

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЄДНАК Іван Павлович

**Модуль edge-обчислень для цифрового розвитку
інноваційного виробництва / Edge computing module for digital
development of innovative production**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
І. П. Єднак

Науковий керівник:
к.т.н., ст. викладач
М.З. Домбровський

Кваліфікаційну роботу допущено до
захисту

«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЄДНАК Івана Павловича
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль edge-обчислень для цифрового розвитку інноваційного виробництва / Edge computing module for digital development of innovative production

керівник роботи М. З. Домбровський, к.т.н., старший викладач кафедри ІОСУ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджено наказом по університету від “ ____ ” _____ 20__ року № ____

2. Строк подання студентом закінченої кваліфікаційної роботи _____

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- провести аналіз існуючих рішень edge-обчислень, вимог до систем реального часу у виробництві;
- оцінити сучасні підходи до вирішення задач проєктування модуля edge-обчислень, визначити основні компоненти;
- розробити концепцію архітектури модуля, запропонувати алгоритмічні та інформаційні рішення, що включають механізми збору, попередньої обробки та аналізу даних діагностики виробництва;
- розробити програмно-технологічне забезпечення і компоненти модуля;
- протестувати розробку модуля з активною перевіркою функцій на основі модель імітації поведінки.

5. Перелік графічного матеріалу в роботі: -

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01.2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2025 р.	

Студент _____ Єднак І. П.
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Домбровський М. З
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль edge-обчислень для цифрового розвитку інноваційного виробництва» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 51 сторінки і містить 14 ілюстрацій, 1 таблицю, 1 додаток та 36 використаних джерел.

Метою роботи є дослідити та розробити модуль edge-обчислень у контексті розвитку цифрового інноваційного виробництва для зменшення затримок та підвищення продуктивності обробки даних.

Методами розроблення обрано: аналіз наукової літератури та існуючих рішень щодо предметної області, методи моделювання і алгоритмічний підхід до вибору діагностичної моделі передбачливої експлуатації обладнання, аналіз і класифікація можливих режимів відмов, їх наслідків та критичності згідно стандарту ІЕС 60812, розробка спеціалізованих архітектурних рішень на основі мікроконтролера STM32L496 та процесора ARM Cortex-M4, підключення через протокол MQTT, віддаленого сервера відправки даних data.sparkfun.com.

Результатом виконання кваліфікаційної роботи є розробка модуля edge-обчислень, як складової розподіленої комп'ютерної системи, що забезпечує фільтрацію та попередню обробку, виявлення аномалій, скорочення затримок у передачі та обробці даних; зниження навантаження на хмарні сервіси; стабільної та автономної роботи системи в умовах змінного мережевого середовища.

Результати розробки модуля можуть бути використані у галузях smart виробництв (Industry 4.0) для моніторингу стану обладнання, виявлення дефектів у режимі реального часу, у енергетиці для оптимізації локального енергоспоживання, управління навантаженням, у транспортній інфраструктурі для керування сигналами і контролю дорожньої ситуації.

Ключові слова: . EDGE-ОБЧИСЛЕННЯ, ЦИФРОВИЙ РОЗВИТОК, ІННОВАЦІЙНЕ ВИРОБНИЦТВО, ЗБІР ДАНИХ, ПОПЕРЕДНЯ ОБРОБКА ДАНИХ, ЗМЕНШЕННЯ ЗАТРИМКИ.

ANNOTATION

Qualification work on the topic "Edge computing module for digital development of innovative production " for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 51 pages and it contains 14 figures, 1 table, 1 annexe and 36 sources.

The purpose of the work is to research and develop the edge computing module in the context of the development of digital innovative production to reduce delays and increase data processing performance.

The following development methods were selected: analysis of scientific literature and existing solutions in the subject area, modeling methods and an algorithmic approach to choosing a diagnostic model for predictive equipment operation, analysis and classification of possible failure modes, their consequences and criticality according to the IEC 60812 standard, development of specialized architectural solutions based on the STM32L496 microcontroller and the ARM Cortex-M4 processor, connection via the MQTT protocol to the remote data sending server data.sparkfun.com.

The result of the qualification work is the edge computing module development as component of a distributed computer system that provides filtering and pre-processing, anomaly detection, reduction of delays in data transmission and processing; reduction of the load on cloud services; stable and autonomous operation of the system in a changing network environment.

The results of the module development can be used in the smart manufacturing (Industry 4.0) for monitoring the equipment state, detecting defects in real time, in energy for optimizing local energy consumption, load management, in transport infrastructure for signal control and traffic situation monitoring.

Keywords: EDGE COMPUTING, DIGITAL DEVELOPMENT, INNOVATIVE MANUFACTURING, DATA ACQUISITION, DATA PRE-PROCESSING, LATENCY REDUCTION.

ЗМІСТ

Вступ.....	7
1 Аналіз принципів розвитку інноваційного виробництва та підходів підвищення ефективності бізнес-процесів.....	9
1.1 Стан та аналіз розвитку інноваційного виробництва	9
1.2 Огляд і аналіз існуючих рішень.....	11
1.3 Постановка задачі та шляхи її вирішення.....	16
2. Алгоритмічне та інформаційне забезпечення модуля.....	19
2.1 Загальна структура проєктованого модуля edge-обчислень.....	19
2.2 Обґрунтування алгоритмічного забезпечення діагностики експлуатації та технічного обслуговування виробничого обладнання	21
2.3 Інформаційне забезпечення діагностики експлуатації та технічного обслуговування виробничого обладнання.....	24
3. Програмно-технологічне забезпечення модуля та перевірка його функціонування	31
3.1 Вибір технологій та засобів для реалізації модуля.....	31
3.2 Реалізація ключових компонентів модуля для окраїнних обчислень.....	35
3.1 Опис проведення тестування програмного забезпечення.....	41
Висновки	44
Список використаних джерел	46
Додаток А. Копія опублікованих результатів	50

ВСТУП

Актуальність кваліфікаційної роботи полягає в тому, що застосування цифрових обчислювальних технологій визначає конкурентоспроможність сучасного виробництва, впровадження edge-обчислень уможливорює здійснювати обробку даних безпосередньо біля джерел: сенсорів, контролерів або обчислювальних пристроїв, що зменшує затримки, забезпечує роботу системи при нестабільному підключенні до хмари і оптимізує обчислювальні ресурси. Тема є актуальною, оскільки поєднує в собі дослідження архітектури обчислень, розробку алгоритмів, забезпечення надійності розподілених комп'ютерних систем в контексті стрімкого зростання кількості даних і підвищення вимоги до їх обробки у режимі реального часу. У архітектурі, яка орієнтована переважно на традиційні сервіси хмарних обчислень, не завжди забезпечуються необхідна швидкодія, надійність і стійкість, особливо для критично важливих застосувань - таких як виявлення аномалій у виробничих процесах або оперативне реагування в інтелектуальних системах керування. За таких умов постають задачі перенесення частини обчислень ближче до джерел даних, що передбачає створення модуля edge-обчислень для: скорочення затримок у передачі та обробці даних; зниження навантаження на хмарні сервіси; забезпечення стабільної та автономної роботи системи в умовах змінного мережевого середовища. Функціональність модуля як складової розподіленої комп'ютерної системи, полягає у виконанні: локального збору даних з датчиків IoT, фільтрації та попередньої обробки, виявлення аномалій, передавання лише оптимальної та релевантної інформації до хмари. Таким чином, забезпечується скорочення обсягів переданих даних, зменшення затримок та зростання автономності, що відповідає викликам цифрової трансформації.

Метою дипломної роботи є дослідити та розробити модуль edge-обчислень у контексті розвитку цифрового інноваційного виробництва для зменшення затримок та підвищення продуктивності обробки даних.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Провести аналіз існуючих рішень edge-обчислень, вимог до систем реального часу у виробництві.

2. Оцінити сучасні підходи до вирішення задач проєктування модуля edge-обчислень, визначити основні компоненти.
3. Розробити концепцію архітектури модуля, запропонувати алгоритмічні та інформаційні рішення, що включають механізми збору, попередньої обробки та аналізу даних діагностики виробництва.
4. Розробити програмно-технологічне забезпечення і компоненти модуля.
5. Протестувати розробку модуля з активною перевіркою функцій на основі модель імітації поведінки.

Об'єкт дослідження – процеси збору, представлення, обробки, зберігання, передачі та доступу до інформації в комп'ютерних системах.

Предмет дослідження – методи і технології обробки інформації в рамках концепції edge-обчислень для інтелектуального аналізу даних і прийняття рішень цифрового інноваційного виробництва.

Для виконання кваліфікаційної роботи використано методи аналізу наукової літератури та існуючих рішень щодо предметної області, методи моделювання і алгоритмічний підхід до вибору діагностичної моделі передбачливої експлуатації обладнання, аналіз і класифікація можливих режимів відмов, їх наслідків та критичності згідно стандарту IEC 60812, розробка спеціалізованих архітектурних рішень на основі мікроконтролера STM32L496 та процесора ARM Cortex-M4, підключення через протокол MQTT, віддаленого сервера відправки даних data.sparkfun.com.

Результати розробки модуль можуть бути використані у галузях смарт виробництв (Industry 4.0) для моніторингу стану обладнання, виявлення дефектів у режимі реального часу, у енергетиці для оптимізації локального енергоспоживання, управління навантаженням, у транспортній інфраструктурі для керування сигналами і контролю дорожньої ситуації, у сільському господарстві для управління мікрокліматом теплиць.

Публікація та апробація кваліфікаційної роботи здійснено у матеріалах студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІТАР-2025), (Тернопіль, 27–29 травня 2025 р.). (Додаток А)

1 АНАЛІЗ ПРИНЦИПІВ РОЗВИТКУ ІННОВАЦІЙНОГО ВИРОБНИЦТВА ТА ПІДХОДІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ БІЗНЕС-ПРОЦЕСІВ

1.1 Стан та аналіз розвитку інноваційного виробництва

В еру 4-ї промислової революції і зростаючої комп'ютеризації цифрові технології є ключовими чинниками розвитку інноваційного виробництва та підвищення ефективності бізнес-процесів і сприяння підвищенню конкурентної переваги.

Перехід світової економіки від індустріальної до інноваційної форми породжує необхідність вивчення нових способів управління інноваційною діяльністю організацій. Особливо актуальними є питання трансформації результатів науково-дослідної та промислово-інноваційної діяльності в цифрове середовище. З точки зору підвищення конкурентоспроможності промислової продукції на світових ринках ключовим є інноваційний розвиток промислового виробництва [1, 2].

Стійка конкурентоспроможність промислових підприємств визначається їхньою здатністю постійно проводити інновації методів, продуктів та послуг. детермінанти інноваційного розвитку в рамках цифрової трансформації економіки. При цьому виникає необхідність трансформації способів управління та вивчення функціонування промислового виробництва та визначити стратегічні пріоритети інноваційного розвитку промислового виробництва на основі механізмів управління, що забезпечують стабільне економічне зростання та підвищення конкурентоспроможності промислових підприємств у цифровій економіці [1, 3].

Зростання автоматизації протягом 1980-х років створило низку обмежень у нашому повсякденному житті. Такі системи були розроблені інженерами у формі жорстких комп'ютерних алгоритмів, які, на жаль, не враховували всі контексти використання, що неминуче виникають в операціях, того чи іншого дня. В результаті ми стикаємося із закритою та жорсткою системою, яка не допускає жодної адаптації. Цю жорстку автоматизацію сьогодні необхідно замінити більш гнучкою підтримкою, тобто адаптованою до контексту [3-5].

У контексті глобалізованої економіки, де адаптивність і технологічна складність є критично важливими, цифрова трансформація стає вирішальним фактором у формуванні майбутньої траєкторії розвитку бізнес-середовища. зменшення споживання ресурсів і пом'якшення впливу на навколишнє середовище, узгоджуючи свою практику з дедалі більшим акцентом на корпоративній соціальній відповідальності та принципах циркулярної економіки. Інтеграція цифрових технологій є ключовим рушієм для продуктів, процесів, маркетингу та організаційних інновацій, які разом сприяють конкурентним перевагам і підвищенню ефективності бізнес-процесів [1, 4, 5].

Технології цифровізації надають компаніям можливість швидко адаптуватися до ринкових коливань, оптимізувати продуктивність і підвищити стійкість у все більш динамічному економічному середовищі, де адаптивність і технологічна складність є критично важливими. Діджиталізація має вирішальне значення для оптимізації виробничих процесів, покращення зв'язку між діловими партнерами та зниження операційних витрат у поточній глобальній економіці. Це дозволяє компаніям використовувати дані в реальному часі для прийняття обґрунтованих рішень, автоматизації повторюваних процесів і підвищення ефективності [1, 3, 6].

Стратегічне впровадження цифровізації виходить за рамки вдосконалення операційний, уможливаючи реконфігурацію традиційних бізнес-моделей.

Цифрова трансформація має виходити за межі цифровізації. Це вимагає від організації краще реагувати на зміни та бути більш гнучким. Основні напрямки цифрової трансформації охоплюють автоматизацію, аналітику даних, Інтернет речей (IoT), штучний інтелект (AI), хмарні обчислення та цифрові платформи. Розвиток (IoT) та Industry 4.0 надав можливість для більш складного та комплексного підходу до проектування, виробництва та оптимізації складання та виробництва компонентів [3, 4, 7]. Зокрема, перехід від традиційної інженерії до цифрової інженерії викликає перехід від жорсткої автоматизації до гнучкої автономії [2], що відображено на рисунку 1.1.

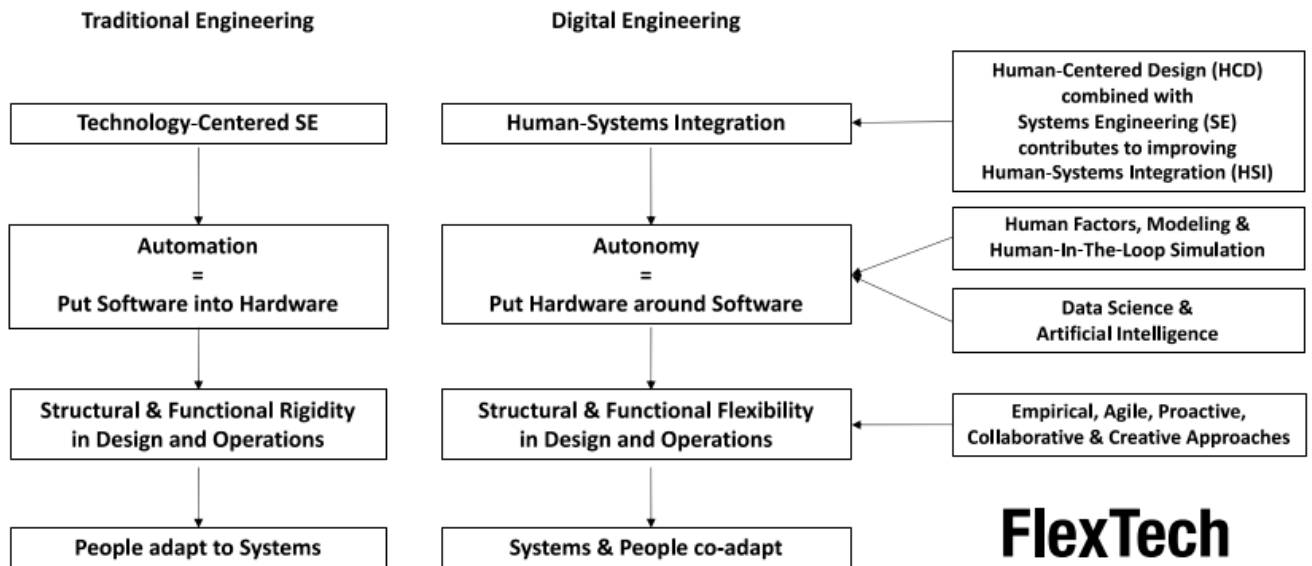


Рисунок 1.1 – Перехід до цифрової інженерії: від жорсткої автоматизації до гнучкої автономії, адаптовано за [2]

Наскрізні моделі цифрової трансформації, дають змогу використовувати автоматизоване проектування та виробництво, сенсорні системи, моделі на основі даних і фізики, прогнозування та діагностику несправностей, а також візуалізацію даних для створення віртуального інструментарію управління активами [1, 3, 7].

Використовуючи цифрові інструменти, підприємства можуть надавати клієнтам персоналізований досвід, покращувати процес прийняття рішень завдяки аналітиці даних у режимі реального часу та досягати ефективнішої координації між ланцюжками створення вартості. Таким чином, ми погоджуємося з авторами роботи [4], у якій стверджується, що якщо промислова революція відбудеться, вона матеріалізується як поєднання соціальних і технологічних інновацій у соціально-цифровій трансформації.

1.2 Огляд і аналіз існуючих рішень

Завдяки новим ІТ, інноваційне виробництво прагне вищого ступеня інтелекту, завдяки можливостям недорогого та повсюдного зондування, передових обчислень та аналітики даних, а також кіберфізичної інтеграції. Конвергенція різних передових та підключених пристроїв і машин є неминучим елементом

інноваційного виробництва. Більше того, зростаюча повсюдність Інтернету речей [2, 6, 8] надає перспективні можливості для створення потужних промислових систем та додатків [3]. В результаті, великий обсяг різноманітних виробничих даних генерується цими підключеними машинами та датчиками. Ці дані необхідно очищувати, зберігати та обробляти для створення інформації та знань, які мають стати основою для інноваційного виробництва [4, 7, 9].

Однак експоненціальне зростання даних виходить за межі загальних можливостей їх обробки користувачами. Значну потребу оброблення зростання даних у багатьох випадках забезпечують хмарні обчислення [8, 10, 11]. Хмарні обчислення - це більше, ніж просто технологічна інновація; це трансформаційний фактор для всього глобального бізнес-середовища [6, 9, 12].

Хмарні обчислення - це інтернет-обчислення, де спільні ресурси (наприклад, сховища та обчислювальні засоби, програмне забезпечення, дані, програми тощо) доступні та використовуються на вимогу зручним способом «плати за використання» [6]. Запровадження хмарних обчислень надає компаніям значні можливості для оптимізації діяльності та покращення інноваційного потенціалу [9, 11, 13]. У парадигмі хмарних обчислень користувачі отримують високоякісні послуги за нижчою ціною. Завдяки потенційним та практичним перевагам для суспільства та економіки, парадигма хмарних обчислень привернула величезну увагу з боку наукових кіл та промисловості. Наприклад, багато компаній, таких як Amazon, Google, Salesforce, IBM, Microsoft тощо, розробили свої платформи хмарних обчислень для надання послуг повсюдним користувачам [12, 14, 15].

При цьому, у поєднанні з виробництвом, хмарні обчислення призвели до появи нової хмарної моделі виробництва. Усі виробничі ресурси та можливості (MR&C) віртуалізовані та інкапсульовані як послуги, які керуються, розподіляються та використовуються на вимогу через хмару [8, 15, 16].

Однак дані, зібрані з машин та датчиків, надсилаються до хмари, а бажаний вихід знову надсилається на потрібні пристрої, що може призвести до затримки відповіді. Крім того, масивні дані вимагають високої пропускну здатності, що є дорогим [13]. Через недоступність мережі ці проблеми стають перешкодами для програм, що потребують обмежень затримки [14, 17, 18].

Глибока інтеграція всіх аспектів виробничого процесу з новими ІТ сприяє значним змінам у виробничій моделі, які можна назвати розумним виробництвом. Завдяки новим ІТ все більше виробничих пристроїв підключаються до Інтернету, і все більше користувачів беруть участь у виробничому процесі. В результаті виробнича система стає все більш складною та динамічною, що вимагає ресурсів та співпраці користувачів. Крім того, дані, що генеруються у виробництві, також стають все багатшими, різноманітнішими та складнішими [6, 19, 20].

Як один з основних факторів, що сприяють виробництву, хмарні обчислення трансформували традиційні моделі виробництва на обчислювальну та сервісно-орієнтовану модель виробництва, яка стає більш сумісною, розумною, адаптивною та розподіленою [12, 21]. З численних хмарних виробничих застосувань [14, 22], архітектуру хмарної виробничої системи можна узагальнити [9], як показано на рисунку 1.2. Архітектура складається з двох основних рівнів: фізичної виробничої системи та хмарного середовища. Нижній рівень представляє MR&C, включаючи виробничі пристрої, а також користувачів, які беруть участь у виробничій діяльності.

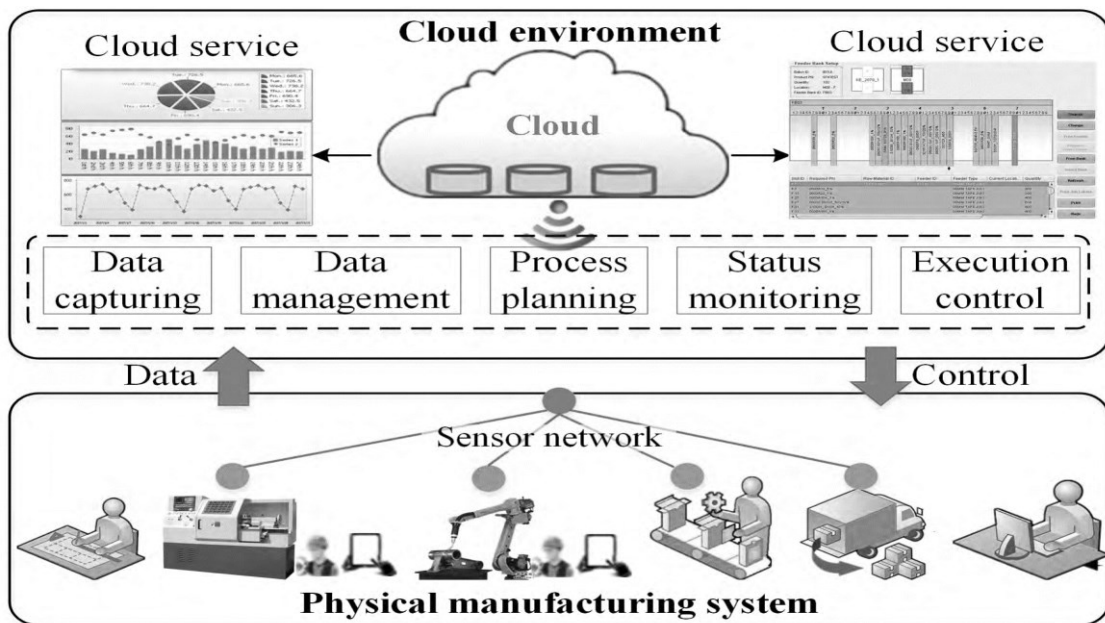


Рисунок 1.2 – Архітектура хмарної виробничої системи

Таким чином, є потреба у розвитку інноваційного виробництва, засновано на туманних обчисленнях та хмарних обчисленнях, щоб надати покращення різних

рівнів інтелекту цеху. Зосередимося на трьох ключових підтримуючих технологіях та напрямках досліджень для впровадження IoT-додатків.

Щоб перенести обчислювальні, сховищні та мережеві ресурси ближче до краю обчислювального середовища та забезпечити роботу майбутніх IoT-додатків, необхідна маломасштабна хмарна обчислювальна платформа. NFV (віртуалізація мережевих функцій) та SDN (програмно-визначувані мережі) виглядають як дві ключові синергетичні технології, що дозволяють реалізувати таке бачення [9, 23].

Завдяки NFV на периферії доступні локальні обчислювальні, сховищні та мережеві ресурси, що розташовані ближче до пристроїв, для додатків (наприклад, відеоспостереження, розпізнавання облич, доповнена реальність), які генерують великий обсяг даних або вимагають низької затримки. В edge-хмарах NFV може бути реалізовано на основі доступних промислових стандартних серверів, комутаторів та сховищ, створюючи віртуальні мережеві функції (VNF), що замінюють традиційне та спеціалізоване обладнання від закритих постачальників. Ці VNF можуть запускатися або зупинятися динамічно відповідно до попиту, а також розміщуватися у більш гнучких місцях. Їх можна об'єднувати у ланцюги та масштабувати вгору або вниз для складних функцій і додатків. Усі зазначені переваги NFV можуть бути використані додатками в edge-хмарі за умови впровадження цієї технології [6, 17, 23].

SDN, з іншого боку, дуже добре поєднується з NFV в edge-області для організації мережі, конфігурації, контролю та управління створеними NFV VNF. SDN може значно знизити витрати та підвищити гнучкість і програмованість VNF в edge-області завдяки відокремленню контролю від передачі даних та використанню централізованого управління і конфігурації мережі. Просту ілюстрацію взаємозв'язку між NFV, SDN та відкритими інноваціями [12] показано на рисунку 1.3.

З технічної точки зору, NFV і SDN є високо доповнюючими для перспектив edge-області. Розділення контролю і передачі даних у SDN може спростити сумісність NFV з існуючими розгортаннями.

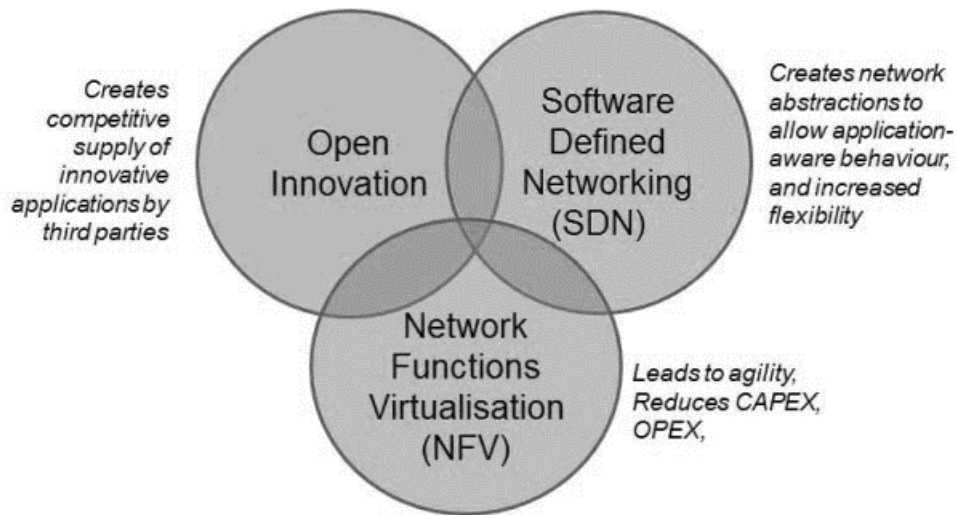


Рисунок 1.3 – Конвергенція NFV та SDN у відкритому інноваційному середовищі

NFV може підтримувати SDN, надаючи інфраструктуру, на якій працює SDN. Конвергенція NFV і SDN в edge-області потенційно відкриває нові можливості для інноваційного, швидкого і економічного впровадження нових послуг та додатків.

З нетехнічної точки зору, зацікавленими сторонами майбутніх edge-областей та ринку додатків будуть інтернет-провайдери, постачальники прикладних сервісів (ASP), виробники пристроїв та постачальники програмного забезпечення, які у проєктуванні рішень враховуватимуть параметри сучасних обчислювальних підходів (Таблиця 1.1).

Таблиця 1.1 – Порівняння параметрів переваг та недоліків хмарних, туманних та крайових обчислень

Параметри	Хмарні обчислення (Cloud Computing)	Туманні обчислення (Fog Computing)	Окрайні обчислення (Edge Computing)
Переваги	Масштабованість, обробка великих даних, необмежені обчислювальні ресурси.	Безпека, що налаштовується користувачем, низька затримка.	Відгук у реальному часі, дуже низька затримка, може працювати без хмари або fog.
Недоліки	Висока затримка, повільний час реакції, відсутність офлайн-режиму, нестача безпеки.	Обмежене зберігання, Fog потребує більше з'єднань із хмарою, ніж Edge Computing, для переносу даних з фізичного рівня в цифровий.	Обмежене зберігання, з'єднання через пропріетарні мережі, високе енергоспоживання.

Як нова парадигма обчислень, edge-обчислення пропонують можливість підвищити ефективність та розробляти інноваційні послуги [5, 23-25], що прискорить розвиток виробництва згідно концепції Індустрії 4.0.

1.3 Постановка задачі та шляхи її вирішення

На підставі викладеного, вимоги до розробки застосунків edge-обчислень можуть бути сформульовані різними способами:

- вирішити дану проблему за менший час;
- вирішити більшу проблему за той самий час;
- отримати достатньо пам'яті для вирішення заданої (або більшої) проблеми;
- вирішувати все більші проблеми незалежно від часу виконання.

Питання полягає у виборі метрики того, який виграш від використання окраїнних процесорів ми очікуємо та які фактори є визначальні.

Дослідження властивостей паралельної системи починається з визначення прискорення обчислювальної швидкодії системи шляхом додавання ресурсів або процесорів. Якщо час (n, x) позначає час, необхідний n ресурсам або процесорам системи для виконання програми розміром x , тоді «прискорення проблеми розміру x з n процесорами є часом виконання на одному процесорі, поділеному на час на n процесорах [25, 26].

Одним із важливих показників є ефективність, яка характеризує збільшення первинного робочого навантаження, для підтримки постійної ефективності, оскільки до неї додається більше ресурсів. Насправді, як зазначав Г. Амдал [9]: «ідеального паралелізму загалом неможливо досягти, оскільки існують певні частини робочого навантаження, які можна виконувати лише послідовно».

Паралельна архітектура та паралельна обробка для зберігання історії операцій, як приклад системи дистанційного моніторингу та обслуговування верстатів з ЧПК, що обслуговує верстатобудівну компанію. Наприклад, для «безшовної інтеграції» «загальнозаводської архітектури автоматизації та

інформації» з можливістю до 120 000 вводів/виводів, використовують SIMATIC IT Manufacturing Execution [15, 16, 27].

Щоб забезпечити керування завданнями на обчислювальному краю в таких програмах, як інтелектуальне виробництво, управління енергією, безпека, захист конфіденційності необхідні edge-обчислень. Наприклад, в інтелектуальному виробництві хмара контролює весь процес, а крайові вузли забезпечують виявлення в реальному часі та можливості вирішення проблем. Завдяки використанню можливостей edge-обчислень у режимі реального часу у співпраці з хмарними обчисленнями ефективність виробництва підвищується, а аномалії обладнання виявляються швидко [15, 16, 28].

Ідея окраїнних обчислень була вперше визначена у 2015 році, а з 2016 року активне розроблення технології почалось з ініціативи NSF та Intel. З того часу edge-обчислення стали важливою частиною інновацій в обчислювальних системах, включаючи платформу LinuxEdgeX та розвиток Kubernetes для використання в хмарних і IoT-системах [8-10, 29]. До основних компонентів системи edge-обчислень [3, 28] (рисунок 1.4) належать:

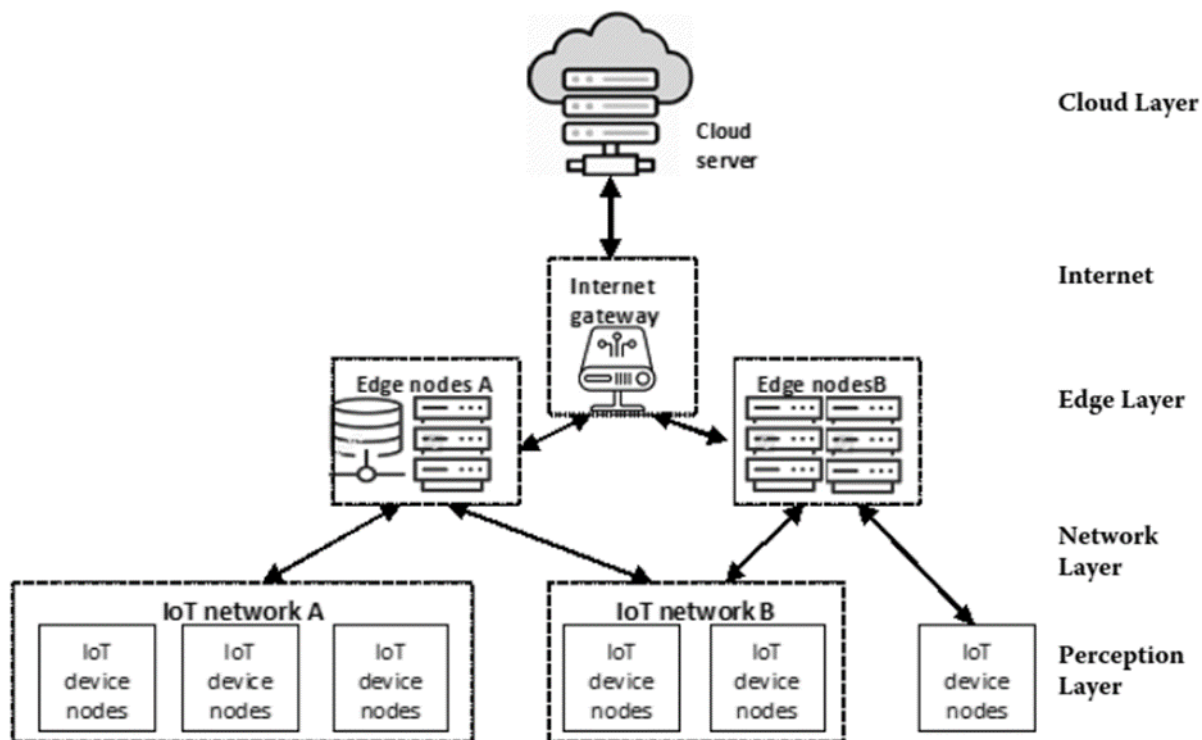


Рисунок 1.4 – Архітектура високого рівня системи edge-обчислень

Edge-обчислення використовують обчислювальні ресурси за межами традиційних центрів обробки даних, розміщуючи їх ближче до джерел даних. Це дозволяє здійснювати кешування, обробку даних, керування Інтернетом речей та забезпечення конфіденційності. Завдяки хмарним обчисленням більшість обчислень перенесено в хмари, але через затримку мережі та витрати на передачу даних, завдання на межі мережі стали більш ефективними. З очікуваним зростанням обчислювальної потужності, завдання на межі мережі також збільшаться [1, 29, 30].

Основні компоненти системи edge-обчислень включають:

- рівень сприйняття: спеціалізовані пристрої з обмеженими обчислювальними і зберігаючими ресурсами, такі як датчики і камери, що збирають і передають дані;
 - мережевий рівень: забезпечує з'єднання пристроїв, окраїнних систем і хмарних серверів через різні протоколи;
 - рівень окраїнних обчислень: включає крайові сервери з більшою обчислювальною потужністю, які виконують завдання, неможливі для пристроїв IoT, а також шлюзи, що управляють мережевими функціями;
 - прикладний рівень: обробка складних завдань з великими обсягами даних, а також використання хмарних серверів для зберігання та обміну даними.
- Ця архітектура підтримує моніторинг, керування та взаємодію з системою [3-7, 31].

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

2.1 Загальна структура проєктованого модуля edge-обчислень

Існує два методи щодо того, як окраїнні пристрої та датчики працюватимуть та взаємодіятимуть з Інтернетом. Датчики та пристрої edge (окраїнного) рівня забезпечуватимуть прямий шлях до хмари. Це означає, що ці вузли та датчики окраїнного рівня матимуть достатньо ресурсів, апаратного та програмного забезпечення, а також угод про рівень обслуговування для безпосередньої передачі даних через глобальну мережу (WAN). Датчики окраїнного рівня формуватимуть агрегації та кластери навколо шлюзів та маршрутизаторів, щоб забезпечити проміжні області, перетворення протоколів та можливості обробки окраїнних даних/туманних сигналів, а також вони керуватимуть безпекою та автентифікацією між датчиками та WAN.

Засоби для edge-обчислень включають: процесори та апаратні аспекти, проектування операційної системи, методи управління, контейнеризація, зберігання та кешування. Хоча ці теми є широкими, необхідно розуміти, як різні аспекти та вибір у проектуванні обладнання можуть впливати на вартість, надійність та цінність. Конструкція окраїнних комп'ютерів настільки різноманітна, що залежить від кожного рішення та клієнта, що несправедливо зосереджуватися на одному випадку використання.

Один із варіантів використання - обробка даних для датчиків розташовані поблизу джерел даних. Тому узагальнимо прийняття обґрунтованих рішень щодо базової технології. Для функціонування edge-обчислень в системі управління виробництвом також потрібно вибрати хмарний інтерфейс.

Обчислювальні платформи, що працюють поза межами корпоративних середовищ та центрів обробки даних, що керуються ІТ, не є чимось новим, Окраїнні системи сьогодні є розширенням загальних компонентів, що керуються ІТ, так, ніби вони знаходяться під контролем та безпекою корпоративного центру обробки даних. Окраїнні системи, по суті, є системами віддалених обчислень (рисунок 2.1) та запозичують елементи з усталених інженерних областей вбудованих систем,

хмарних обчислень, комп'ютерної безпеки та телекомунікацій [31, 32].

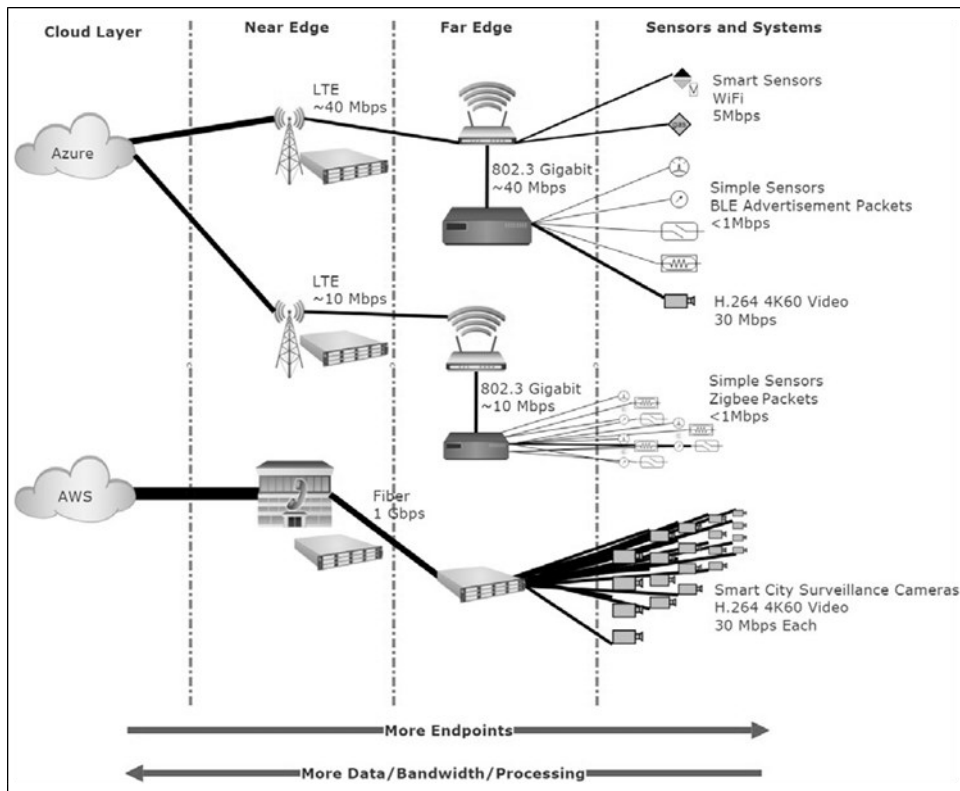


Рисунок 2.1 – Різниця в близькості та ресурсах між обчисленнями на ближньому та далекому краях

Для цілей визначення будемо називати компоненти на ближньому та далекому краї наступним чином. Компоненти на ближньому краї: частина інфраструктури між дальнім краєм та хмарним шаром. Системи на ближньому краї можуть співіснувати з інфраструктурою операторів WAN, такою як колокація обладнання на вежах стільникового зв'язку та комутаційних станціях стільникового зв'язку. Цей рівень може розміщувати складні обчислювальні послуги, такі як програмно-визначена глобальна мережа (SDWAN) (яку буде розглянуто далі в цьому розділі).

Компоненти на далекому краї: складаються з процесорних пристроїв, здатних до зв'язку, управління та обміну даними з хмарними та/або пристроями на ближньому краї. Цей рівень знаходиться найдаліше від загального хмарного рівня, але він все ще підтримує зв'язок з хмарою, а також з її батьківськими компонентами на периферії. Цей рівень знаходиться найближче до кінцевих користувачів або сенсорних систем. Цей рівень має такі вимоги, як жорсткий реальний час або

критично важлива для безпеки конструкція та низька затримка, і може розміщувати великі RAN-мережі як шлюз.

Слід розуміти, що просте підключення датчика до комп'ютерного пристрою у віддаленому місці не є периферійними обчисленнями, так само як людина, яка використовує непідключений ПК для гри у відеогру, не є онлайн-геймером. Однак, як тільки цей датчик передає дані до хмарного сервісу або людина грає в інтерактивну гру віртуальної реальності, вона працює з периферійними обчислювальними пристроями, які повинні діяти узгоджено з централізованими хмарними сервісами. Загалом, чим далі від хмарних рівнів вони знаходяться, тим більше обмежені в ресурсах обчислювальні системи. Однак, кінцевих точок буде більше, оскільки ми віддаляємося від хмарних рівнів у вигляді не-IP-систем, таких як датчики Bluetooth та системи SCADA. Ми називаємо це розширенням розгалуження [17, 20, 23].

Багатоканальні окрайні обчислення (MEC): Раніше відомі як мобільні Окrajні обчислення, MEC дозволяють існувати та розповсюджуватися програмам з низькою затримкою, високою пропускну здатністю та роботою в режимі реального часу на периферії великих мереж. MEC визначено Європейським інститутом телекомунікаційних стандартів (ETSI) і передбачає можливість розробникам запускати програми в мережі радіодоступу (RAN) оператора зв'язку. RAN. Системи та протоколи зв'язку дальнього радіусу дії (WAN). Як правило, RAN фізично існує з контролером радіомережі на базовій станції стільникового зв'язку. MEC може дозволити потокове відео з низькою затримкою або хмарні ігри [18, 20].

2.2 Обґрунтування алгоритмічного забезпечення діагностики експлуатації та технічного обслуговування виробничого обладнання

Діагностика відіграє важливу роль у прийнятті рішень щодо завдань експлуатації та технічного обслуговування виробничого обладнання (машин). Для діагностики стану машини можна використовувати два основні підходи.

Підходи, засновані на даних (просте моделювання трендів, нейронна мережа, розпізнавання образів, статистичний, гістографічний підхід Парето або інші числові підходи). Ці методи, як правило, автоматизовані, не вимагають глибоких знань механізму виникнення та поширення несправностей, але вимагають навчання алгоритму з використанням великого набору спостережуваних даних про несправності.

Підходи, засновані на знаннях, які спираються на чітке представлення поведінки або симптомів несправності, наприклад, за допомогою моделей несправностей, моделей правильної поведінки або опису випадку.

Визнано, що ці методи можуть перетинатися, і можуть бути розроблені рішення, які використовують комбінацію кількох підходів. Слід зазначити, що для аналізу всього процесу моделювання з метою діагностики також необхідне визначення «специфікації спостереження». Ця специфікація може забезпечити процедуру, яка описує та інтерпретує спостереження з отриманих даних. Можуть знадобитися різні рівні деталізації: наприклад, як модель спостереження повинна обробляти описи еволюції параметрів, форми, інформації, пов'язаної з часом, рівнів та кореляцій. Вибір відповідних діагностичних підходів (рисунок 2.2) залежить від:

- застосування або обладнання;
- кінцевого користувача діагностичного підходу;
- методу моніторингу;
- складності знань, що моделюються;
- необхідності мати пояснювальну модель;
- необхідності перенавчання моделі;
- наявності існуючих даних з відомими несправностями та нормальною роботою.

Дуже важливо врахувати на початковому етапі проектування, як розділити обробку діагностики несправностей онлайн та офлайн, беручи до уваги можливості програмного та апаратного забезпечення щодо статистичного аналізу даних та міркування на основі випадків.

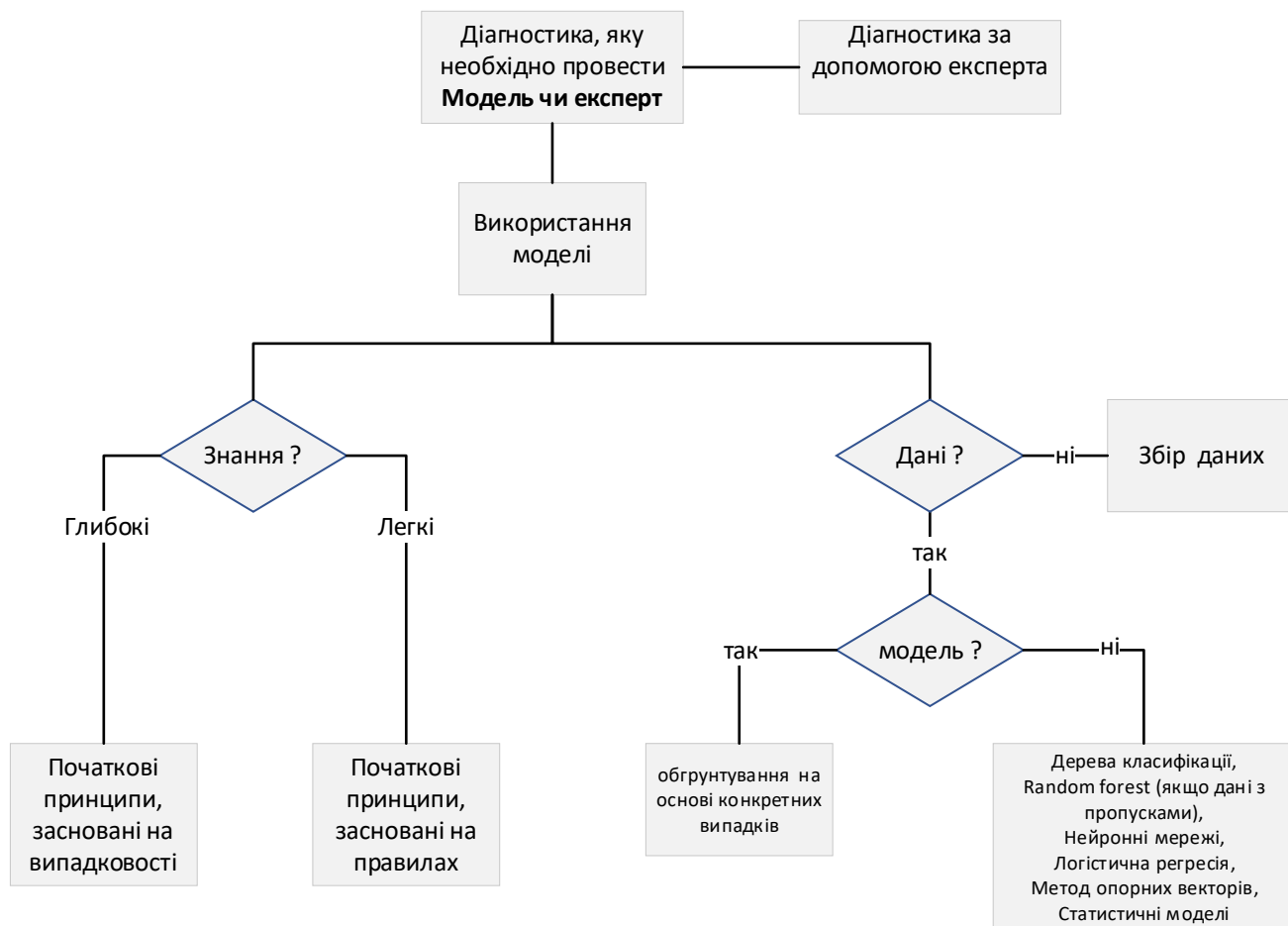


Рисунок 2.2 — Загальний підхід до вибору діагностичної моделі

Основний принцип, що лежить в основі всіх підходів, керованих даними, полягає в використанні моделі для класифікації різних робочих умов машини: нормальних, з помилкою один, з помилкою два тощо. Це досягається шляхом попереднього навчання моделі на раніше записаних даних з кожного стану, а потім подання моделі нових даних для класифікації. Спочатку описуються сильні та слабкі сторони, спільні для всіх підходів, керованих даними. Потім для кожного підходу в наступних абзацах детально описано принцип, деякі рекомендації щодо побудови моделі та конкретні сильні та слабкі сторони, коли це необхідно. Порівняно з підходами до діагностики, заснованими на знаннях, методи, керовані даними, мають перевагу в тому, що не вимагають глибоких знань про систему для діагностики. Крім того, вони не мають обмежень щодо формату незалежних змінних, які можуть бути дихотомічними (наприклад, увімкнено або вимкнено), категоріальними (наприклад, стан машини, такий як «розігрів», «нормальна

робота», «максимальна потужність», «холостий хід») або безперервними (наприклад, вимірювані температури, тиску, швидкості).

Ці типи моделювання мають деякі недоліки для діагностики машин. Перед використанням моделі необхідні фази навчання, калібрування та потенційно налаштування. Можливо, знадобиться повторити ці фази, якщо обладнання або його використання значно зміняться, або коли виникнуть нові випадки. Для побудови моделі необхідна відносно велика кількість ідентифікованих зразків несправностей та відмов обладнання.

2.3 Інформаційне забезпечення діагностики експлуатації та технічного обслуговування виробничого обладнання

Для забезпечення ефективності процедури діагностики виробничого обладнання (машин) слід виконати оцінку змісту та обсяг даних відповідно до несправностей, які можуть виникнути в машині. Тому рекомендується провести попереднє дослідження під час підготовки вимог до системи моніторингу стану та діагностики машини.

Принцип визначення потреб у діагностиці показано на рисунку 2.3. V-подібна форма була навмисно обрана для представлення проблем високого рівня (технічне обслуговування: машина, оцінка ризиків) та проблем «низького рівня» (вимірювання: моніторинг, періодичні випробування, обробка даних).

Ліва гілка відповідає попередньому дослідженню, яке готує для конкретної машини необхідні дані для моніторингу стану та діагностики.

Права гілка на рисунку 2.3 відповідає заходам з моніторингу стану та діагностики, які зазвичай виконуються після введення машини в експлуатацію.

Кожен рівень складається з підготовчої фази проектування (ліворуч) та фази використання (праворуч).

Кроки діагностичного дослідження включають наступне:

- а) аналіз доступності машини, ремонтпридатності, критичності стосовно всього процесу;

- б) перелік основних компонентів та їхніх функцій;
- в) аналіз видів відмов та їх причин як несправностей компонентів;
- г) визначення критичності, враховуючи значущість (безпека, доступність, витрати на обслуговування, якість виробництва) та виникнення;

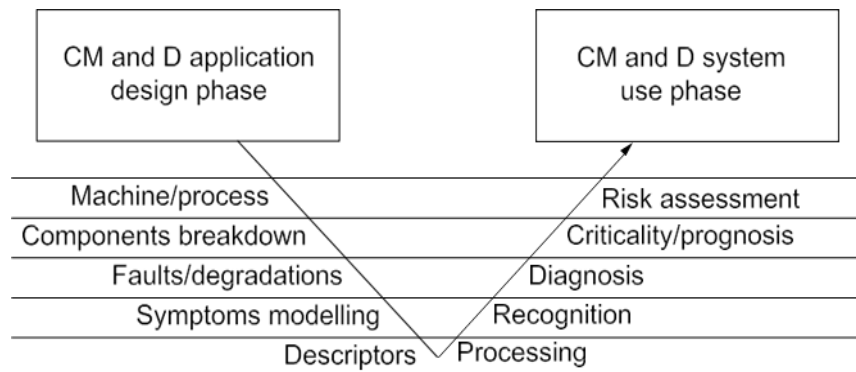


Рисунок 2.3 – Цикл моніторингу та діагностики стану (CM та D): проектування та використання програми на машині

- д) відповідно рішення про те, які несправності повинні бути охоплені діагностикою («діагностовані»);
- е) аналіз, за яких умов експлуатації різні несправності можна найкраще спостерігати, та визначення контрольних умов;
- ж) визначення симптомів, які можуть служити для оцінки стану машини та які використовуються для діагностики;
- з) перелік дескрипторів, які використовуються для оцінки (розпізнавання) різних симптомів;
- и) визначення необхідних вимірювань та перетворювачів, з яких виводяться або обчислюються дескриптори.

Кроки, наведені в а), б), в) та г), можна виконати за допомогою оптимізації технічного обслуговування, такої як FMEA (аналіз режимів та наслідків відмов) або FMESA (аналіз режимів відмов, їх наслідків та критичності). Вони також можуть бути виконані в рамках більш загального процесу оптимізації технічного обслуговування, такого як RCM (технічне обслуговування, орієнтоване на надійність). Кроки, наведені у в), д), е), ж), з), можна виконати за допомогою методології FMSA. Процедури FMEA та FMESA викладено в ІЕС 60812 [33].

Процес аналізу симптомів режимів відмов є вибір технологій та стратегій моніторингу, які максимізують рівень достовірності в діагностиці та прогнозі будь-якого заданого режиму відмов, а також у виборі методів моніторингу, які забезпечують найбільшу чутливість до виявлення та швидкості зміни заданого симптому. Якщо впевненість у чутливості методики та отриманій точності діагнозу/прогнозу є сумнівною, рекомендується використовувати додаткові методи для подальшої кореляції. Цей процес, по суті, є модифікацією процесу FMECA з акцентом на симптомах, що виникають внаслідок кожного виявленого режиму відмови, та подальшому виборі найбільш підходящих методів і стратегій виявлення та моніторингу. Класифікація можливих режимів відмов має такі елементи:

- перелік задіяних компонентів;
- перелік можливих режимів відмови для кожного компонента;
- перелік наслідків кожного режиму відмови;
- перелік причин кожного режиму відмови;
- перелік симптомів, що виникають внаслідок кожного режиму відмови;
- перелік найбільш підходящого методу моніторингу;
- перелік розрахункової частоти моніторингу;
- ранжування кожного режиму відмови за виявленням, тяжкістю, достовірністю діагностики та достовірністю прогнозу, що призводить до визначення номера пріоритету моніторингу (MPN);
- перелік найбільш підходящих методів кореляції;
- перелік частоти моніторингу для методів кореляції.

Найбільша складність виникає у встановленні правильних термінів для режиму відмови, наслідку та причини. Режим відмови - це визначення того, як спостерігатиметься відмова, тобто зігнутий, корозія тощо. У процесах FMECA, які мали бути виконані до процесу FMSA, існують області перекриття між термінами, що використовуються для режимів відмови, наслідків та причин. Елемент може з'являтися як причина відмови в одному рядку, коли розглядається компонент, і як режим відмови в іншому. Термін також може з'являтися як наслідок в одному рядку, коли розглядається компонент, і як режим відмови, коли розглядається вузол. Це також залишається вірним для процесу FMSA. Слід бути обережним, щоб

уникнути дублювання режиму та причини відмови в одному рядку. Для будь-якого одного елемента режим, наслідок та причина відмови повинні логічно зчитатися на сторінці. Може бути корисним використовувати таку форму:

«режим відмови може призвести до наслідку, спричиненого причиною».

Розглядаючи стратегії моніторингу, також можна використовувати таку форму:

- режим відмови викликає симптоми, які найкраще виявляються за допомогою методу первинного моніторингу, що призводить до високої впевненості в діагнозі та прогнозі при моніторингу з заданою частотою моніторингу;
- підвищеної впевненості в діагнозі та прогнозі можна досягти, використовуючи «методи кореляції» при моніторингу з заданою «частотою моніторингу».

Кожному стовпцю присвоюється рейтинг, який оцінює ймовірність виявлення, точність прогнозу та ступінь тяжкості. За умови, що користувач застосовує послідовну оцінку протягом усіх аналізів, категорії з вищим ризиком відображають вищий номер пріоритету моніторингу.

Рейтинг виявлення відмов. Ймовірність виявлення відмов оцінюється від 1 до 5 і призначена для відображення загальної виявлюваності режиму відмови незалежно від наступної точності діагностики чи прогнозу. Цей рейтинг призначений для виділення режимів відмови, які:

- викликають симптоми, які можна виявити, але не повторюються;
- викликають симптоми, які неможливо виявити;
- викликають симптоми, які неможливо виміряти на практиці; або
- викликають симптоми, які можуть бути замасковані іншими симптомами режиму відмови.

Це оцінюється за шкалою від 1 до 5, де:

- 1 означає «Існує НЕВЕЛИКА ЙМОВІРНІСТЬ виявлення цього режиму відмови».
- 2 означає «Існує НИЗЬКА ЙМОВІРНІСТЬ виявлення цього режиму відмови».

- 3 означає «Існує ПОМІРНА ЙМОВІРНІСТЬ виявлення цього режиму відмови».
- 4 означає «Існує ВИСОКА ЙМОВІРНІСТЬ виявлення цього режиму відмови».
- 5 означає «МАЙЖЕ ВПЕВНЕНО, що цей режим відмови буде виявлено».

Ранжування відмов має відображати будь-який попередній аналіз FMESA та розроблено для ранжування окремих режимів відмови за ризиком. Це оцінюється за шкалою від 1 до 4, де:

- 1 означає «Будь-яка подія, яка може спричинити погіршення функції(й) продуктивності системи, що призведе до незначної шкоди для системи або її середовища, та без шкоди для життя чи здоров'я».
- 2 означає «Будь-яка подія, яка погіршує функцію(ї) продуктивності системи без помітної шкоди для системи, життя чи здоров'я».
- 3 означає «Будь-яка подія, яка потенційно може спричинити втрату основної(их) функції(й) системи, що призведе до значної шкоди для зазначеної системи або її середовища та незначної небезпеки для життя чи здоров'я».
- 4 означає «Будь-яка подія, яка потенційно може спричинити втрату основної(их) функції(й) системи, що призведе до значної шкоди системі або її середовищу, та/або спричинить втрату життя чи кінцівок».

Достовірність діагнозу відмов. Прогнозована точність діагнозу також оцінюється від 1 до 5. Цей рейтинг призначений для виявлення режимів відмови з:

- виявленими, але неповторюваними симптомами;
- невідомими симптомами; або
- симптомами, які не відрізняються від інших симптомів режиму відмови.

Це оцінюється за шкалою від 1 до 5, де:

- 1 означає «Існує ВІДДАЛЕНА ЙМОВІРНІСТЬ того, що цей діагноз режиму відмови буде точним».
- 2 означає «Існує НИЗЬКА ЙМОВІРНІСТЬ того, що цей діагноз режиму відмови буде точним».

- 3 означає «Існує ПОМІРНА ЙМОВІРНІСТЬ того, що цей діагноз режиму відмови буде точним».
- 4 означає «Існує ВИСОКА ЙМОВІРНІСТЬ того, що цей діагноз режиму відмови буде точним».
- 5 означає «МАЙЖЕ ВПЕВНЕНО, що цей діагноз режиму відмови буде точним».

Частота моніторингу також сприяє визначенню точності очікуваного прогнозу, тобто чим більша частота моніторингу, тим вища впевненість у очікуваному коефіцієнті відмов та прогнозі.

Номер пріоритету моніторингу (MPN). Ранжування є множенням чотирьох попередніх ранжування та призводить до загального рейтингу кожного режиму відмови. Високе значення MPN вказує на те, що призначений метод є найбільш придатним для виявлення, діагностики та прогнозування пов'язаного режиму відмови. Слід зазначити, що низьке значення MPN не означає, що моніторинг не є необхідним, а радше означає, що за допомогою призначеного методу та частоти моніторингу можна очікувати низького рівня впевненості у виявленні, аналізі та прогнозі.

Найменш сприятливий випадок – це режим відмови з високою вагою, низькою виявлюваністю, низькою впевненістю діагностики та низькою впевненістю прогнозу. Найбільш сприятливий випадок – це режим відмови з низькою вагою, який легко виявляється, з відомими режимами відмови та пов'язаними з ними закономірностями, а отже, високим рівнем впевненості діагностики та прогнозу.

Таким чином, впровадження системи перевірки та моніторингу FMSA повинно здійснюватися з урахуванням:

- ризику безпеки кожного режиму відмови;
- очікуваної швидкості погіршення кожного режиму відмови;
- середнього часу між відмовами (MTBF) для кожного режиму відмови;
- вторинних/наступних режимів відмови;
- взаємозв'язків між режимами відмови;
- необхідного часу виконання технічного обслуговування;

- наявності запасних частин;
- необхідної надійності та доступності.

Постійну переоцінку слід проводити після отримання досвіду роботи з новою установкою або після проведення модифікації устаткування .

Звіт про вимоги до діагностики зазвичай повинен включати:

- представляти прийняте розбиття машини на компоненти;
- перераховувати несправності, пов'язані з цими компонентами;
- наводити потенційно спостережувані симптоми для кожної несправності;
- називати дескриптори моніторингу стану, які будуть використовуватися;
- вказати метод та параметри, що використовуються для розрахунку дескрипторів.

Може статися так, що не всі критичні несправності охоплюються моніторингом стану і, як такі, не піддаються діагностиці. З цієї причини рекомендується чітко виділити у звіті несправності, які враховуються, та ті, які не враховуються. Це може бути корисним для переоцінки цінності додавання можливості виявлення конкретних несправностей.

Формально звіт про вимоги до діагностики може складатися з двох частин:

- опис машини : ідентифікація, роль у процесі, компоненти, аналіз критичності;
- аналіз режимів згідно вище перелічених симптомів відмов режимів відмов, симптоми, дескриптори та вимірювання, які будуть використовуватися для діагностики.

Процес діагностики зазвичай запускається виявленням аномалії під час рутинного моніторингу, рутинного аналізу, випадкового аналізу або людського сприйняття. Це виявлення здійснюється шляхом порівняння поточних дескрипторів машини та еталонних значень (зазвичай званих базовими значеннями або даними), обраних з досвіду, зі специфікацій виробника, з пусконаладжувальних випробувань або обчислених зі статистичних даних (наприклад, довгострокового середнього значення). Можливий результат будь-якого діагностичного процесу повинен включати сценарій, коли несправності немає.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ ТА ПЕРЕВІРКА ЙОГО ФУНКЦІОНУВАННЯ

3.1 Вибір технологій та засобів для реалізації модуля

Взаємодія та керування виробництвом на основі edge-обчислень показано на рисунку 3.1. Окраїнні обчислення працюють в ізольованих окраїнних вузлах на відміну від туманних обчислень, які спираються на взаємозв'язок між вузлами. Окраїнні обчислення надають відповідні послуги поблизу джерела даних для задоволення критичних вимог до гнучкого підключення, оптимізації в режимі реального часу, інтелектуальних додатків, безпеки та конфіденційності.

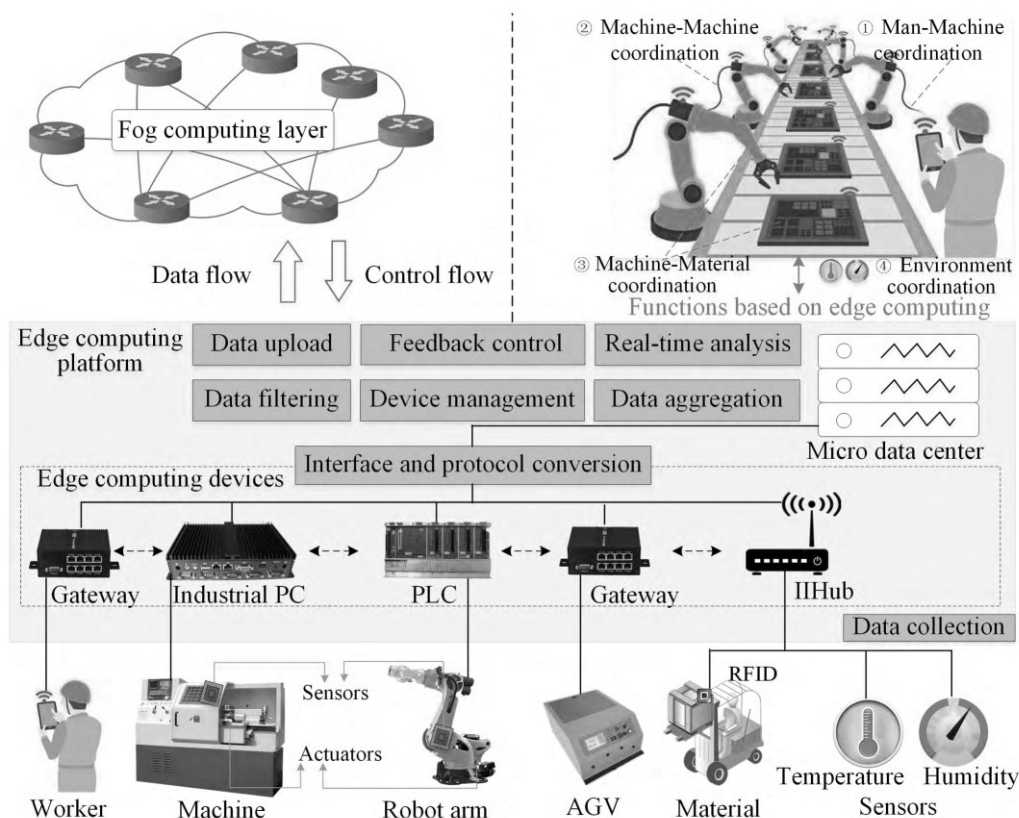


Рисунок 3.1 – Взаємодія та управління виробництвом на основі окраїнних обчислень

Окраїнний шар складається з усіх виробничих ресурсів, що беруть участь у різних видах виробничої діяльності. У цеху або на заводі основні ресурси можна підсумувати як «Людина-Машина-Матеріал-Середовище». Людина стосується працівників та менеджерів. Машина включає обробне обладнання (наприклад,

верстати, роботи тощо), транспортне обладнання (наприклад, конвеєри для матеріалів, автоматично керовані транспортні засоби).

Матеріал складається з сировини, матеріалів, деталей та компонентів, напівфабрикатів, готової продукції тощо. Крім того, щоб бути в курсі змін у реальному часі, на виробничих ресурсах розгортається велика кількість різноманітних сенсорних пристроїв (наприклад, різні інтелектуальні датчики, RFID тощо). Разом з виконавчими механізмами та драйверами вони утворюють сенсорну систему та систему виконання. Крім того, слід також контролювати фактори навколишнього середовища, такі як температура, вологість, тиск тощо. Ці ресурси тісно співпрацюють один з одним для виконання різних виробничих дій на периферійному рівні.

Засоби виробництва: верстати з ЧПК, інтелектуальні роботи, датчики, термінали та пристрої крайнього підключення не тільки виробляють дані, але й мають можливості аналізу і обробки даних та надає інтелектуальні послуги ближнього рівня. На кордоні, найближчому до фізичного обладнання, периферійний рівень використовує обмежені апаратні ресурси (наприклад, шлюз, промисловий ПК, ПЛК, ПHub) для завершення збору даних, перетворення протоколів, фільтрації даних, завантаження даних, зберігання даних, аналізу даних та інших операцій. Перш за все, дані про стан у режимі реального часу (наприклад, звичайна фізична величина, умови навколишнього середовища та експлуатації) та різні параметри виробничих ресурсів та користувачів збираються в режимі реального часу.

Тому, першочерговим завданням пристроїв edge-обчислень є перетворення інтерфейсу та протоколу для перетворення різних форматів даних та протоколів зв'язку в єдиний протокол. Взаємодія даних на периферійному рівні реалізується на основі єдиного протоколу. Крім того, для подальшої обробки зібрані дані також піддаються обробці шуму, очищенню від аномальних даних, за необхідності, стисканню даних. Крайінні пристрої також мають певну можливість зберігання даних (може бути невеликою) для зберігання даних про стан у режимі реального часу, а також сигналів тривоги, несправностей та іншої інформації.

Крім того, мікроцентр обробки даних ефективно використовується для зберігання даних про стан виробничого поля в режимі реального часу. Оскільки обсяг даних великий, пристрої edge-обчислень перезаписувати вихідні дані протягом певного періоду часу новими даними. Щоб заощадити кошти на передачу даних та пропускну здатність, лише необхідні окраїнні дані завантажуються на верхній рівень. Крім того, значна частина зібраних даних може не мати великої цінності, лише повідомляючи про «нормальні» умови. Тому перед тим, як окраїнні дані будуть перезаписані, edge-обчислювальні пристрої будуть фільтрувати, відкидати або більш ефективно повторно збирати дані для передачі в туман або хмару для зберігання та аналізу. Для зібраних даних у режимі реального часу окраїнні обчислювальні пристрої можуть виконувати просту обробку та аналіз даних такої як виявлення порогу або фільтрація за алгоритмом показаним на рисунку 3.2. Надсилаючи лише корисні та готові до використання дані, ця програма мінімізує зв'язок. Наприклад, завдяки аналізу даних різних датчиків, окраїнні обчислення можуть визначити, чи є стан виробничого обладнання нормальним, і яку дію воно виконує.

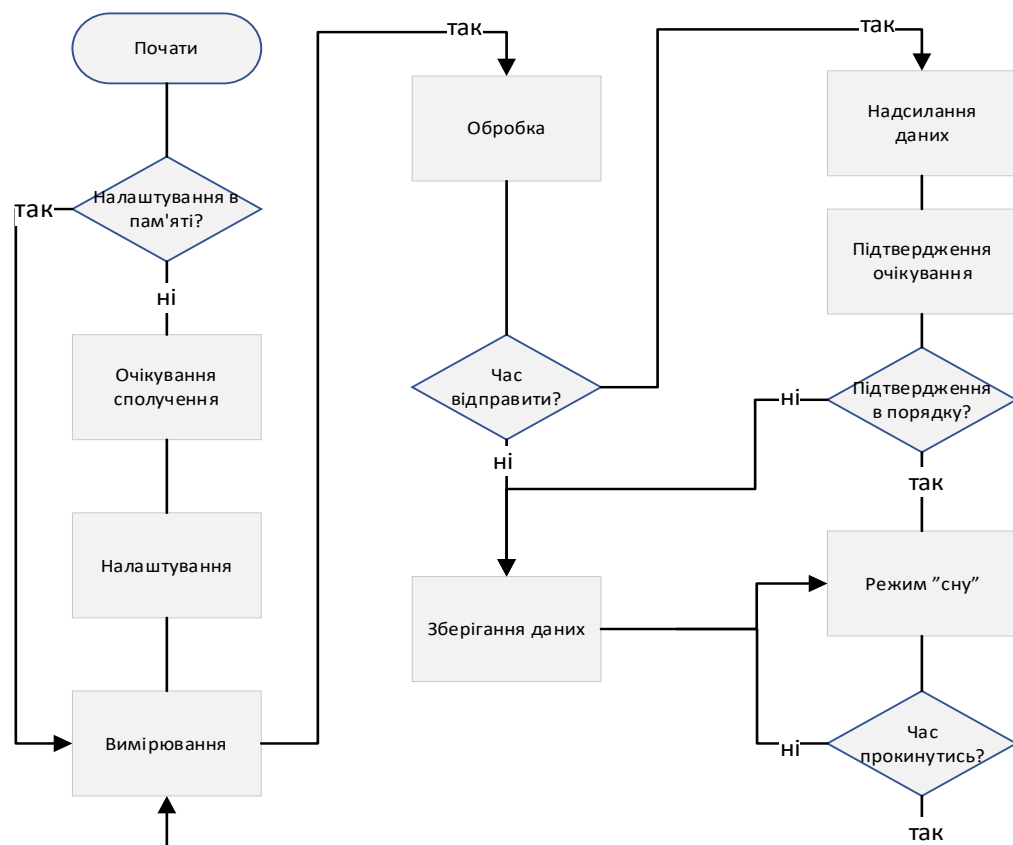


Рисунок 3.2 – Алгоритм попередньої обробки даних для edge-обчислень

Другий варіант використання спирається для бездротової передачі необроблених даних з окраїнних пристроїв безпосередньо на edge-обчислювач. Останній виконує всю обробку даних та відображає динаміку значень. Цей сценарій використовується для відстеження розвитку процесу або фізичної величини в часі. У більшості випадків, інтервал між двома вимірами становить близько секунди. При цьому, edge- обчислювач може бути пов'язаний не лише з одним, з багатьма окрайнними пристроями. На рисунку 3.3 показано відповідну блок-схему.

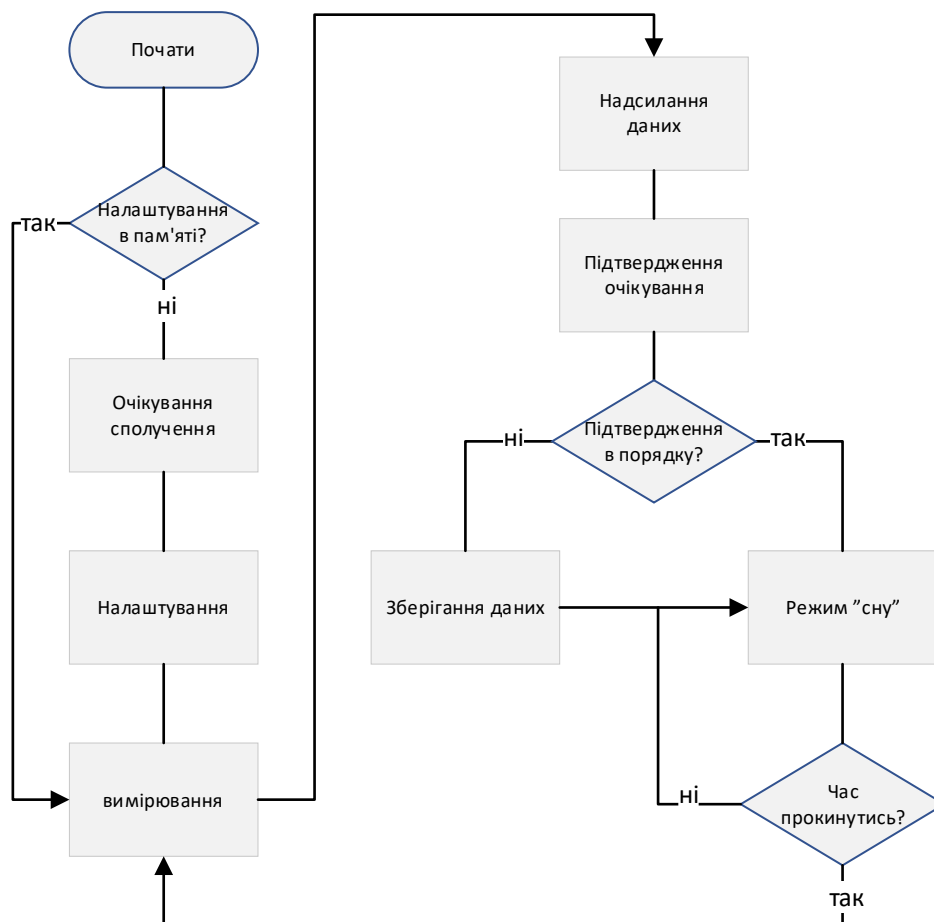


Рисунок 3.3 – Алгоритм підсумкової обробки даних і відображення динаміки відстеження фізичної величини процесів в часі

Крім того, після простої обробки та аналізу даних можна реалізувати автоматичне керування зворотним зв'язком для роботи виробничого обладнання відповідно до заздалегідь встановлених правил. Продуктивності окраїнних обчислень є достатньою для забезпечення керування зворотним зв'язком в режимі

реального часу. Навіть у випадку офлайн-роботи, Окраїнні обчислення все ще здатні надавати локалізовані прикладні послуги.

На основі реалізації інтелектуального сприйняття та взаємозв'язку, можна реалізувати такі функції на основі окраїнних обчислень:

а) координація людина-машина, завдяки інтеграції точності машини та гнучкості людини, виробнича система може бути швидко налаштована для адаптації до динамічної виробничої діяльності, як результат, людина та машини можуть гармонійно співіснувати;

б) координація машина-машина, машини співпрацюють одна з одною, що може ефективно підвищити швидкість та ефективність. Завдяки співпраці машини, виробнича система може мати кращу стійкість та здатність саморегуляції. Як результат, виробництво буде більш гнучким, щоб більш ефективно виробляти різноманітну продукцію;

в) координація машина-матеріал, на основі датчиків, RFID, машинного зору тощо, матеріали та продукти можуть розпізнаватися та позиціонуватися як інтелектуальної, з огляду на окраїнні обчислення, розумні машини автоматично виконують відповідну програму обробки;

г) координація середовища, виробниче середовище динамічно змінюється для забезпечення якості продукції та безпеки обладнання відповідно до конкретних вимог.

Для реалізації edge-обчислень обираємо спеціалізовані апаратні компоненти, призначені для запуску програм на краю комп'ютерного середовища.

3.2 Реалізація ключових компонентів модуля для окраїнних обчислень

Процесори є ключовим компонентом систем окраїнних обчислень, забезпечуючи обчислювальну продуктивність та можливість, необхідні для різних завдань та програм. Більша кількість ядер процесора забезпечує швидшу обробку, що робить його придатним для пристроїв, які потребують швидкої обробки даних.

Розумне виробництво має велику кількість пристроїв, які генерують миттєві дані, тому як процесорний блок модуля для edge-обчислень (ЕО) використовуємо

пристрій Raspberry PI 4, який має однорідну архітектуру на основі мікроконтролера STM32L496 та процесора ARM Cortex-M4. Модуль ПО має ємність пам'яті 1 МБ від STM32 та 8 МБ від зовнішньої флеш-пам'яті (сховище пристрою класу 0), а також слот для SD-карти. У STM32 також є 320 КБ пам'яті, що відповідає пам'яті класу 0.

Використання мікроконтролера STM32L496 та процесора ARM Cortex-M4 дає змогу створити рішення ЕО з використанням технології Інтернет речей, зрозуміле середовище для програмування і в цілому зручні засоби для удосконалення та розвитку. Популярність платформи Raspberry в тому, що вона має доступне середовище розробки і мови програмування для мікроконтролерів на основі C/C++ доповнений простими і зрозумілими функціями для управління введенням/ виводом на контактах мікроконтролера. На контролер програми переносяться через USB (не потрібний програматор). Перевагою використання є те, що Raspberry має відкритий початковий код (та основа, на якій створюється платформа, її програмне ядро, за допомогою якого і створюються усі потрібні програми, в тому числі користувачами за власними вимогами).

Організація підключення окраїнних пристроїв до мережі інтернет через протокол MQTT. Розглянемо засоби, щоб отримати доступ для моніторингу і управління з будь-якої локації по мережі Інтернет. Для організації доступу мережі Інтернет для контролера потрібні зовнішні модулі. В якості зовнішнього модуля часто використовують модулі ESP8266, плата Ethernet Shield W5100 або W5500.

Обрано організацію доступу в Інтернет з використанням GPRS Shield - плати розширення, яка дає змогу працювати в мережах стільникового зв'язку по технологіях GSM/GPRS для прийому і передачі даних, SMS і голосовому зв'язку. В якості GPRS - Shield ми використовуватимемо модуль GSM/GPRS SIM800L з антеною на платі (рисунок 3.4).

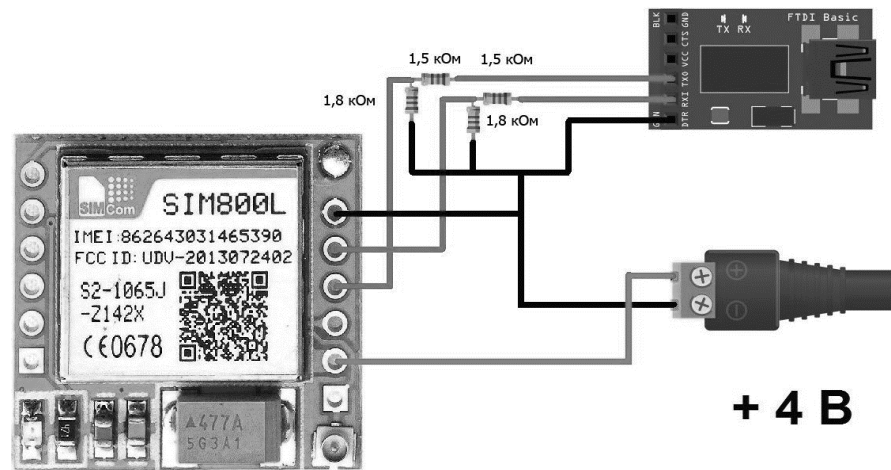


Рисунок 3.4 – Підключення модуля GSM/GPRS SIM800L до мікроконтролера

Стандартний інтерфейс управління компонента SIM800L надає доступ до сервісів мереж GSM/GPRS 850/900/1800/1900 МГц для відправки дзвінків, Смс-повідомлень і обміну цифровими даними GPRS. Також можна підключити додаткові антени для поліпшення якості сигналу. Цей модуль має 1 слот під мікро- SIM -карту. Швидкість модуля за умовчанням 9600. Модуль підтримує AT-команди.

Управління модулем GSM/GPRS SIM800L за допомогою AT – команд. Для цього підключимо модуль до комп'ютера, використовуючи перехідник FTDI і дільник напруги. Для живлення модуля використовуватимемо перетворювач напруги на мікросхемі LM2596, на якому встановимо напругу живлення модуля 4 В. Підключаємо модуль до комп'ютера і запускаємо програму обміну даними по послідовному порту. Можна використати монітор послідовного порту. Відправлятимемо на модуль GSM/GPRS SIM800L AT -команди. Завантажуємо скетч з лістингу. Після цього підключимо модуль GSM/GPRS SIM800L до мікроконтролера і відправити повідомлення с даними на сервер, наприклад <http://sparkfun.com>.

Підключимо модуль GSM/GPRS SIM800L до плати мікроконтролера і зробимо тестову відправку даних по GPRS на віддалений сервер. В якості віддаленого сервера для відправки даних зручно використати сервер data.sparkfun.com – для якого не потрібно реєстрації і можна швидко перевірити

правильність відправки даних. На головній сторінці переходимо по посиланню створення «вільного» потоку даних (рисунк 3.5).

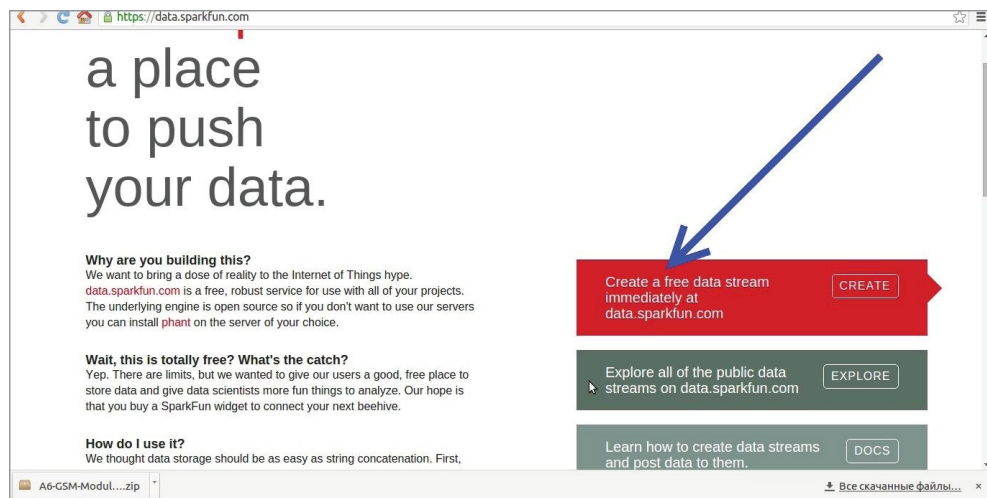


Рисунок 3.5 – Посилання для створення «вільного» потоку даних

На сторінці створення потоку заповнюємо необхідні дані: назва (title), опис (description), приватність (visible/hidden), список полів (fields), і підтверджуємо.

При створенні потоку генеруються публічний ключ (Public key), приватний ключ (Private key), які нам знадобляться при створенні скетчу відправки даних, і сторінка для перегляду відправлених даних (рисунк 3.6).

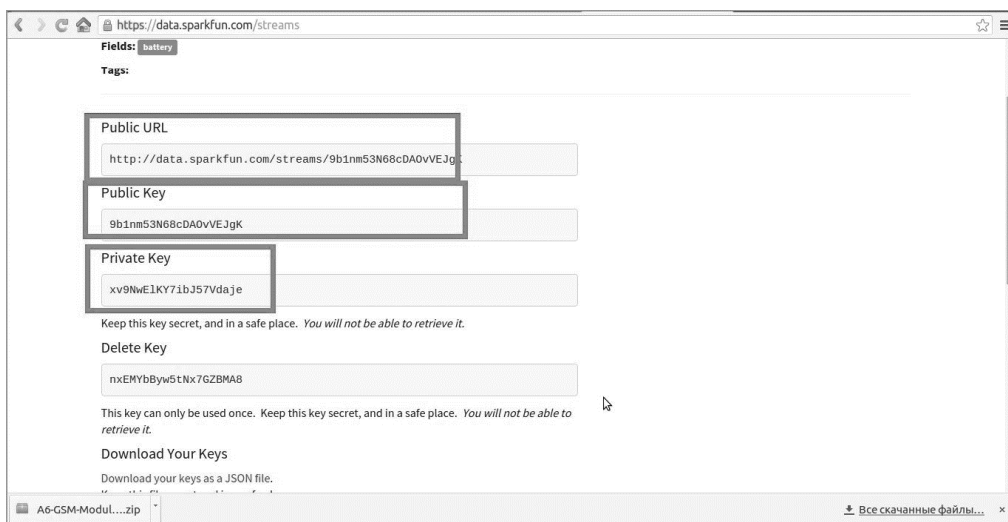


Рисунок 3.6 – Дані необхідні для створення програми відправки

Для підключення модуля до мережі Інтернет необхідно підключитися по Wi - Fi до точки доступу, яка має вихід в Інтернет. Завантажимо в плату модуля скетч

підключення до Wi - Fi точки доступу для відправки даних в сервіс data. sparkfun.co, при цьому для підключення до точки доступу в сервісі data.sparkfun.com вводимо:

```
const char* ssid = "my_point"; const char* password = " my_point pass";  
const char* host = "data.sparkfun.com";  
const char* streamId = "  
const char* privateKey = "
```

Після завантаження скетчу на плату запускаємо монітор послідовного порту і спостерігаємо процес відправки даних з аналогового входу в сервіс data.sparkfun.com кожної секунди.

Обираємо найбільш популярний і одночасно простий протокол обміну для облаштувань Інтернету речей - Message Queue Telemetry Transport (MQTT). Основні особливості протоколу MQTT:

- асинхронний протокол;
- компактні повідомлення;
- робота в умовах нестабільного зв'язку на лінії передачі даних;
- підтримка декількох рівнів якості обслуговування (QoS);
- легка інтеграція нових пристроїв.

Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем або підписником (publisher/subscriber) повідомлень і брокером (broker) повідомлень (наприклад, Mosquitto MQTT). Видавець і підписник не передають один одному повідомлень безпосередньо, і не встановлюють прямого контакту, можуть не знати про існування один одного. Видавець відправляє дані на MQTT брокера, вказуючи в повідомленні певну тему, (topic). Передплатники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки.

Топіки є символами з кодуванням UTF - 8. Ієрархічна структура топіків має формат «дерева», що спрощує їх організацію і доступ до даних. Топіки складаються з одного або декількох рівнів, які розділені між собою символом «/»,/home/living /living - room/temperature.

Облаштування MQTT використовують певні типи повідомлень для взаємодії з брокером, нижче представлені основні:

- connect - встановити з'єднання з брокером;
- disconnect - розірвати з'єднання з брокером;
- publish - опублікувати дані в топик на брокерові;
- subscribe - підписатися на топик на брокерові;
- unsubscribe - відписатися від топика.

В якості брокерів вибираємо сервіс CloudMQTT.com (<https://www.cloudmqtt.com/>) в якому можна створити безкоштовний аккаунт (по посиланню Control Panel). Для реєстрації необхідно ввести адресу електронної пошти (в якості логіна) і пароль. Після цього в панелі управління і створюємо брокер (натиснення по кнопці +Create). Вводимо назву, вибираємо датацентр (Європа), тарифний план - безкоштовний Cute Cat і зберігаємо. При цьому можна створити декілька брокерів. Тепер натискаємо на кнопку Details і переходимо до налаштувань вибраного брокера:

- ім'я хоста m13.cloudmqtt.com;
- порт 18274 (для скетчу мікроконтролера);
- WebSockets порт 38274 (для мобільного застосування).

Тут же знаходиться менеджер користувачів, де можна створити користувачів для доступу до даних брокера і призначити їм права (Read, Write). У поле Topic вводимо # (до всіх топиків). Тепер встановлюють мобільний додаток IoT Manager. Запускаємо. Необхідно зробити налаштування. Натискаємо на Settings і у формі, що з'явилася, вносимо дані свого брокера:

- MQTT hostname - m13.cloudmqtt.com;
- MQTT Websocket port - 38274;
- MQTT username - nodemcu1.

Тепер виходимо на сторінку Dashboard, тут ми повинні побачити встановлене з'єднання з написом No data, яке не повинно бентежити - дані в топіки ще не передавалися. Проте IoT Manager не лише підписаний на теми, але і виступає в ролі publisher - публікує дані в теми. Це значення слайдерів і статус кнопки. Ці дані модуль, підписані в якості subscriber на ці теми, може використати для управління підключеними до плати пристроями.

Розглянемо відправку даних з окраїнних пристроїв брокерові. Відправлятимемо брокерові дані з двох окраїнних пристроїв – ДПП11 і ДПП22. Правки здійснюємо в скетчі з попереднього розділу. Встановлюємо кількість віджетів для відображення по кількості датчиків. Функції отримання даних з ДПП11 - `get_data_hu - midity()`. Завантажуємо повний скетч на плату модуля, відкриваємо монітор послідовного порту і бачимо відправку даних з окраїнних пристроїв.

У процедурі `initVarO` прописуємо налаштування для віджетів управління виконавчими пристроями. Нам потрібний віджет `toggle`. Параметру `page[]` привласнюємо значення `RelaysNo - demsu`, що дозволить на екрані мобільного додатка IoT Manager розділяти датчики і виконавчі пристрої по різних вкладках.

3.3 Опис проведення тестування програмного забезпечення

Оскільки модуль містять мікроконтролери, програмне забезпечення є невід'ємною частиною його роботи. Тестування програмного забезпечення під час розробки активно використовує модульне тестування, де кожна функція модуля має набір модульних тестів, які перевіряють очікувану роботу на відповідність вимогам (завдання). Цей тип тестування називається «білою скринькою», оскільки він використовує можливість аналізувати код «всередині» програми модуля.

У цій роботі, орієнтованій на виробництво, тестування має на меті описати тестування модуля (фізичного пристрою) без підключення до речей усередині нього, тому тестування буде здійснювались через доступні точки доступу зовні пристрою, звідси таке тестування має назву «тестування чорної скриньки». Таким чином доступ до програмного забезпечення здійснюється через функції, доступні користувачеві. Під час проведення функціонального тестування модуля, який використовує програмне забезпечення, конкретний метод тестування спрямований на перевірку функції модуля на відповідність вимогам. Під час проектування тестування (валідації прийнятих рішень) проводиться один раз.

У зв'язку із складною поведінкою використання модуля на виробництві, реалізована модель імітації його поведінки (рисунок 3.7).

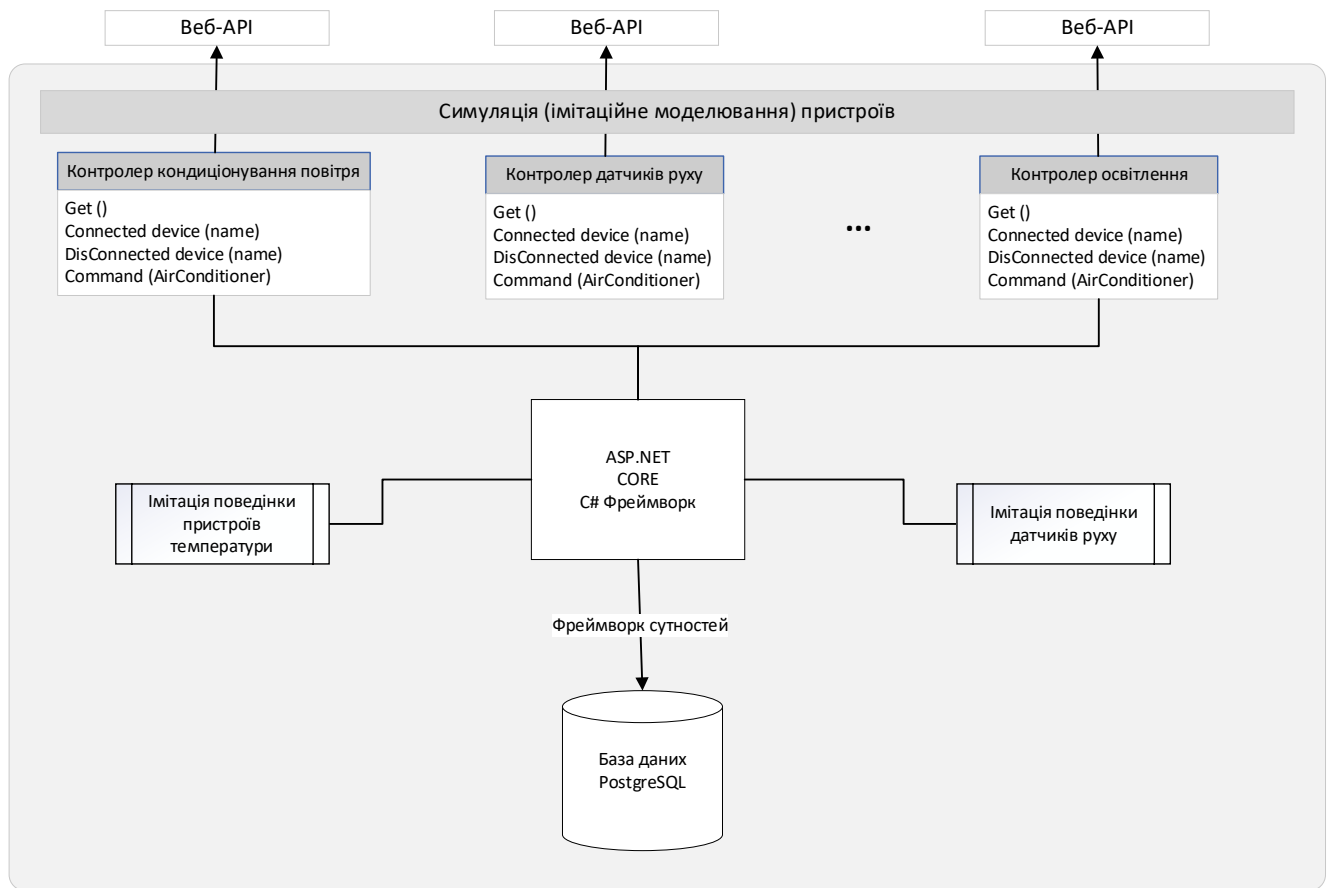


Рисунок 3.7 – Загальна структура системи імітації

При цьому модуль взаємодіє з симуляцією пристроїв. Симуляція, приймаючи запити міняє стан існуючих пристроїв. На кожную категорію пристроїв, в симуляції передбачений контролер, із своїм набором API команд.

Окрім контролерів, в системі асинхронно виконують імітацію поведінки наявних пристроїв. Дані про існуючі пристрої зберігаються у базі даних. Якщо необхідно додати в систему новий пристрій, досить визначити його у базі.

Особливості бази даних. Усі класи наслідують від батьківського класу Device. Клас Device відрізняється від аналогічного класу в основній системі тим, що в ньому відсутні поля Category. Це пояснюється тим, що самим пристроям немає необхідності зберігати інформацію про їх місце знаходження, тому що ця інформація міститься в основній системі.

Поле Category відсутнє тому що основна система зберігає усі пристрої однаковим чином, а клас повинен містити дані про те, до якої категорії він належить.

Алгоритми роботи системи симуляції пристроїв схожі на алгоритми роботи фізичної системи управління і автоматизації, за тією відмінністю, що в системі симуляції дані про стан пристроїв подають у форматі JSON, а користувачем виступає основна система (рисунок 3.8).

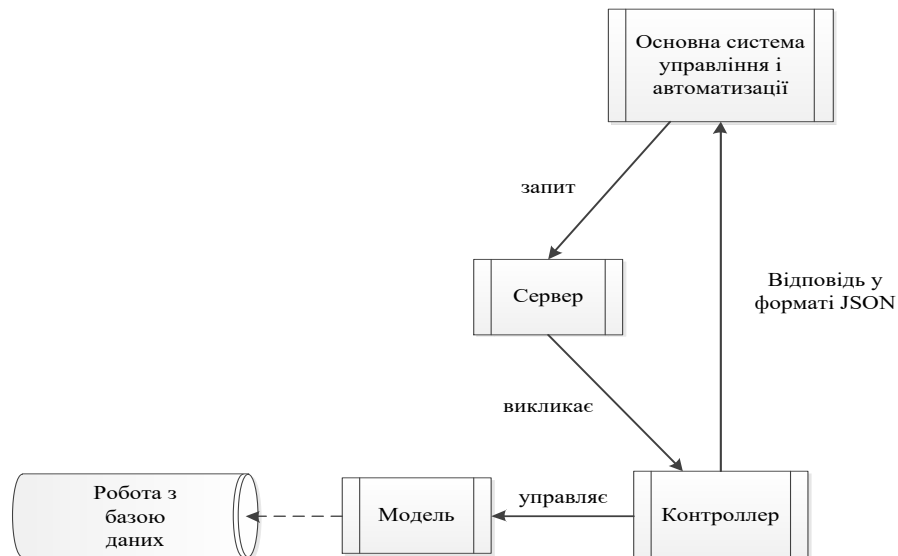


Рисунок 3.8 – Загальний алгоритм роботи системи симуляції

ВИСНОВКИ

У підсумку виконання кваліфікаційної роботи присвяченої темі розроблення модуля edge-обчислень для цифрового розвитку інноваційного виробництва отримано наступні висновки:

1. Аналіз існуючих рішень для обчислень у виробничих системах реального часу показує, що все більше ресурсів зараз переміщується з хмари на край мережі, оскільки передача даних на краю забезпечує скорочення часу реакції та кращу надійність. Фактично, пропускну здатність можна зменшити, якщо значний обсяг даних зберігати на краю, а не завантажувати в хмару. У обчислювальній парадигмі від збирача даних до виробника/користувача даних, розвиток Інтернету речей та мобільних застосунків змінив становище краю. Обробка або розвантаження даних на краю мережі є більш ефективною.

2. Edge-обчислення використовують обчислювальні ресурси за межами традиційних центрів обробки даних, розміщуючи їх ближче до джерел даних, основні компоненти включають: рівень сприйняття, мережевий рівень, рівень окраїнних серверів обчислень і шлюзів управління мережевими функціями, прикладний рівень.

3. Розроблено концепцію архітектури модуля, що включає мікроцентр обробки даних для зберігання даних про стан виробничого поля в режимі реального часу, edge-обчислювальні пристрої фільтрують, відкидають або більш ефективно повторно збирають дані для передачі використовуючи алгоритм попередньої обробки даних, надсилаючи лише корисні та готові до використання дані, мінімізуючи потреби каналів зв'язку, також edge-обчислювач виконує всю обробку даних та відображає динаміку значень діагностики виробництва.

4. Розроблено програмно-технологічне забезпечення на основі мікроконтролера STM32L496 та процесора ARM Cortex-M4, підключення через протокол MQTT, віддаленого сервера відправки даних data.sparkfun.com.

5. Тестування програмного забезпечення під час розробки активно використовує модульне тестування, де кожна функція модуля має набір модульних тестів, які перевіряють очікувану роботу на відповідність завданням для

забезпечення можливості аналізувати код «всередині» програми модуля; у зв'язку із складною поведінкою використання модуля на виробництві, реалізована модель імітації його поведінки.

Отже для розвитку інноваційного цифрового виробництва, коли мілісекунди мають значення, наприклад, в автоматизованому виявленні дефектів на виробничій лінії у режимі реального часу, затримки, спричинені передачею даних у хмару та очікуванням відповіді, можуть призвести до втрачених можливостей. Ці затримки не тільки впливають на ефективність, але й можуть мати серйозні наслідки для безпеки. Для безпосереднього вирішення цих проблем доцільно застосовувати методи edge-обчислень, забезпечивши значне скорочення часу обробки даних та підвищення надійності управління обслуговуванням обладнання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vărzaru, A. A., & Bocean, C. G. (2024). Digital transformation and innovation: The influence of digital technologies on turnover from innovation activities and types of innovation. *Systems*, 12(9), 359.
2. Boy, G. A. (2021). Articulating human systems integration. In *Design for Flexibility: A Human Systems Integration Approach* (pp. 29-47). Cham: Springer International Publishing.
3. Bayoumi, A. M., Matthews, R. M., & Abdel Fatah, A. A. (2020). The Fourth Industrial Revolution: Digital Transformation and Industry 4.0 Applied to Product Design, Manufacturing and Operation. In *Advances in Asset Management and Condition Monitoring: COMADEM 2019* (pp. 1023-1037). Cham: Springer International Publishing.
4. Pinheiro, P., Putnik, G. D., Putnik, Z., & Alves, C. (2022, June). Industry 4.0: from Illusion to Revolution through Digital Transformation. In *International Symposium on Industrial Engineering and Automation* (pp. 189-200). Cham: Springer International Publishing.
5. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39.
6. Fisher, O., Watson, N., Porcu, L., Bacon, D., Rigley, M., & Gomes, R. L. (2018). Cloud manufacturing as a sustainable process manufacturing route. *Journal of manufacturing systems*, 47, 53-68.
7. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5), 637-646.
8. Lian, J. W., Yen, D. C., & Wang, Y. T. (2014). An exploratory study to understand the critical factors affecting the decision to adopt cloud computing in Taiwan hospital. *International journal of information management*, 34(1), 28-36.
9. Qi, Q., & Tao, F. (2019). A smart manufacturing service system based on edge computing, fog computing, and cloud computing. *IEEE access*, 7, 86769-86777.
10. Ai, Y., Peng, M., & Zhang, K. (2018). Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks*, 4(2), 77-86.

11. Sunyaev, A., & Sunyaev, A. (2020). Fog and edge computing. *Internet computing: principles of distributed systems and emerging internet-based technologies*, 237-264.
12. Pan, J. and McElhannon, J., 2017. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, 5(1), pp.439-449
13. Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017, May). Internet of Things (IoT) communication protocols. In *2017 8th International conference on information technology (ICIT)* (pp. 685-690). IEEE.
14. Taherizadeh, S., Jones, A. C., Taylor, I., Zhao, Z., & Stankovski, V. (2018). Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review. *Journal of Systems and Software*, 136, 19-38.
15. Bosch implements edge computing for real-time quality control," Bosch, 2020. [Online]. Available: <https://www.bosch.com/stories/edge-computing-in-manufacturing/>
16. "Schneider Electric deploys edge computing for predictive maintenance," Schneider Electric, 2021. [Online]. Available: <https://www.se.com/ww/en/work/campaign/innovation/edgecomputing.jsp>
17. ETSI, F. (2014). Sophia Antipolis, "Mobile-edge computing introductory technical white paper," Mobile-Edge Comput. *Ind. Initiative, White Paper*.
18. Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., & Giust, F. (2016). Mobile-edge computing architecture: The role of MEC in the Internet of Things. *IEEE Consumer Electronics Magazine*, 5(4), 84-91.
19. Taleb, T., Dutta, S., Ksentini, A., Iqbal, M., & Flinck, H. (2017). Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3), 38-43.
20. Alli, A. A., & Alam, M. M. (2020). The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications. *Internet of Things*, 9, 100177.
21. Turner, V., Gantz, J. F., Reinsel, D., & Minton, S. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16, 13-19.
22. Shi, W., Zhang, X., Wang, Y. F., & Zhang, Q. (2019). Edge computing: state-of-the-art and future directions. *Journal of Computer Research and Development*, 56(1), 69-89.

23. Shi, W., Sun, H., Cao, J., Zhang, Q., & Liu, W. (2017). Edge computing-an emerging computing model for the internet of everything era. *Journal of computer research and development*, 54(5), 907-924.
24. Hong, X., & Wang, Y. (2018). Edge computing technology: Development and countermeasures. *Strategic Study of Chinese Academy of Engineering*, 20(2), 20-26.
25. Putnik, G., Sluga, A., ElMaraghy, H., Teti, R., Koren, Y., Tolio, T. and Hon, B., 2013. Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. *CIRP Annals*, 62(2), pp.751-774.
26. Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE access*, 8, 85714-85728.
27. Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2021). Edge computing: current trends, research challenges and future directions. *Computing*, 103(5), 993-1023.
28. Jain, K., & Mohapatra, S. (2019). Taxonomy of edge computing: Challenges, opportunities, and data reduction methods. *Edge Computing: From Hype to Reality*, 51-69.
29. Pérez, J., Díaz, J., Berrocal, J., López-Viana, R., & González-Prieto, Á. (2022). Edge computing: A grounded theory study. *Computing*, 104(12), 2711-2747.
30. Kumari, K. A., Sadasivam, G. S., Dharani, D., & Niranjanamurthy, M. (2021). *Edge Computing: Fundamentals, Advances and Applications*. CRC Press.
31. Qiu, Tie, et al. "Edge computing in industrial internet of things: Architecture, advances and challenges." *IEEE Communications Surveys & Tutorials* 22.4 (2020): 2462-2488.
32. Chirlawar, V. (2024). Edge computing: revolutionizing data processing and driving Industry 4.0. *Journal of advanced research engineering and technology (JARET)*, 3(1), 25-33.
33. Failure Modes and Effects Analysis (FMEA and FMECA), document IEC60812, International Electrotechnical Commission, 2018.
34. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122

«Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с

35. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

36. Єднак І., Домбровський М. Модуль edge-обчислень для цифрового розвитку інноваційного виробництва. *Інтелектуальні інформаційні технології в прикладних дослідженнях (ІІТАР-2025)*: тези доп. студент. науково-практичної конф. м. Тернопіль, 27–29 травня 2025 р.. Тернопіль: ЗУНУ, 2025. С. 49–51.

Додаток А
Копія опублікованих результатів

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ІТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Дідур Павло, Сапожник Григорій	45
РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ	45
Єднак Іван, Домбровський Михайло	49
МОДУЛЬ EDGE-ОБЧИСЛЕНЬ ДЛЯ ЦИФРОВОГО РОЗВИТКУ ІННОВАЦІЙНОГО ВИРОБНИЦТВА	49
Ковальчук Ярослав	52
СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА РИНКУ СТРАХОВИХ ПОСЛУГ НА ОСНОВІ МАШИННОГО НАВЧАННЯ	52
Крушельницький Святослав	55
ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ НАСТРОЮ ТЕКСТОВИХ ВІДГУКІВ	55
Кутереба Іван	58
ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЯ ДЕФЕКТІВ ФАРБУВАННЯ АВТОМОБІЛІВ НА ОСНОВІ ГЛИБОКОЇ НЕЙРОННОЇ МЕРЕЖІ	58
Лучечко Христина, Сапожник Григорій	62
СЕМАНТИЧНИЙ АНАЛІЗ ТЕКСТОВИХ ДАНИХ ІЗ ВИКОРИСТАННЯМ МОДЕЛЕЙ ВЕКТОРНОГО ПРЕДСТАВЛЕННЯ СЛІВ	62
Мураль Роман, Лендюк Тарас	65
МОДУЛЬ АНАЛІЗУ НАСТРОЇВ У ВІДГУКАХ ПРО ПРОДУКТИ ХАРЧУВАННЯ НА ОСНОВІ WORD2VEC	65
Никоряк Денис, Майків Ігор	68
ВЕБ-ОРІЄНТОВАНА СИСТЕМА ДЛЯ АНАЛІЗУ ТА ОПТИМІЗАЦІЇ МАРШРУТІВ ЗАМОВЛЕНЬ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ WEB WORKER	68
Оберванюк Уляна, Осолінський Олександр	71
КОНЦЕПЦІЯ МОДУЛЯ МОНІТОРИНГУ РОБОТИ ЕЛЕКТРОДВИГУНА З ВИКОРИСТАННЯМ ІОТ	71
Полюшко Денис, Дорош Віталій	74
ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ ІНТЕГРАЦІЇ ТЕХНОЛОГІЙ ГЛИБОКОГО НАВЧАННЯ ТА АНАЛІТИКИ ВЕЛИКИХ ДАНИХ	74
Проць Анастасія	78
НЕЙРОМЕРЕЖЕВА ПРОГРАМНА СИСТЕМА РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ВОГНЮ ТА ДИМУ	78
Созанська Тетяна, Ліп'яніна-Гончаренко Христина	81
МОДУЛЬ КЛАСИФІКАЦІЇ ВІДХОДІВ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ	81

Єднак Іван
студент групи КН-41
taschergames@gmail.com
Домбровський Михайло
к.т.н., ст. викладач
m.dombrovskyi@wunu.edu.ua
Західноукраїнський національний університет
Тернопіль, Україна

МОДУЛЬ EDGE-ОБЧИСЛЕНЬ ДЛЯ ЦИФРОВОГО РОЗВИТКУ ІННОВАЦІЙНОГО ВИРОБНИЦТВА

Завдяки розвитку сучасних інформаційних технологій, інноваційне виробництво прагне вищого ступеня інтелекту, використовуючи можливості недорогого та повсюдного використання сенсорів, передових обчислень та аналітики даних, а також кіберфізичної інтеграції. Конвергенція різних передових та підключених пристроїв і машин є неминучим компонентом інноваційного виробництва. Більше того, зростаюча повсюдність Інтернету речей [1] надає перспективні можливості для створення потужних промислових систем та застосунків [2, 3]. В результаті, великий обсяг різноманітних виробничих даних генерується цими підключеними машинами та сенсорами. Ці дані необхідно очищувати, зберігати та обробляти для створення інформації та знань, які мають стати основою для інноваційного виробництва [4].

Зі швидким зростанням обсягу даних і зростаючими вимогами до обробки інформації, edge-обчислення стали важливим рішенням для цифрового розвитку інноваційного виробництва. Ця інформаційна технологія здійснює обробку даних безпосередньо на кордоні мережі, зокрема на термінальних пристроях, підвищуючи стабільність та надійність послуг. Edge-обчислення зберігають та обробляють дані поблизу джерел інформації, забезпечуючи швидшу, надійнішу та безпечнішу обробку інформації в реальному часі [2, 5].

Переваги edge-обчислень включають зменшення енергоспоживання, зниження витрат і навантаження на мережу порівняно з просто хмарними обчисленнями. Ця технологія допомагає вирішити проблеми з затримками та пропускнуою здатністю, що є критичними запитом в умовах цифрового переходу. Edge-обчислення застосовуються в різних секторах, таких як виробництво, енергетика, розумні будинки та транспорт [2, 4, 5].

На рівні edge-обчислень виробничі пристрої та не тільки виробляють дані, але мають можливість аналізу та обробки даних, що підтримує такий тип обчислення та надає інтелектуальні послуги ближнього рівня. Перш за все, дані про стан (наприклад, звичайна фізична величина, умови навколишнього середовища та експлуатації) та різні параметри виробничих ресурсів та користувачів збираються в режимі реального часу. Тому, першочерговим завданням пристроїв edge-обчислень є перетворення інтерфейсу та протоколу для різних форматів даних та протоколів зв'язку в єдиний протокол. Крім того, для подальшої обробки зібрані дані також піддаються обробці від шуму, очищенню від аномальних даних, за необхідності, стисканню даних. Пристрої edge-обчислень мають певну можливість зберігання даних (може бути невеликою) для зберігання даних про стан у режимі реального часу, а також сигналів тривоги, несправностей та іншої чутливої інформації.

Крім того, мікроцентр обробки даних ефективно використовується для зберігання даних про стан виробничого поля в режимі реального часу. Оскільки обсяг даних великий, пристрої edge-обчислень повинні перезаписувати вихідні дані протягом певного періоду часу новими даними. Щоб заощадити кошти на передачу даних та пропускну здатність, лише необхідні дані завантажуються на верхній рівень інтелектуальної обробки. Крім того, значна частина зібраних даних може не мати великої цінності, лише повідомляючи про «нормальні» умови. Тому перед тим, як дані будуть перезаписані, edge-обчислювальні пристрої будуть фільтрувати, відкидати або більш ефективно повторно збирати дані для передачі в середовище хмарного сервісу для зберігання та аналізу. Отже, для зібраних даних у режимі реального часу edge-обчислювальні пристрої можуть виконувати просту обробку та аналіз даних такої як виявлення порогу або фільтрація за алгоритмом, що подано на рисунку 1. Надсилаючи лише корисні та готові до використання дані, цей підхід мінімізує затримки зв'язку.

Крім того, після простої обробки та аналізу даних уможлиблюється реалізувати автономне керування зворотним зв'язком для роботи виробничого обладнання відповідно до заздалегідь встановлених правил. Продуктивності edge-обчислень є достатньо для забезпечення керування зворотним зв'язком в режимі реального часу. Навіть у випадку офлайн-роботи, периферійні обчислення все ще здатні надавати локалізовані прикладні послуги.

Отже за умов, коли більше ресурсів переміщується з хмари на край мережі, перехід до edge-обчислень забезпечить скорочення часу реакції та кращу надійність функціонування. Фактично, пропускну здатність можна зменшити, якщо значний обсяг даних зберігати безпосередньо на краю, а не завантажувати в хмару.

Коли мілісекунди мають значення, наприклад, в автоматизованому виявленні дефектів на виробничій лінії у режимі реального часу затримки, спричинені передачею даних у хмару та очікуванням відповіді, можуть призвести до втрачених можливостей. Ці затримки не тільки впливають на ефективність, але й можуть мати серйозні наслідки для безпеки, тоді як застосування edge-обчислень безпосередньо

вирішує ці проблеми.

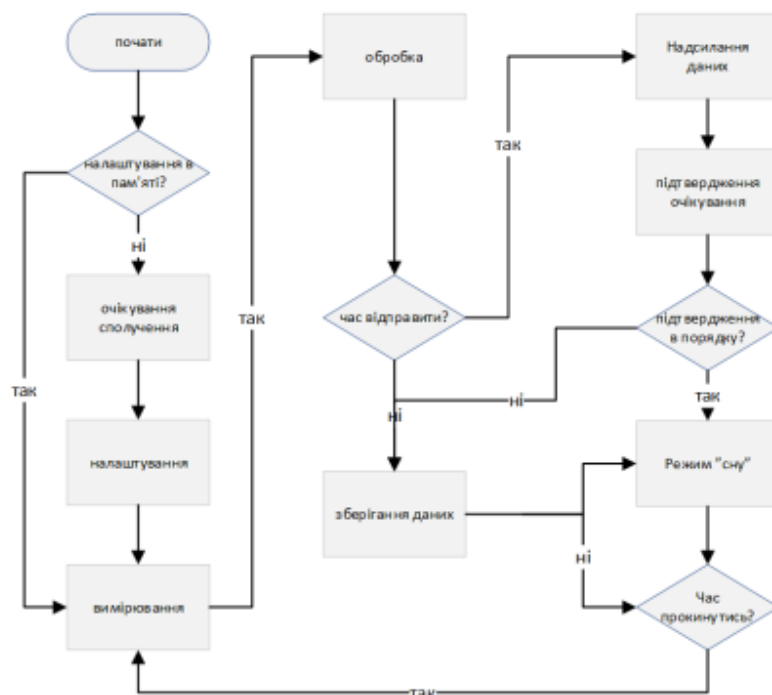


Рисунок 1 - Алгоритм попередньої обробки даних edge-обчислень

Наукове та практичне значення дослідження полягає в тому, що edge-обчислення, туманні обчислення та хмарні обчислення взаємодіють один з одним, краще задовольняючи вимоги програм впровадження інноваційного виробництва.

Список використаних джерел

1. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5), 637-646.
2. Shi, W., Zhang, X., Wang, Y. F., & Zhang, Q. (2019). Edge computing: state-of-the-art and future directions. *Journal of Computer Research and Development*, 56(1), 69-89.
3. Ai, Y., Peng, M., & Zhang, K. (2018). Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks*, 4(2), 77-86.
4. Sunyaev, A., & Sunyaev, A. (2020). Fog and edge computing. *Internet computing: principles of distributed systems and emerging internet-based technologies*, 237-264.
5. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30-39.