210-5219. DOI:<https://doi.org/10.1109/JIOT.2020.2977910>
- [5] Nummiliukki, J., Saxholm, S., Kärkkäinen, A., & Koskinen, S. (2023). Digital Calibration Certificate in an industrial application. *Acta IMEKO*, 12(1), pp. 1-6. DOI: 10.21014/actaimeko.v12i1.1402.
- [6] Rajan, R. A. P. (2018, December). Serverless architecture-a revolution in cloud computing. In *2018 Tenth International Conference on Advanced Computing (ICoAC)*, pp. 88-93. IEEE. 23 December 2019. DOI:<https://doi.org/10.1109/ICoAC44903.2018.8939081>
- [7] Yang, Li & Shami, Abdallah. (2022). IoT Data Analytics in Dynamic Environments: From An Automated Machine Learning Perspective. 10.48550/arXiv.2209.08018. 105 p. DOI:10.48550/arXiv.2209.08018
- [8] Mukherjee, S. (2019). Benefits of AWS in modern cloud. *arXiv preprint arXiv:1903.03219*. DOI:<https://doi.org/10.48550/arXiv.1903.03219>
- [9] Wittig, A., & Wittig, M. (2023). *Amazon Web Services in Action: An in-depth guide to AWS*. Simon and Schuster. ISBN: 163343916X
- [10] Aquino, G., Queiroz, R., Merrett, G., & Al-Hashimi, B. (2019). The circuit breaker pattern targeted to future iot applications. In *Service-Oriented Computing: 17th International Conference, ICSOC 2019, Toulouse, France, October 28–31, 2019, Proceedings 17*, pp. 390-396. Springer International Publishing. DOI:https://doi.org/10.1007/978-3-030-33702-5_30
- [11] H.M. Khan, A. Khan, F. Jabeen, A. Anjum, G. Jeon (2021). Fog-enabled secure multiparty computation-based aggregation scheme in smart grid. *Computers & Electrical Engineering*, Vol. 94, 107358. DOI:<https://doi.org/10.1016/j.compeleceng.2021.107358>.
- [12] Song, B., Yu, Y., Zhou, Y., Wang, Z., & Du, S. (2018). Host load prediction with long short-term memory in cloud computing. *The Journal of Supercomputing*, 74, pp. 6554-6568. DOI: <https://doi.org/10.1007/s11227-017-2044-4>
- [13] Le, D. M., Dang, D. H., & Nguyen, V. H. (2018). On domain driven design using annotation-based domain specific language. *Computer Languages, Systems & Structures*, 54, pp. 199-235. DOI:<https://doi.org/10.1016/j.cl.2018.05.001>