МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова праця

на правих рукопису

ДЯЧОК РОМАН ВАСИЛЬОВИЧ

УДК 04.04:004.4:004.93

ДИСЕРТАЦІЯ
МЕТОДИ ТА ЗАСОБИ ІНТЕЛЕКТУАЛІЗАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ З МУЛЬТИСЕНСОРНОЮ КОНФІГУРАЦІЄЮ

123 – Комп’ютерна інженерія

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ / Р. В. Дячок /

Науковий керівник: Клим Галина Іванівна, д.т.н., професор

Львів-2023

АНОТАЦІЯ

Дячок Р.В. Методи та засоби інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія, галузь знань 12 – Інформаційні технології, Національний університет «Львівська політехніка», Міністерство освіти і науки України, Львів, 2023.

Дисертація присвячена розв'язанню актуального науково-технічного завдання розроблення методів та засобів інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією.

У вступі обґрунтовано актуальність теми дисертаційного дослідження, сформульовано мету дослідження та науково-технічні завдання, необхідні для її досягнення, показано зв'язок дослідження з науковими програмами та темами, наведено наукову новизну отриманих результатів, їх практичну цінність та особистий внесок здобувача, надано інформацію про апробацію результатів роботи.

У першому розділі представлено аналіз літературних джерел щодо сучасних підходів із застосування методів та засобів інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією, а саме їх структури, архітектурних рішень та серверного програмного забезпечення. Також розглядаються архітектури мультисенсорного злиття даних, що є одним із ключових завдань при проектуванні мультисенсорної системи. Аналіз показав, що поява систем з великою кількістю сенсорів спричиняє проблеми з передачею та обробкою даних.

У другому розділі представлено методи та засоби обробки інформації в досліджуваних системах, результат виконання яких є оцінюванням стану досліджень. Також приведена розумна архітектура моніторингу фізичних об'єктів, особливості статистичної обробки результатів вимірювання, методи перевірки сигналів в інформаційно-вимірювальних системах.

У третьому розділі представлено результати дослідження щодо вдосконалення

методу динамічного пошук помилок в інформаційно-вимірювальній системі, вдосконалення методу очищення мережевих даних в бездротовому сенсорі на базі технології штучного інтелекту в інформаційно-вимірювально системі. У розділі також представлено результати дослідження методу очищення даних управління талантами в бездротових сенсорних мережах на основі технології інтелекту. Проаналізовано конкретні форми застосування бездротових сенсорних мереж. Представлено характеристики структури бездротових сенсорних мереж та запропоновано технологію очищення даних на основі моделі кластеризації. Представлено алгоритм видалення запису реплікації на основі кластерів та перевірено точність методів очищення даних. Крім цього, викладено теорію динамічного аналізу пошкоджень промислових протоколів інтернету та визначено необхідні дані динамічного мультимодального зв'язку сенсора. Запропоновано метод нечітких тестів у поєднанні з динамічними мультимодальними передачами даних. Приведено результати перевірки методу тестування промислових протоколів.

У четвертому розділі представлено оптимізований метод довіри на основі туману для запобігання втручання третіх сторін при встановлені довірчих відносин між постачальниками сенсорних та хмарних послуг у мультисенсорних системах. Довіра щодо поведінки між вузлами встановлена на рівні бездротових сенсорних мереж, а довіра даних вузлів і об'єктів – у шар туману. Завдяки детальнішому аналізу даних у шарі туману стає можливим відстежувати стан довіри всієї мережі, виявляти атаки на дані та відновлювати вузли неправильної оцінки. В такий спосіб шар туману може бути побудований як надійна третя сторона. Отриманий результат засвідчує, що запропонований нами механізм довіри має певні переваги щодо зменшення споживання енергії, забезпечення довірчого стану граничних вузлів і мережі, а також виявлення деяких прихованих атак на дані та відновлення вузлів з неправильною оцінкою.

П'ятий розділ присвячено практичному застосуванні розроблених методів та засобів інтелектуалізації до ряду інформаційно-вимірювальних систем, зокрема, системи визначення положення тіла людини у віртуальному світі, платформі для

визначення рівня радіаційного забруднення навколишнього середовища, системі управління розумним будинком, а також інформаційно-вимірювальній системі на базі технології SCADA для контролю роботи промислових об’єктів у реальному часі.

Ключові слова: інформаційноінформаційно-вимірювальні системи, інтелектуалізація, бездротові сенсорні мережі, передача даних, мультисенсорні платформи, кіберфізичні системи, Інтернет речей, апаратно-програмне забезпечення, методи тестування, мережеве керування, промислові протоколи інтернету, хмарні сервіси.

ABSTRACT

Dyachok R.V. Methods and means of intellectualization of information and measurement systems with a multi-sensor configuration.

Dissertation for the degree of Doctor of Philosophy in specialty 123 - Computer Engineering, field of knowledge 12 - Information Technology, Lviv Polytechnic National University, Ministry of Education and Science of Ukraine, Lviv, 2023.

The dissertation is devoted to solving the urgent scientific and technical task of developing methods and means of intellectualization of information and measurement systems with a multisensor configuration.

The introduction substantiates the relevance of the topic of the dissertation research, formulates the purpose of the research and the scientific and technical tasks necessary to achieve it, shows the connection of the research with scientific programs and topics, presents the scientific novelty of the results obtained, their practical value and the personal contribution of the applicant, and provides information on the testing of the results.

The first chapter presents an analysis of literature sources on modern approaches to the use of methods and means of intellectualization of information and measurement systems with a multisensor configuration, namely their structure, architectural solutions, and server software. The article also considers the architectures of multisensory data fusion, which is one of the key tasks in the design of a multisensory system. The analysis

shows that the emergence of systems with a large number of sensors causes problems with data transmission and processing.

The second section presents the methods and tools for processing information in the studied systems, the result of which is an assessment of the state of research. It also describes a smart architecture for monitoring physical objects, features of statistical processing of measurement results, and methods for checking signals in information and measurement systems.

The third section presents the results of a study on improving the method of dynamic error detection in an information and measurement system, improving the method of cleaning network data in a wireless sensor based on artificial intelligence technology in an information and measurement system. The section also presents the results of a study of the method of cleaning talent management data in wireless sensor networks based on intelligence technology. The specific forms of application of wireless sensor networks are analyzed. The characteristics of the structure of wireless sensor networks are presented and a data cleaning technology based on a clustering model is proposed. An algorithm for deleting a replication record based on clusters is presented and the accuracy of data cleansing methods is verified. In addition, the theory of dynamic fault analysis of industrial Internet protocols is presented and the necessary data for dynamic multimodal sensor communication is determined. A method of fuzzy tests in combination with dynamic multimodal data transmissions is proposed. The results of testing the method for industrial protocols are presented.

The fourth section presents an optimized fog-based trust method to prevent third-party interference in establishing trust relationships between sensor and cloud service providers in multisensor systems. Trust in the behavior between nodes is established at the wireless sensor network level, and trust in the data of nodes and objects is established in the fog layer. By analyzing the data in the fog layer in more detail, it becomes possible to monitor the trust state of the entire network, detect data attacks, and recover misjudged nodes. In this way, the fog layer can be built as a trusted third party. This result shows that our proposed trust mechanism has certain advantages in terms of reducing energy

consumption, ensuring the trust state of the edge nodes and the network, and detecting some hidden attacks on data and recovering misjudged nodes.

The fifth section is devoted to the practical application of the developed methods and means of intellectualization to a number of information and measurement systems, in particular, a system for determining the position of the human body in the virtual world, a platform for determining the level of radiation pollution in the environment, a smart home management system, as well as an information and measurement system based on SCADA technology for monitoring the operation of industrial facilities in real time.

Keywords: Keywords: information and measurement systems, intellectualization, wireless sensor networks, data transmission, multisensor platforms, cyber-physical systems, Internet of Things, hardware and software, testing methods, network management, industrial Internet protocols, cloud services.

СПИСОК ПРАЦЬ ОПУБЛІКОВАНИХ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

В яких опубліковані основні наукові результати дисертації:

1. Diachok R., Klym H. Data cleaning method in wireless sensor-based on intelligence technology // Measuring Equipment and Metrology, 2022, 83(2), No 2, p. 5-10. (фахове видання України за спеціальністю 123 - Комп'ютерна інженерія). *Особистий внесок – виконання експериментальних досліджень, математичне опрацювання результатів та написання статті.*
https://science.lpnu.ua/uk/istcmtm/vsi-vypusky/vypusk-83-no2-2022/data-cleaning-method-wireless-sensor-based-intelligence

2. Klym H., Diachok R. Dynamic search for errors in industrial internet protocols for application in multisensory control systems // Computer systems and information technologies, 2022, No 3, p. 65-74. (фахове видання України за спеціальністю 123- Комп'ютерна інженерія). *Особистий внесок – виконання експериментальних досліджень, участь в обговоренні основних результатів, математичне опрацювання результатів та написання статті.*
https://csitjournal.khmnu.edu.ua/index.php/csit/article/view/172/105

3. Diachok R., Klym H. Modified fog-based trust method of data monitoring for multi-sensor configuration systems // Measuring Equipment and Metrology, 2022, 83(4), 47-55. (фахове видання України за спеціальністю 123- Комп'ютерна інженерія). *Особистий внесок – виконання експериментальних досліджень, участь в обговоренні одержаних результатів, їх математичне опрацювання та написання статті.*
https://science.lpnu.ua/uk/istcmtm/vsi-vypusky/volume-83-no4-2022/modified-fog-based-trust-method-data-monitoring-multi-sensor

4. Diachok R., Klym H. Current state of development of intelligent information and measuring systems for environmental monitoring with multisensor configuration // Visnyk of Kherson National Technical University, 2022, No 2(81) 55-69.) (фахове видання України за спеціальністю 123- Комп'ютерна інженерія). *Особистий внесок – проведення аналізу літературних джерел закордонних та вітчизняних видань, участь в обговоренні та узагальненні основних результатів, їх опрацювання і написання статті.* http://kntu.net.ua/index.php/eng/content/view/full/85326

5. Трач І.Б., Клим Г.І., Дячок Р.В., Карбовник І.Д. Проєктування мікропроцесорних пристроїв для визначення напрямку до джерела звуку // Військово-технічний збірник. – Випуск, 2022, No 27, с. 35-45. (фахове видання України, технічні науки). *Особистий внесок – участь у виконанні експериментальних досліджень та в обговоренні одержаних результатів, їх математичне опрацювання і написання статті.*

http://vtz.asv.gov.ua/article/view/268041

6. Diachok R., Trach I., Klym H., Karbovnyk I., Dunets R. Hardware and software complex of intellectualized ornithopter-type UAV for military applications // Electronics and information technologies, 2018, Issue 10, p. 31-40. (фахове видання України, технічні науки). *Особистий внесок – участь у виконанні експериментальних досліджень та в обговоренні одержаних результатів, їх математичне опрацювання і написання статті.* http://elit.lnu.edu.ua/pdf/10_3.pdf

Які засвідчують апробацію матеріалів дисертації:

7. Diachok R., Klym H. Monitoring trust status during Fog level data analysis of the sensor network // Proceedings of the 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2022, p. 1-6. (індексуються у Scopus). *Особистий внесок – виконання експериментальних досліджень, участь в обговоренні основних результатів, математичне опрацювання результатів та написання статті.* https://ieeexplore.ieee.org/abstract/document/10018674

8. Diachok R., Klym H., Vasylchyshyn I., Karbovnyk I. Definition system of human body position in virtual reality // Proceedings of the16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2022, p. 358-361. (індексуються у Scopus). *Особистий внесок – участь у виконанні експериментальних досліджень і в обговоренні одержаних результатів, їх математичне опрацювання та написання роботи.*

https://ieeexplore.ieee.org/abstract/document/9766850

9. Klym H., Dunets R., Horbatyi I., Diachok R. Security subsystem and smart home management system // Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 2018, p. 194-197.

(індексуються у Scopus). *Особистий внесок – участь у виконанні експериментальних досліджень та в обговоренні одержаних результатів, їх математичне опрацювання і написання роботи.*
https://ieeexplore.ieee.org/abstract/document/8409126

10. Diachok R., Dunets R., Klym H. System of detection and scanning bar codes from raspberry Pi web camera // Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 2018, p. 184-187. (індексуються у *Scopus*). *Особистий внесок – участь у виконанні експериментальних досліджень та в обговоренні одержаних результатів, їх математичне опрацювання і написання роботи.*
https://ieeexplore.ieee.org/abstract/document/8409124

11. Diachok R., Klym H., Vasylchyshyn I. Real-time mobile-based platform for determining level and location of radiation background. Proceedings of the 22nd International Conference on Computational Problems of Electrical Engineering (CPEE), 2021, p. 1-4. (*індексуються у Scopus та Web of Science). Особистий внесок – виконання експериментальних досліджень, участь в обговоренні основних результатів, математичне опрацювання результатів та написання роботи.*
https://ieeexplore.ieee.org/abstract/document/9585271

12. Дячок Р. В., Клим Г. І. Метод очищення мережевих даних на базі технології інтелекту // Всеукраїнська науково-практична конференція молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості», 13 жовтня 2022, Івано-Франківськ, Україна, С. 209-210. *Особистий внесок – виконання експериментальних досліджень, участь в обговоренні основних результатів, математичне опрацювання результатів та написання роботи.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БСМ - бездротова сенсорна мережа;

КВК - контрольно-вимірювальний комплекс;

ІВС - інформаційно-вимірювальної системи;

СУБД – система управління базами даних;

БД – база даних;

ML - машинне навчання;

AI - штучний інтелект;

WSN - мережа бездротових сенсорів;

КФС – кіберфізична система;

IoT – інтернет речей;

IIoT – промисловий інтернет речей;

ІКТ – інформаційно-комунікаційні технології;

WSAN – бездротова сенсор-актуаторна мережа;

API – прикладний програмний інтерфейс;

BSN – мережа сенсорів;

SSL – протокол захищених сокетів;

SSN - семантична сенсорна мережа;

RAM - оперативна пам’ять;

HMI - інтерфейс людини і машини;

ISTM - інкрементальна просторово-часова модель;

PMF - імовірнісна матриця факторизації.

ВСТУП

Обґрунтування вибору теми дослідження. Стрімкий розвиток інформаційно-вимірювальних технологій та їх широкий спектр впровадження у багатьох сферах призвів до появи складних кіберфізичних систем, у тому числі промислових, розумних будинків, тощо. Такі системи містять значну кількістю сенсорів, які надають дані з різних модальностей для забезпечення надійного сприйняття та розуміння середовища. Завдяки прогресу в апаратних технологіях і технологіях штучного інтелекту, попитом на автоматизацію та автономію, а також появі інноваційних додатків, які потребують інтелектуальних та адаптивних рішень, виникає необхідність у інтелектуалізації мультисенсорних систем, яка передбачає, зокрема, їх інтеграцію у архітектурні рівні Інтернету речей (IoT). Однак збільшення кількості інформаційно-вимірювальних пристроїв, підключених до IoT мережі, призводить до накопичення значної кількості даних, які необхідно оперативно передавати та опрацьовувати, а також ідентифікувати та надавати великі обсяги даних щодо безпеки, цілісності та конфіденційності. Необхідність в обміні такої інформації супроводжується проблемами ефективності їх передачі від нижчого рівня, на якому розміщені сенсори, на вищі рівні туману та хмари і у зворотному напрямку. Крім цього, процеси збору та передачі даних часто супроводжуються одержанням помилкових або суперечливих даних. Такі записи накопичуються, що є недопустимим в базі даних, тому їх необхідно знаходити, очищати та, за потреби, видаляти. Саме тому наявність інтегрованих інформаційно-вимірювальних систем з мультисенсорною конфігурацією у нижньому архітектурному рівні IoT створює необхідність у вдосконаленні та розробленні нових методів і засобів опрацювання інформації, оскільки наявним все важче справлятися з потоками даних, які постійно зростають. Відповідно актуальною є наукова задача інтелектуалізації мультисенсорних інформаційно-вимірювальних систем шляхом удосконалення та розроблення методів та засобів ефективної передачі та валідації інформаційних даних між вимірювальними пристроями і системами та архітектурними рівнями IoT.

Зв'язок роботи з науковими програмами, планами, темами. Дисертація виконувалася відповідно до науковому напряму кафедри «Спеціалізовані комп'ютерні системи»: «Теорія спеціалізованих комп'ютерних систем», відповідно до тематичних планів і науково-дослідних робіт Національного університету «Львівська політехніка», відповідно до Закону України № 433-IV – Про пріоритетні напрями інноваційної діяльності в Україні. Частина дисертаційних досліджень виконана на кафедрі «Спеціалізовані комп'ютерні системи» в межах держбюджетних науково-досліджених робіт Міністерства освіти і науки України для молодих вчених «Наноструктуровані скло-керамічні середовища для високонадійних оптоелектронних та сенсорних застосувань» (№ державної реєстрації 0116U004411) та «Оптимізовані нанокомпозити та сенсорні структури для оборонних систем контролю безпеки та виявлення загроз» (№ державної реєстрації 0116U004411).

Мета і завдання дослідження. Метою дисертації є розроблення та удосконалення методів і засобів передачі та валідації інформаційних даних для інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією в архітектурних рівнях IoT.

Для досягнення поставленої мети необхідне розв'язання таких завдань:

- проаналізувати відомі методи, моделі та засоби інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією в архітектурних рівнях IoT;

на основі проведеного аналізу:

- удосконалити метод валідності промислових протоколів IoT для використання в системах керування з мультисенсорною конфігурацією;
- покращити кодове покриття тестових випадків IoT для використання в мультисенсорних системах керування;
- підвищити ймовірність виявлення аномалій у реалізації протоколу IoT для використання в мультисенсорних системах керування;
- удосконалити метод очищення даних в бездротових сенсорних мережах;

- запропонувати метод видалення реплікованих даних на основі кластерів та перевірити точність методів очищення даних;

- апробувати вдосконалені та розроблені методи і засоби у інформаційно-вимірювальних пристроях і системах.

Об'єктом дослідження є процес інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією в архітектурних рівнях IoT.

Предметом дослідження є методи, засоби передачі та валідації інформаційних даних в інтелектуалізованих інформаційно-вимірювальних системах з мультисенсорною конфігурацією.

Методи дослідження базуються на принципах системного аналізу (ієрархічності, декомпозиції та інше). Для розв'язання поставлених у дисертації завдань використано методи створення інтелектуальних вимірювальних систем, управління талантами, теорії похибок, теорії обчислювальної математики, теорії комп'ютерних систем і мереж, теорії кіберфізичних систем, традиційного моделювання та програмно-математичного забезпечення.

Наукова новизна одержаних результатів:

- *вперше* запропоновано метод нечітких тестів у поєднанні з динамічними мультимодальними передачами даних, який дозволяє відстежувати виконання програми, знаходить поля введення за допомогою динамічного аналізу, збільшуючи здатність виконувати код на глибокий рівень, що дозволило покращити валідність тестових випадків і швидкість покриття коду на 11 %, а також підвищити ймовірність виявлення аномалій у реалізації протоколу;

- *набув подальшого розвитку* метод очищення даних в бездротових сенсорних мережах на основі моделі кластеризації, який дозволив покращити невідповідність при ідентифікації даних із одного і того ж об'єкту на 15 % у порівнянні з методами сортування без попередньої обробки;

- *удосконалено* метод довіри на основі туману, завдяки якому у шарі туману можна відстежувати стан довіри всієї мережі, виявляти атаки на дані та запобігати втручанню третіх сторін у встановлення довірчих відносин між

постачальниками сенсорних і хмарних сервісів у системах з мультисенсорною конфігурацією;

- *набули подальшого розвитку* методи побудови інтелектуальних вимірювальних системи з використанням бази даних та бази знань, а також комунікаційні зв'язки при передачі даних вимірювань та зміни дестабілізуючих факторів, що в кінцевому результаті дало змогу зменшити навантаження на рівні туман/хмара.

Практичне значення одержаних результатів.

Практичне значення дисертаційного дослідження полягає у розробленні та удосконаленні методів та засобів передачі інформаційних даних в широкому спектрі інформаційно-вимірювальних систем з мультисенсорною конфігурацією, зокрема; метод очищення мережевих даних, який дозволяє вирішити проблему некоректної ідентифікації даних і значно покращити тестування промислових інтернет-протоколів, валідність тестів, швидкість покриття кодів і знижує ймовірність аномалій у реалізації протоколів; модифікований метод довіри на основі туману для забезпечення довірчого стану граничних вузлів і мережі, виявлення деяких прихованих атак на дані та відновлення вузлів з неправильною оцінкою. Запропоновані та удосконалені методи та засоби передачі даних застосовані в розроблених системах визначення положення тіла людини у віртуальному світі, рівня радіаційного забруднення навколишнього середовища, управління розумним будинком, а також інформаційно-вимірювальній системі на базі технології SCADA для контролю роботи промислових об'єктів в реальному часі.

Одержані результати досліджень дисертації також впроваджено у навчальний процес для викладання дисциплін «Дослідження і проектування програмних систем» та «Дослідження і проектування спеціалізованих комп'ютерних систем» для студентів освітньо-кваліфікаційного рівня «магістр», що навчаються за спеціальністю 123 «Комп'ютерна інженерія, спеціалізацію 123.03. «Спеціалізовані комп'ютерні системи», у науково-дослідну роботу для молодих вчених «Оптимізовані нанокомпозити та сенсорні структури для оборонних систем контролю безпеки та виявлення загроз» (№ держ. реєстру 0122U000807) та

Державне підприємство Науково-телекомунікаційний центр «Українська академічна і дослідницька мережа» Інституту фізики конденсованих систем НАН України.

Особистий внесок здобувача. Основний зміст роботи, всі теоретичні та практичні результати, дослідження і висновки, які представлено до захисту, одержані автором особисто. Особисто здобувачеві належать наступні наукові результати: метод нечітких тестів у поєднанні з динамічною мультимодальною передачею даними; удосконалений метод очищення даних в бездротових сенсорних мережах на основі моделі кластеризації та метод довіри на основі туману; методи побудови інтелектуальних вимірювальних системи з використанням бази даних та бази знань, а також комунікаційні зв'язки при передачі даних вимірювань та зміні дестабілізуючих факторів; апробація запропонованих методів та засобів інтелектуалізації у ряді спроектованих інформаційно-вимірювальних пристроїв та систем, зокрема, системі для визначення положення тіла людини у віртуальному світі, мініатюрному безпілотному орнітоптері, платформі для визначення рівня та локації радіаційного фону, системі управління розумним будинком, системі для визначення напрямку до джерела звуку. Особистий внесок здобувача у колективно опублікованих працях: досліджено метод очищення даних в бездротових сенсорних мережах [1]; запропоновано метод нечітких тестів у поєднанні з динамічною мультимодальною передачею даними [2]; запропоновано оптимізований метод довіри на основі туману [3,7,12]; проаналізовано сучасний стан та перспективи застосування мультисенсорних пристроїв та систем [4]; апробовано розроблені методи у ряді інформаційно-вимірювальних систем [5,6,8-11], спроектовано пристрій для визначення напрямку на джерело звуку [5]; запропоновано рішення щодо створення мініатюрного безпілотного орнітоптера [6]; проаналізовано переваги та недоліки блокчейну та Fog обчислень на основі ICPS [7]; спроектовано та досліджено систему для визначення положення тіла людини у віртуальному світі [8]; оптимізовано параметри системи для точного вимірювання, контролю та

керування в режимі реального часу [9,10]; запропоновано та досліджено мобільну платформу для визначення рівня та локації радіаційного фону [11]. У всіх публікаціях здобувач самостійно проводив експерименти, здійснював математичне моделювання та описував одержані результати. Ідеї, положення чи гіпотези інших авторів, які наявні в дисертації, мають відповідні посилання і використані лише для підсилення ідей та результатів.

Апробація результатів дисертації. Основні теоретичні положення та практичні результати дисертації доповідалися і обговорювались на семінарах та конференціях: наукових семінарах кафедри спеціалізовані комп'ютерні системи Національного університету «Львівська політехніка» (2018-2023 рр.), Всеукраїнській науково-практичній конференції молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості» (Івано-Франківськ, Україна, 2022 р.), Міжнародних конференціях 12th International Conference on Dependable Systems, Services and Technologies DESSERT (Athens, Greece, 2022), 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering TCSET (Lviv, Ukraine, 2021), 22nd International Conference on Computational Problems of Electrical Engineering CPEE, (Czech Republic, 2021), 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), (Kiev, Ukraine, 2018).

Публікації. За матеріалами дисертації опубліковано 12 наукових праць, з них: 6 статей у наукових фахових виданнях України та 6 публікацій у матеріалах та збірниках доповідей наукових конференцій, з яких 5 індексуються у наукометричній базі даних Scopus.

Структура та обсяг роботи. Дисертаційна робота складається із вступу, п'ятьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг основного тексту складає 167 сторінок, 57 рисунків, 8 таблиць, список використаних джерел з 214 найменувань на 21 сторінці, додатки на 13 сторінках.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ПЕРСПЕКТИВ ЗАСТОСУВАННЯ МУЛЬТИСЕНСОРНИХ СИСТЕМ

В розділі представлено аналіз літературних джерел із застосування методів та засобів інтелектуалізації інформаційно-вимірювальних систем з мультисенсорною конфігурацією. Інформаційно-вимірювальні системи на сьогоднішній день виконують функції контролю та координації в будинку – керування освітленням, утепленням, безпекою, звуком, переміщенням жалюзей та інших речей. У цій ситуації розумний дім нагадує екосистему, якою керує центральний «мозок», а управління здійснюється за допомогою смартфону. Для контролю параметрів сприяння росту рослин у агропромисловому комплексі (теплицях), а саме контролем за вологістю та температурою повітря, температурою та вологістю грунту, вмістом у повітрі вуглекислого газу (CO_2) та інших супутніх газів за інтенсивність освітленням. У випадку контролю за безпекою будівлі, тобто виявлення пожежі, несанкціонованого проникнення, широко застосовуються технології, засновані на соціальних мережах. В огляді також розглядаються архітектури мультисенсорного злиття даних, як одного із ключових завдань при проектуванні мультисенсорної системи. Поява систем з великою кількістю сенсорів, наприклад Інтернет речей, може внести новизну в цю добре вивчену тему.

1.1. Аналіз поширених мультисенсорних інформаційно-вимірювальних систем

На сьогоднішній день існує необхідність у постійному моніторингу параметрів життєвого середовища в режимі реального часу. Зважаючи на це, нині використовуються недорогі системи, які відстежують основні параметри якості життєвого середовища в будівлях за допомогою IoT (див. рис. 1.1.). Кожен пристрій таких систем має унікальну ідентифікацію і можливість автономного збору даних у режимі реального часу [1]. Основні конструктивні блоки IoT складаються з сенсорів, процесорів, шлюзів та додатків.

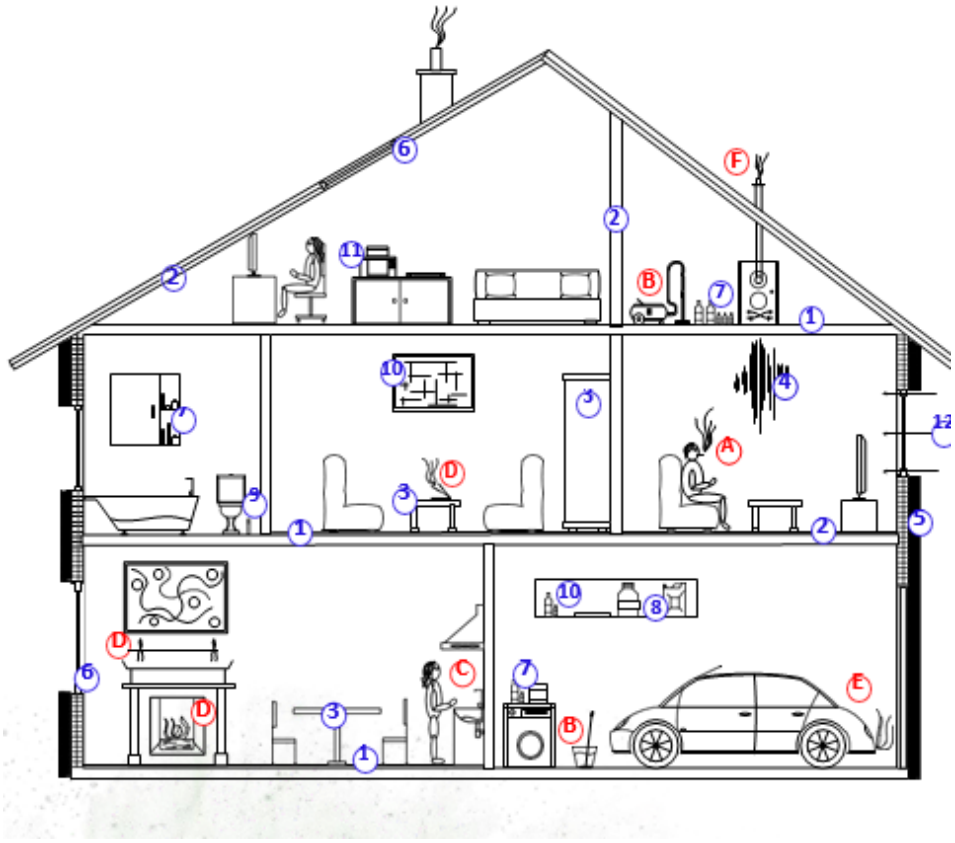


Рис.1.1. Типове представлення розумної будівлі [1]

Мережа бездротових сенсорів (WSN) та технології бездротового зв'язку все більше розвиваються для надання допомоги в особистих та професійних щоденних потребах людини. Програми бездротових технологій були розроблені для збору даних, контролю будівель, систем моніторингу навколишнього середовища та автоматизації виробничих процесів за останні роки [1]. Сучасні бездротові мережі мають безперечні переваги, зокрема низькі витрати як на встановлення, так і на обслуговування, а також великий час безперебійної роботи. Мережа віддалених сенсорів може використовуватися для стаціонарних або мобільних сенсорних мереж, а саме обстеження розвитку міської інфраструктури, моніторингу в екології, телемедицині чи дистанційному медичному обслуговуванні, спостереженні у сільському господарстві, зокрема нагляд за рибальством, фермерством, безпекою кордонів, управління дорожнім рухом, управління лісовим господарством та запобігання катастрофам, тощо. WSN складається з компактно розподілених вузлів сенсорів для зондування, обробки сигналів, вбудованих обчислень та підключення [2]. Така система забезпечує взаємодію між людьми чи комп'ютерами та

навколишнім середовищем за допомогою бездротового зв'язку. Хоча спочатку WSN використовувались у військових та промислових програмах, сучасні WSN-програми використовуються для різних цілей - від простих до складних промислових систем [3].

Система WSN дозволяє користувачам контролювати та керувати підключеними пристроями від базової станції за допомогою різних стандартів бездротового зв'язку, таких як Wi-Fi, General Packet Radio Service (GPRS), Bluetooth, ZigBee, ідентифікація радіочастот (RFID) та стільникові технології. Користувачі можуть відстежувати дані через бездротову мережу, яка може бути розроблена на основі одного із цих стандартів бездротового зв'язку. Перевагами WSN є низьке енергоспоживання, отримання значного обсягу даних, віддалений моніторинг, швидке встановлення мережі, широка зона покриття, висока точність моніторингу та низький робочий цикл. WSN у реальному світі практично не обмежується фізичною безпекою, моніторингом навколишнього середовища та зміною клімату.

Інтернет речей був розроблений паралельно WSN і є фізичною мережею, яка з'єднує всі речі для обміну даними та інформацією через пристрої зондування даних, такі як сенсори, виконавчі механізми та комп'ютери відповідно до протоколів [4]. Особливо важливі ці аспекти у випадку систем моніторингу.

1.1.1. Аналіз сенсорних систем для моніторингу

Багато речей пов'язано в мережі в тій чи іншій формі. Цілі інтелектуального виявлення, моніторингу, пошуку, відстеження та контролю речей досягаються IoT. Існує безліч програм IoT, таких як теги RFID, сенсорні технології, мобільні технології та інші інтелектуальні технології [5]. В роботах [6,7] запропоновано ефективну систему моніторингу якості води на основі бездротової сенсорної мережі (WSN), яка аналізує якість води для зрошення. Karthik Kumar та співавтори [8] досліджували підводну бездротову сенсорну мережу, що живиться від сонячної панелі, для моніторингу якості води. Через WSN різні дані, такі як *p*H, мутність, вміст кисню, зібрані певними сенсорами вузла, надсилаються на базову станцію. Дальше зібрані дані відображаються візуально та аналізуються за допомогою

різних інструментів моделювання. Marco Zennaro et al. [9,10] запропонували проєкт системи контролю якості води та, спираючись на технологію Sunspot, прототип реалізації бездротової сенсорної мережі (WQWSN) як рішення для проблеми моніторингу якості води. Patil [11] також досліджував бездротову сенсорну мережу для системи моніторингу водного середовища, яка забезпечує он-лайн моніторинг температури, каламутності, рівня води та солоності.

A.C.Khetre, S.G.Hate запропонували систему моніторингу якості води в режимі реального часу за допомогою IoT [12]. Звичайний метод тестування якості води полягає у збиранні зразків вручну та відправленні до лабораторії для тестування та аналізу. Цей метод трудомісткий та неекономічний. Запропонована система перевіряє якість води за допомогою різних сенсорів (по одному для кожного параметра: *p*H, провідності, температури). Модуль ZigBee в системі передає дані, зібрані сенсорами, на мікроконтролер бездротовим способом, а модуль GSM передає бездротові дані далі від мікроконтролера на смартфон / ПК. Система також має сенсори наближення, які попереджають про забруднення, надсилаючи повідомлення через модуль GSM на випадок, якщо хтось намагається забруднити водойму.

Brinda Das та інші запропонували розумне управління водопровідними мережами за допомогою технології LPWAN IoT [13]. Нова технологія широкосмугової мережі низької потужності (LPWAN) під назвою LoRa була використана для зв'язку пристроїв IoT. Пристрої LoRa можуть обмінюватися даними в межах 2-4 км, працюючи від акумуляторів. Запропонована система управління водопровідними мережами включає сенсори, розгорнуті в різних стратегічно обраних місцях для вимірювання якості води шляхом генерування даних у реальному часі. Система також забезпечує механізм оповіщення, який повідомляє про проблеми за допомогою електронної пошти та SMS. Сенсори з мікроконтролером в модулі LoRa, будуть зв’язуватися із хмарним середовищем через шлюз LoRa. Веб-сторінка забезпечує інтерфейс для оцінки якості води після аналізу даних за допомогою алгоритму прогнозування.

M. Saravanan, A. Das, V. Iyer, представили систему контролю забрудненням за допомогою IoT [14]. У запропонованій системі використовується сенсор ультрафіолетового світла, який видає аналоговий сигнал про кількість вичавленого УФ-світла, а також сенсор температури та *p*H води для моніторингу якості води, температури, мутності, тощо. Зібрані параметри надсилаються у хмару.

Anekwong Yoddumnern та ін., [15] запропонували систему IoT для моніторингу якості води в аквакультурі, яка контролює якість води з допомогою бездротових сенсорних мереж та IoT для оптимізації ресурсів догляду за ставком. Cesar Encinasn з співавторами та Jozsef Konyha і інші [16,17] запропонували систему вимірювання якості води для широких територій на основі сітки. Вона являє собою прототип простої в установці технології, за допомогою якої можна вимірювати різні показники якості поверхневих вод. Завдяки модульній конструкції сенсора-трубки можна здійснювати вимірювання від 1 до 7 показників. Широкообласна система вимірювання (WAMS) здійснює обмін інформацією через мережу GPRS. Pradeep Kumar та інші [18,19] запропонували моніторинг якості води за допомогою RF-модуля в режимі реального часу. У системі задіяні сенсор рівня *p*H, сенсор температури та сенсор мутності (LED-LDR-монтаж). Виміряні показники передаються на віддалену базову станцію за допомогою РЧ-модуля (2,4 ГГц), що робить зручним моніторинг у віддаленому форматі.

Raja Vara Prasad та інші, [20] запропонували систему, яка інтегрувала такі технології як перескокову частоти та віртуальні прилади для виконання бездротової передачі даних для моніторингу якості повітря. Несуча частота регулюється відповідно до результату і використовується повний радіочастотний спектр. Бездротова передача даних здійснюється без втручання в дослідний зразок в режимі реального часу. Отримані дані легко читаються та чітко відображаються.

Devarakonda S. та інші, [21] запропонували систему для контролю забруднення повітря в режимі реального часу за допомогою WSN. Калібрування газових сенсорів CO_2 та NO_2, здійснюється за допомогою технологій калібрування, а потім формується WSN за допомогою алгоритму агрегування даних із кількома стрибками. Дані про забруднення відображаються у вигляді цифр і діаграм за

допомогою веб-інтерфейсу, а також доступні в Інтернеті. Параметри температури та вологості вимірюються разом з газами, а дані аналізуються даними синтезу.

В роботах [22,23] запропоновано систему аналізу параметрів води для промислового застосування з використанням IoT. У систему адаптовані технології IoT, WSN та стандарти зв'язку. IoT та WSN використані для збору даних про фізичні речі за допомогою стандартного протоколу зв'язку. Аналіз продуктивності запропонованої системи проводиться шляхом збору даних параметрів води від різних чутливих сенсорних елементів (каламутність, густину, температура та *p*H) на базовій станції та порівняння з її пороговим значенням на блоці моніторингу.

Kim J.Y., та інші [24] представили бездротові сенсорні рішення для моніторингу навколишнього середовища. У цьому дослідженні запропонована архітектура бездротової сенсорної мережі, яка поєднує зондуючі вузли з кількома параметрами для надійного моніторингу параметрів якості води поверхневих вод. Особлива увага приділяється проектуванню схем кондиціонування сигналів провідності, температури та помутніння, висвітлюючи важливі питання, пов’язані з лінеаризацією, вимірюванням динамічного діапазону та впровадженням комерційних компонентів та пристроїв. Silvani X. та інші [25] запропонували розумну систему моніторингу якості води для на базі IoT та технології дистанційного зондування. У запропонованій Jelicic, V., R. Ramya [26] системі моніторинг якості води в реальному часі здійснюється за показниками температура, *p*H, мутність, провідность. Основним контролером для крайових обчислень використано Raspberry Pi B+. Отримані дані передаються у хмару. Sandeep Kumar Polu [27] запропонував бездротову систему закупівлі для моніторингу якості води. Запропонована система орієнтована на розробку бездротової системи збору, яка є основним елементом системи моніторингу якості води, та дистанційно вимірює каламутність і *p*H води. Система побудована з використанням мікроконтролера Pericheral Interface Controller (PIC), складається з двох секцій, а саме: секції передавача, яка збирає показники *p*H та помутніння з віддаленого місця, та секції приймача, яка збирає передані показники за допомогою протоколу бездротового зв'язку ZigBee. Результати класифікуються за трьома

класами з використанням різних рівнів *p*H та мутності для отримання індексу якості води. Результати відображаються на РК-дисплеї, а також на ПК впродовж різних періодів часу.

У всіх проаналізованих системах сенсори для вимірювання якості повітря та води, такі як *p*H, температура, мутність, сенсори CO_2 та сенсори MQ підключені до блоку мікроконтролера для подальшої обробки. Блок послідовного зв'язку діє як фаза між MCU та модулем GPRS, модуль GPRS передає дані на робочу станцію, а згодом дані зберігаються у хмарі для подальшого використання. Більшість використовуваних сенсорів дадуть аналоговий вихід АЦП, присутній у контролері, передаються виміряні дані за допомогою модуля GPRS, підключеного до мікроконтролера за допомогою протоколу UART.

Подібні системи при зміні конфігурації сенсорів можуть застосовуватися для біомедичної діагностики, у побуті та виробництві, тощо [28-30]. Для прикладу, система біомедичної діагностики може бути використана для загального управління стану здоров’я організму як у виробничому, так і повсякденному середовищі. В подібних системах зібраний та перетворений мікроелектричний сигнал від сенсорів спочатку відновлюється в результаті складного обчислювального процесу. Далі інформація передається за участі штучної нейронної мережі (ANN) яка підключена до Інтернету через вбудовану функцію зв'язку.

1.1.2. Структури інформаційно-вимірювальних систем моніторингу в концепції IoT

Типова інформаційно-вимірювальна система для моніторингу тих чи інших параметрів на базі IoT складається з набору сенсорів, інтегральної схеми зчитування (ROIC), модулів управління та зв'язку, таких як блок мікроконтролера (MCU) та інтерфейс Bluetooth, який може взаємодіяти зі смартфоном і застосовуватися до різних додатків за допомогою функції зв'язку, приведено на рис. 1.2 [28].

Таким чином, смартфон або ноутбук ілюструє зручний обмін інформацію за допомогою модуля зв'язку. Кожен сенсор перетворює зміну електричних сигналів

як напругу, струм, опір та ємність. У випадку з напругою та струмом, високоефективний аналого-цифровий перетворювач (АЦП) може перетворити на цифровий код, який використовується для управління та зв'язку для MCU. Для обробки інформації про опір та ємність, ROIC складається з інтерфейсу та АЦП, що включає отримання, обробку та перетворення сигналів сенсорів. Отже, специфічна ROIC функціонально необхідна для обробки змін сенсора в системі. Реконфігурований ROIC має переваги, які підтримують сумісність IoT та заощаджують витрати на виготовлення ROIC.

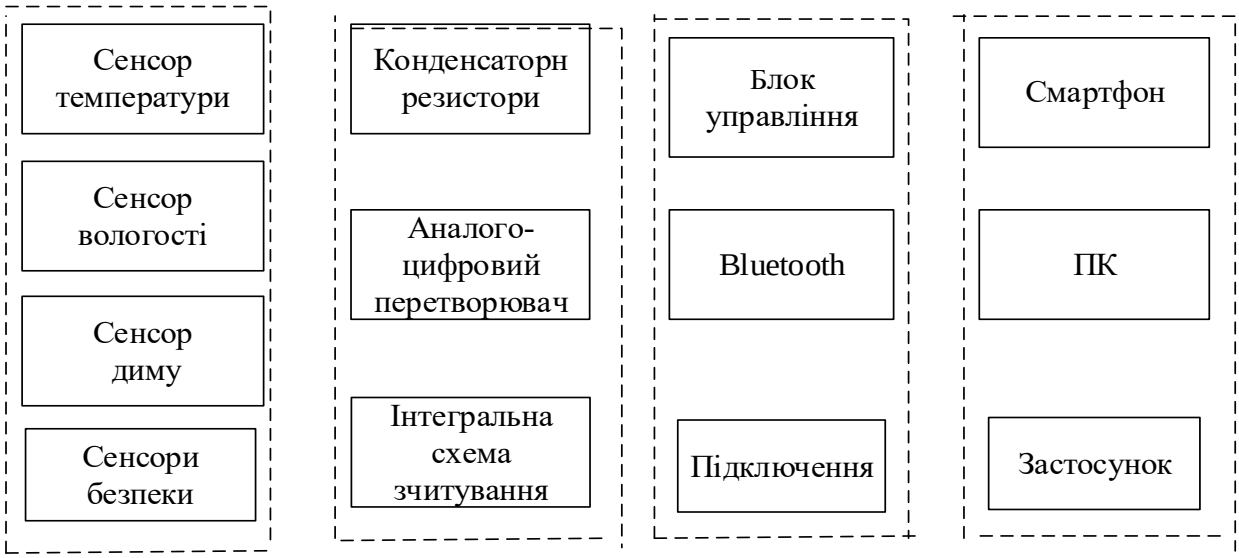


Рис. 1.2. Типова конфігурація сенсорної системи з прикладними пристроями [28]

В найтиповіших системах моніторингу для того, щоб відповідати промисловому попиту, в якості апаратної платформи для обробки даних декількох сенсорів використовують плату Arduino. Це платформа з мікроконтролером та відкритим кодом, що дозволяє приєднувати різноманітні сенсори та пристрої [29]. Для формування вузлів сенсора обрана плата Arduino Uno на основі Wi-Fi. Крім того, розроблена локальна база даних для розміщення та управління даними безлічі сенсорів, а технологія великих даних через Apache Spark реалізована для організації отриманих чисельних даних та підтримки подальшого злиття та аналізу даних. Завдяки характеристиці бездротової сенсорної мережі, систему можна легко

розширити шляхом додавання більшої кількості сенсорів. Запропонована структура системи моніторингу зображена на рис. 1.3.

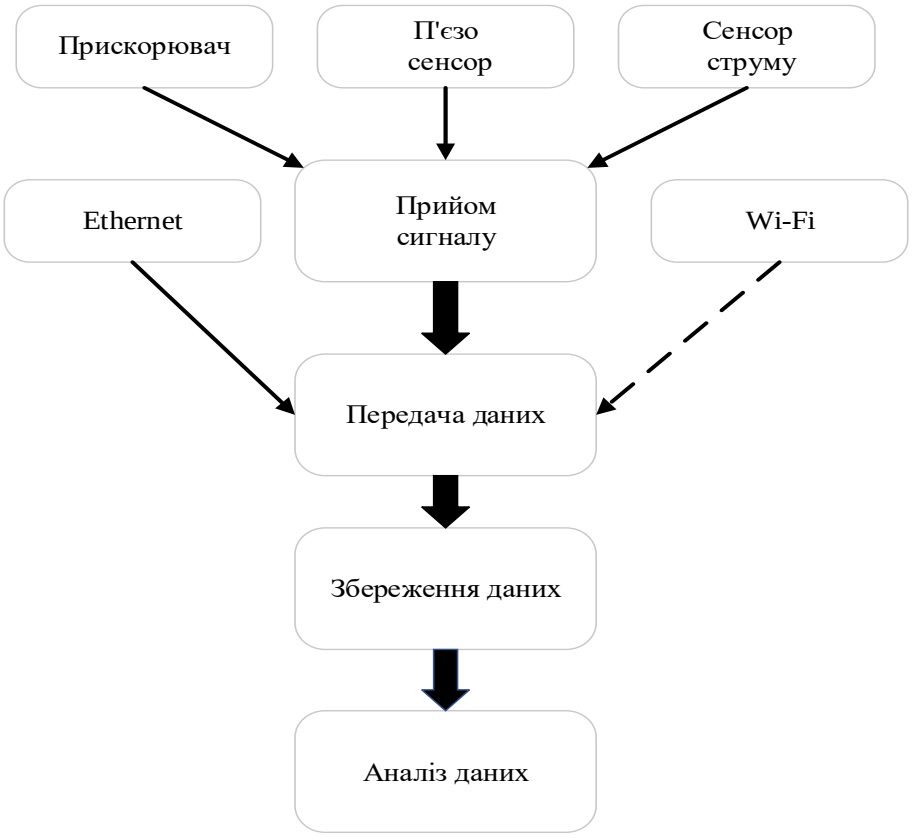


Рис. 1.3. Система моніторингу та аналізу даних

Важливим компонентом у будь-яких інтелектуальних інформаційно-вимірювальних системах є пристрій з мультисенсорним елементом Wi-Fi [30]. Варто відзначити, що мультисенсори повинні відстежувати дані, інформувати про стан середовища (навколо будинку, тощо) і підтримувати будь-які вимоги в умовах цифрового суспільства. Значна кількість робіт присвячена проектуванні розумних будинків, важливим аспектом у яких є мультисенсорна конфігурація [31]. Зокрема, важливим моментом є спостереження за безпекою будинку, виявлення пожежі із застосуванням декількох технологій, заснованих на соціальних мережах. Відомо, що калібрування мультисенсора використовує час калібрування, а самокалібрування - фільтр імпульсної характеристики Fi-nite (FIR). Далі повномасштабний фільтр Калмана (FSKF) допомагає заповнити дані та оцінити точність. Після цього механізм виявлення пожежі використовує нечітку логіку для виявлення та надсилання попереджувальних повідомлень упродовж періоду дії IF

This Then That (IFTTT). Домашня подія змінює дані мультисенсорного Wi-Fi. Було взято ефект діапазону даних до пропорції пожежі всередині будинку. Мультисенсорний вузол Wi-Fi повинен використовувати більше одного детектора, що буде мати високу стабільність і високу точність. Розроблені методології та пристрої, здатні виявляти задимленість та пожежі. Вони надсилають попереджувальне повідомлення на смартфон через соціальну мережу після повідомлення Facebook, Gmail або Line.

Така система спостереження комплектується трьома сенсорами для вимірювання температури навколишнього середовища, відносної вологості та виявлення пожежі на основі соціальної мережі для перевірки вмісту чадного газу в повітрі з використанням розумного мультисенсорного вузла Wi-Fi. У запропонованій системі для прогнозування ймовірності пожежі також використано елементи нечіткої логіки при опрацюванні отриманих значень від сенсорів. Обмін даними здійснюється через хмару (рис. 1.4) [32].

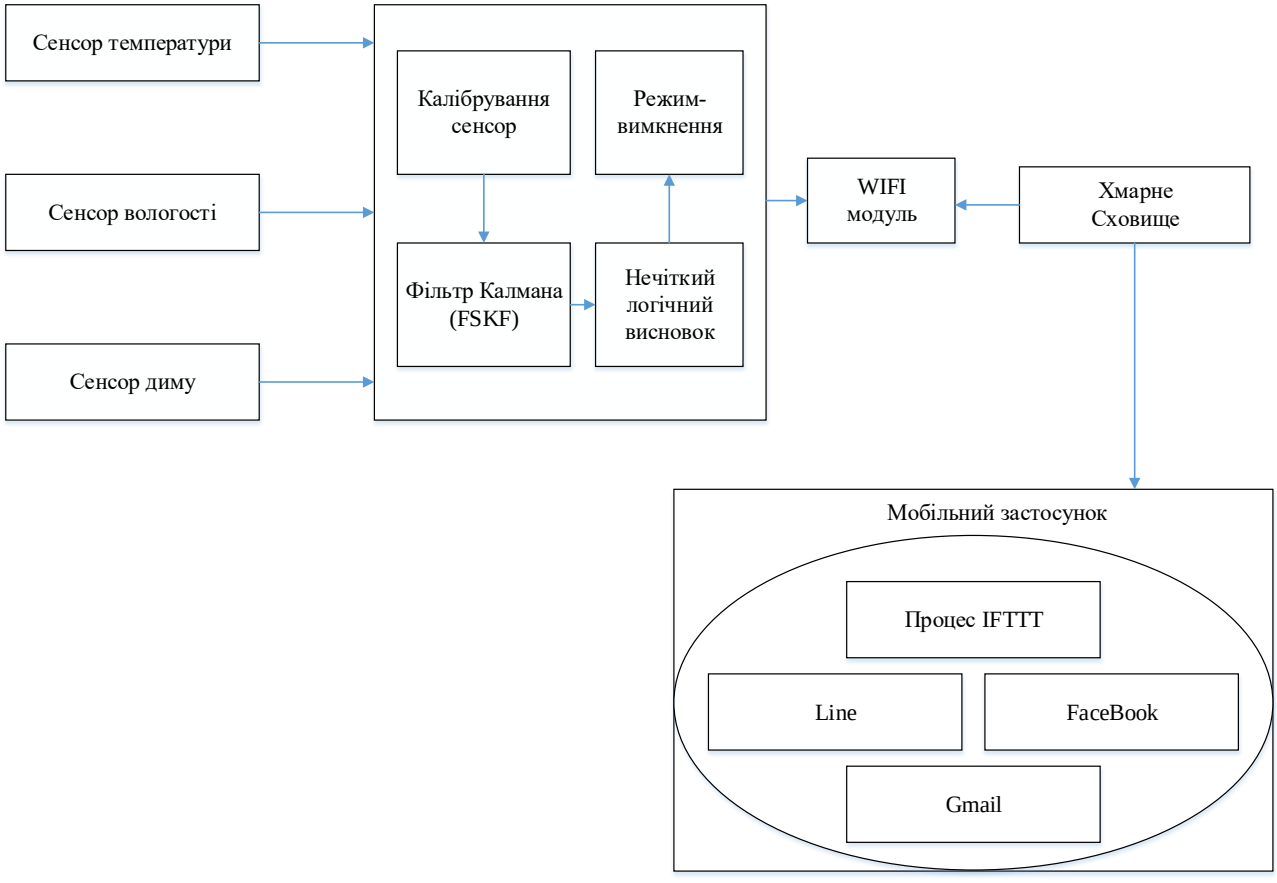


Рис. 1.4. Система спостереження мікроклімату з надсиланням сповіщень через мобільні додатки

Крім цього, використовується обробка IFTTT для перевірки та отримання повідомлення чи попередження через соціальну мережу на смартфон. Щодо платформ, на яких реалізуються такі системи, то зазвичай використовуються плати Arduino, ESP32, Raspberry Pi, ESPresso lite або NodeMCU. Розробити та впровадити систему моніторингу довкілля є складним завданням, оскільки необхідно враховувати управління живленням, вибір сенсорів та тип мережі.

У роботі [32] переносні сенсорні вузли інтегровані з існуючою бездротовою мережевою інфраструктурою сенсорів (рис. 1.5). Існуюча мережа WSN використовується для маршрутизації даних з вузлів на хмарний сервер. Система управління живленням включає акумуляторну батарею, перетворювач підвищення напруги і чотириканальні перемикачі SPST (ADG811). Перемикачі використовуються для управління вмиканням та вимиканням сенсорів відповідно до різного споживання енергії та додатків. Для управління включенням і вимиканням сенсорів використовуються лише два перемикачі. XBee та MCU налаштовані на одночасне вмикання та вимикання. MCU ATmega збирає дані з кожного сенсора, а потім надсилає дані на базову станцію через радіочастотну лінію зв'язку. Він також контролює споживання енергії.

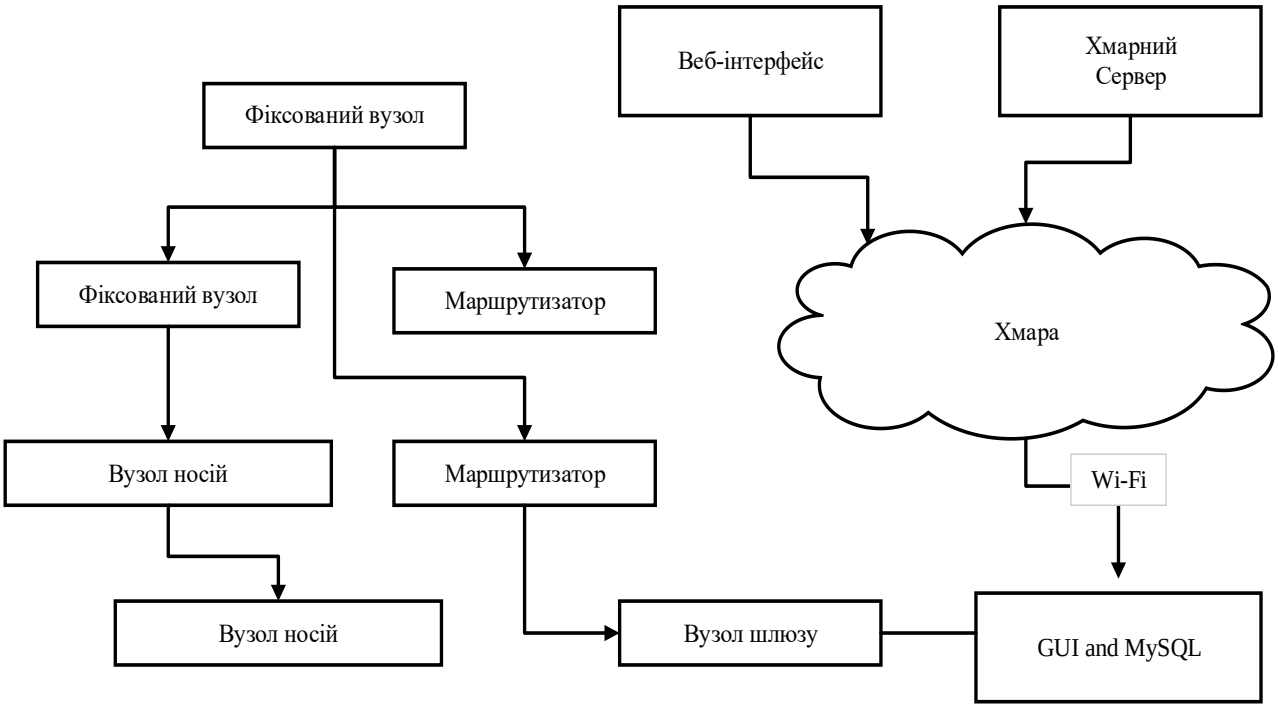


Рис. 1.5. Архітектура та реалізація системи з бездротовою мережевою інфраструктурою сенсорів

Модуль XBee-Pro 900HP використовує мережевий протокол DigiMesh. Мережа дозволяє кожному вузлу спати і прокидатися одночасно. Кожен вузол у мережі є одноранговим і не вимагає наявності додаткових маршрутизаторів у мережі, які не можуть перебувати в режимі сну.

Дані про температуру, вологість та тиск вимірюються за допомогою сенсора BME280, а дані про рівень CO_2 оцінює сенсора COZIR CO_2. Сенсор світла TSL2591 може вимірювати рівень LUX при споживанні низької потужності. Для вузла сенсора фіксованого розташування кожен вузол перебуває в режимі сну, а потім прокидається для вимірювання. Під час пробудження MCU вимірює температуру, вологість, тиск, рівень освітленості та значення CO_2, а потім пакетує дані. XBee перевіряє, чи доступний радіочастотний канал для передачі. Якщо так, дані будуть надіслані на адресу призначення. Після успішної передачі даних весь вузол сенсора повернеться в режим сну [32].

Такі переносні вузли сенсорів за функціональністю схожі на фіксовані вузли, за винятком того, що їм потрібно пробуджуватися частіше. Також їх можна запрограмувати для роботи в режимі постійного моніторингу, який буде оновлювати дані відповідно до вимог користувачів. Детальний програмний алгоритм для переносного вузла сенсора представлений на рис. 1.6 [33]. Базова станція отримує дані як від фіксованих вузлів сенсорів, так і від носіїв сенсорів. Він буде відображати дані в локальному графічному інтерфейсі користувача (GUI) і зберігати дані в локальній базі даних MySQL. Дані в кінцевому підсумку будуть передані на хмарний сервер через Ethernet [34]. У цьому випадку топологія мережі описується типом *mesh + cluster*. Вузли сенсорів фіксованого розташування мають сітчасте з'єднання. Вони періодично прокидаються для надсилання даних. Існує також декілька фіксованих вузлів маршрутизатора для підтримки зв'язку сенсорних вузлів, як показано на рис.1.6 [34].

У роботі [35] представлена розроблена автоматизована бездротова система моніторингу клімату в теплицях з особливим акцентом на аспекти програмування та тестування сенсора температури та вологості. Запропонована система включає

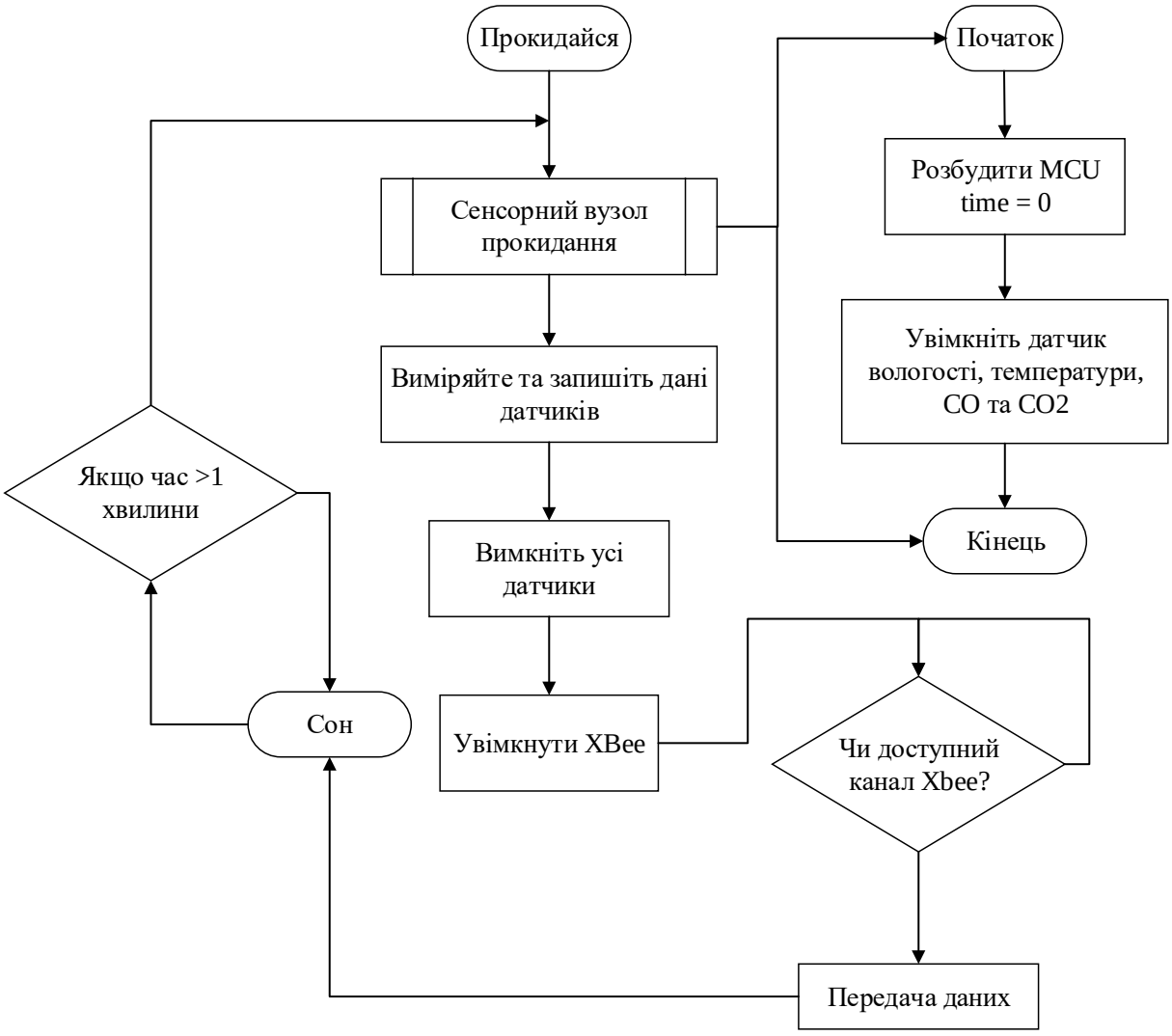


Рис. 1.6. Програмний алгоритм переносного вузла сенсора

три блоки: сенсорну станцію (СС), станцію координатора (СС) та центральну станцію управління (ССУ). Магістраль бездротової мережі базується на модулях ZigBee для зв'язку між SS і CS, тоді як для зв'язку між CS і CCS використовується власний RF-модем XStream. Проведені польові випробування встановили функціональність та надійність спроектованої бездротової сенсорної мережі. Система може контролювати до шести параметрів навколишнього середовища, а саме: атмосферну температуру, вологість, вуглекислий газ (CO_2), інтенсивність світла, вологість та температуру ґрунту (рис. 1.7).

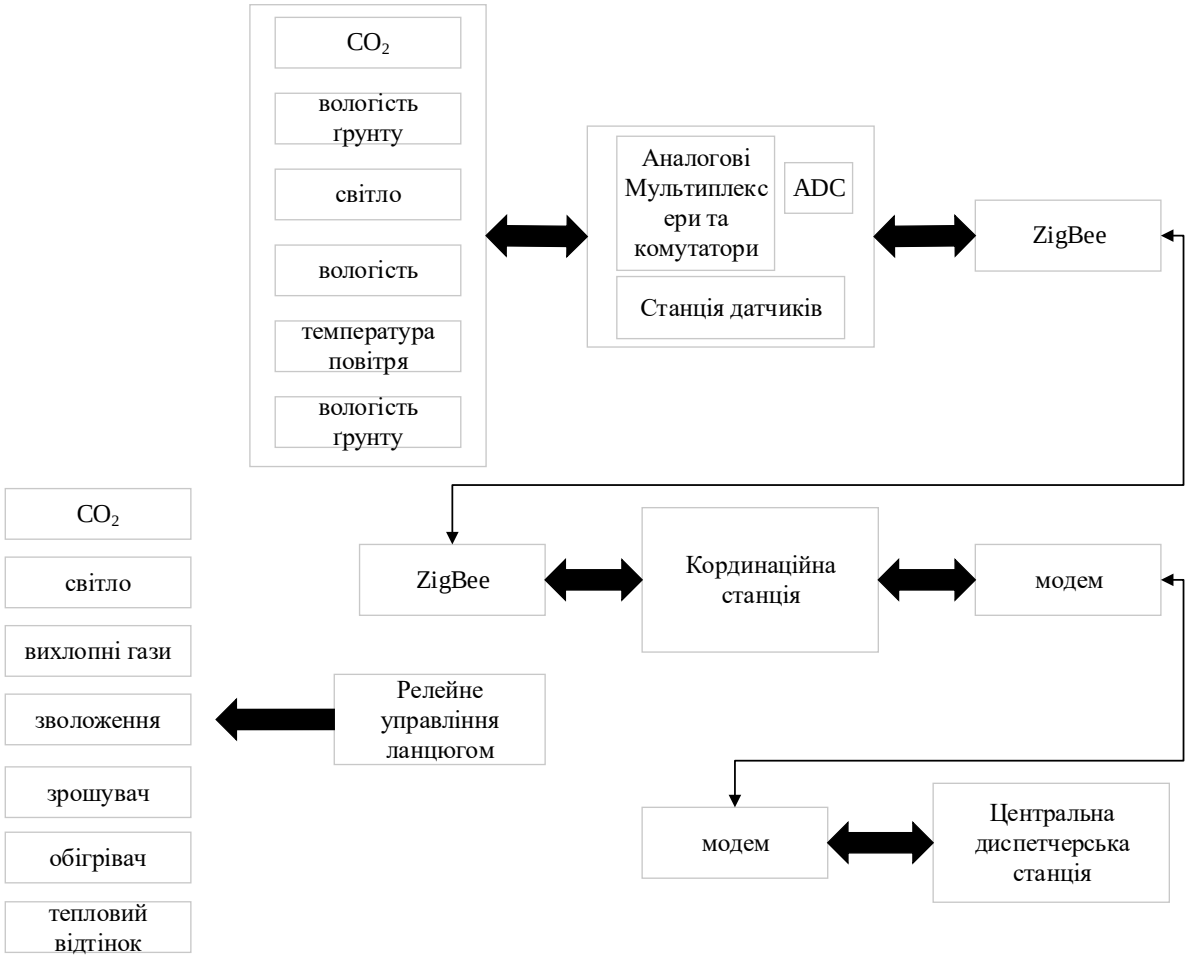


Рис. 1.7. Функціональна схема мультисенсорної інтегрованої системи для бездротового моніторингу парникового середовища

Система має чотири сенсорні станції та взаємодіє з координаторською станцією за допомогою RF-модулів ZigBee. Міжміський зв’язок між координаційною станцією та центральною контрольною станцією використовує власний RF-модем.

Сенсорна станція (СС) відповідає за збір даних кліматичних вимірювань і передає їх координаційній станції (КС). Станція координатора діє як маршрутизатор, який заздалегідь запрограмовано керує потоком даних та інструкціями між Сенсорною станцією та Центральною станцією управління. Також керує включенням/вимкненням спринклерів, зволожувача повітря, тощо. Взаємодіє з сенсорами станцій відбувається за допомогою бездротового протоколу ZigBee. Модулі ZigBee з'єднані з мікроконтролером станції координатора за допомогою UART (універсальний асинхронний приймач-передавач). Він

зв'язується з центральною контрольною станцією за допомогою RF-модемів Xstream, які здатні передавати дані за допомогою дипольних антен і працюють на частоті 2,4 ГГц. Модем підключається до мікроконтролера за допомогою іншого каналу UART.

У функції центральної контрольної станції входить видати інструкцію комп'ютерам нижчого рівня, обробляти вхідні дані, надавати команди управління для регулювання кліматичних умов теплиці відповідно до вимог виробника. Додаток, що працює на CCS, розроблений у Visual C #. Програма має зручний графічний інтерфейс для моніторингу даних Sensor Station. Різні кліматичні параметри зображуються в режимі реального часу за допомогою графіків. CCS підключений до RF-модему через порт USB.

Таким чином мультисенсорні системи стають елементом дизайну із використанням штучного інтелекту. Інтелектуальні інновації для Smart Home існують вже деякий час, оскільки гаджети, програмне забезпечення та додатки для автоматизації дому здебільшого обертаються навколо конкретних будівель, теплиць та інших об'єктів моніторингу. Наступним етапом інтеграції штучного інтелектуального є здатність інтегрувати різні системи, постійно навчатись та адаптуватися за допомогою складного обстеження інформації. Таку процедура навчання контролює уніфіковану структуру, яка координує конкретні параметри і якою управляє центральний «мозок».

1.2. Огляд існуючих методів зв'язку в інтелектуальних вимірювальних системах

У сенсорних мережах IoT протоколи бездротового зв'язку широко використовуються для процесу обміну інформацією. Ці протоколи зв'язку працюють як неліцензовані діапазони частот, що полегшує гнучкість і масштабованість розгортання сенсорів. Однак використання протоколів зв'язку для WSN в неліцензованих діапазонах частот викликає неконтрольовані перешкоди. Сигнали перешкод можуть призвести до неправильної передачі даних і даних сенсорів із шумом, відсутніми значеннями, викидами та надмірністю. У цьому огляді розглядаються різні аналізи даних, які виконуються для вирішення проблем

із даними сенсорів IoT, таких як шумозаглушення, імпутація відсутніх даних, виявлення викидів даних та агрегація даних.

1.2.1 Зниження шуму

Об’ємні дані сенсорів, які генеруються в мережі Інтернету речей, потребують аналізу даних переважно з прийняттям рішень у реальному часі. Характеристики даних сенсорів є складними, що включають високі швидкості, величезні обсяги, а також динамічні значення та типи. Крім того, дані сенсори забруднюють, створюють численні перешкоди, доки не розроблять необхідний аналіз даних і не будуть приймати рішення в режимі реального часу. Шум, як правило некорельований компонент сигналу, який вносить небажані зміни та модифікації вихідних векторів сигналу. Методи вейвлет-перетворення здатні представляти сигнал і ефективно вирішувати проблему оцінки сигналу. Важливо, що вейвлет-перетворення зберігає вихідні коефіцієнти сигналу, усуваючи шум всередині сигналу. Це досягається шляхом встановлення порогового значення коефіцієнта шумових сигналів, тому ідеальна схема порогового значення є важливою. Нехай $e(t)$, $t \in P$ представляє енергію сигналу, тоді як вона повинна задовольняти обмеження, визначене як :

$$/e/^2 = \int_{-\infty}^{\infty} e(t)2\, dt < \infty \qquad (1.1)$$

де енергія сигналу $e(t)$, що задовольняє обмеження в рівнянні (1.1), належить пошуку в квадраті простір $L_2(R)$. Вейвлет-перетворення також використовується для аналізу енергії сигналу за дискретний час і ліквідувати в шум з енергії сигнали. Вейвлет трансформація метод дозволяє розслідувати характеристики сигналу шляхом масштабування в різних масштабах часу. Експериментальні результати показали значущі покращення в шумозаглушення в сигналах сенсора.

Існує два типи вейвлет-перетворень, а саме безперервне вейвлет-перетворення (CWT) і дискретний вейвлет перетворення (DWT).

Безперервне вейвлет-перетворення (CWT). У CWT енергія сигналу $e(t)$ представлена за допомогою набору вейвлет-функції $C=W_\psi(\alpha,\beta)$, $\alpha \in R^+$; $\beta \in R$, де

α - коефіцієнт розширення, а β – фактор локалізації часу зсуву, ψ - вейвлет функція. Вейвлет коефіцієнт на частотно-часовій площині визначається рівнянням (1.2):

$$W_\psi(\alpha,\beta) = \int_{-\infty}^{\infty} (1/\sqrt{\alpha})\psi_o((\lambda-\beta)/\alpha))e(\lambda)d\lambda, \quad (1.2)$$

де ψ_0 являє собою зміщену та розширену форму вихідного вейвлета $\psi_0(t)$. CWT - це функція контролюється двома параметрами. CWT має на меті знайти коефіцієнти вихідного сигналу e(t) на основі переміщення фактора f (β) і фактора розширення (α).

Дискретне вейвлет-перетворення (DWT). DWT для безперервних сигналів відноситься до перетворення сигналу, що виконуються за сигналом з дискретним часом. Коефіцієнти, отримані в результаті цього перетворення визначені в підмножині $D=W_\psi(2\alpha,2\alpha\beta)$, $\alpha \in Z$, $\beta \in Z$. Для даного безперервного сигналу e(λ), коефіцієнти DWT отримують за допомогою інтегрування підмножини D, як визначено рівнянням (1.3).

$$W(\alpha,\beta)=(\psi_0(2^\alpha,2^\alpha\beta), e) = \int_{-\infty}^{\infty} 2^{-\alpha/2}\, \psi_0`(2^{-\alpha}\lambda-\beta)e(\lambda)d\lambda \quad (1.3)$$

Автори роботи [36] обговорюють, що в деяких випадках сигнали, отримані від сенсорних пристроїв IoT, мають розумне співвідношення сигнал/шум (SNR), але не можуть досягти необхідного рівня бітових помилок (BER). Для подолання таких проблем найкращим рішенням є усунення нижчих вейвлет-коефіцієнтів. Це усунення покращує SNR на основі певного граничного значення. Це можливо, оскільки менші коефіцієнти мають тенденцію до більшої кількості даних шуму, ніж бажані дані сигналу. Далі зазначається, що енергетичні сигнали зосереджені на певній частині спектру сигналу. Таким чином, якщо ця конкретна частина спектру сигналу була перетворена за допомогою вейвлет-коефіцієнтів, це покращує значення SNR. Крім того, якщо сигнальна функція має великі області нерегулярних шумів і невеликі області плавного сигналу, то вейвлет-коефіцієнти відіграють життєво важливу роль у покращенні енергії сигналу.

Автори [36] докладно розглянули потокову передачу даних сенсорів і необроблених сигналів сенсорів, щоб розпізнати характеристики та різні проблеми, пов’язані із шумом від сигналів сенсорів.

1.2.2. Імпутація відсутніх даних

Імпутація є важливим завданням попередньої обробки в аналізі даних для роботи з неповними даними. Різні галузі, такі як розумні міста, охорона здоров'я, GPS, розумний транспорт тощо, використовують Інтернет речей як ключову технологію, яка генерує велику кількість даних [37]. Алгоритми навчання, які аналізують дані Інтернету речей, зазвичай припускають, що дані повні. Хоча відсутні дані є поширеними в Інтернеті речей, аналіз даних, який виконується для відсутніх або неповних даних IoT, може призвести до неточності або ненадійності результатів. Тому для IoT необхідна оцінка відсутнього значення. Для вирішення цієї проблеми необхідно виконати три основні завдання. Перше - знайти причину відсутності даних. Погане підключення до мережі, несправні системи сенсорів, фактори навколишнього середовища та проблеми з синхронізацією є різними причинами неповних результатів. Відсутні дані поділяються на три типи: відсутні повністю випадково (MCAR), відсутні випадково (MAR) і не відсутні випадково (NMAR). Друге завдання передбачає вивчення схеми відсутніх даних. Тут можливі два підходи – це монотонні відсутні шаблони (MMP) і випадкові відсутні шаблони (AMP). Третє завдання формулює модель імпутації відсутнього значення для IoT, щоб використовувати модель для наближення значення для відсутніх даних. У літературі деякі алгоритми імпутації відсутніх значень включають одиночні алгоритми імпутації, багатоваріантні алгоритми імпутації, тощо.

Алгоритми кластеризації. Модель гаусової суміші (GMM) є алгоритмом кластеризації [38]. Це імовірнісна модель, яка використовує підхід м'якої кластеризації для розподілу точок даних у різні кластери. Гаусова суміш визначається наступним чином: $G = \{GD_1, GD_2, \ldots, GD_k\}$, де k позначає число кластерів. Кожна GD_i є функцією розподілу Гаусса, яка містить середнє значення μ, яке представляє її центр, коваріація Σ і ймовірність π, яка вказує, наскільки велика чи мала функція Гаусса буде бути. Припускаючи набір даних створених з використанням GMM компоненти функція $f(k)$ (GD_i) представляють ймовірну щільність функції з k-го компоненту. Ймовірність з GD_i , $P(GD_i)$ створений за GMM, представляється рівнянням (1.4) .

$$P(GD_i) = \sum \pi_i f_i (GD_i \setminus \mu_i, \Sigma_i) \quad (1.4)$$

Щоб обробити відсутні дані, імпутація даних сенсора IoT за допомогою моделі GMM включає п’ять кроків, а саме: створення екземпляра, кластеризацію, класифікацію, вимірювання відстані та заповнення даних. По-перше, екземпляри в наборі даних D поділяються на два окремих набори даних (D_1 і D_2). D_1 містить усі екземпляри без пропущених значень, тоді як D_2 містить усі екземпляри в наборі даних, який має відсутні значення. По-друге, модель GMM на основі алгоритму EM використовується для кластерування повного набору даних D_1. Для кожного кластера визначається центр кластера. Після цього обчислюється кластер для кожного екземпляра в D_1. По-третє, за тестовий набір приймається неповний набір даних D_2. Кожен екземпляр у D_2 класифікується відповідно до результату кластеризації. Наприклад, $\alpha_1 \in D_2$, α_1 належить до кластера, якщо він є ближче до центру скупчення цього скупчення на евклідову відстань. На четвертому кроці для кожного екземпляра D_2 потрібно визначити один або кілька повних екземплярів, найближчих до α_1 в тому ж кластері, використовуючи евклідову відстань як міру відстані. Нарешті, потрібно заповнити пропущене значення екземпляра α_1, знайшовши середнє значення найближчого екземпляра в кластері.

Просторова та часова кореляція [39]. Вузли сенсорів періодично виявляють дані. Оскільки дані сенсора чутливі до часу, інші результати можна було б отримати, використовуючи дані інших сенсора для аналізу. Зв'язок між вузлами сенсорів у різні періоди неоднаковий, тому необхідно вибрати правильні дані для аналізу. На думку авторів, для точних даних потрібна відповідна вибірка даних аналізу. Для оцінки відсутніх даних автори пропонують алгоритм тимчасової та просторової кореляції (TSCA). По-перше, він одночасно зберігає всі отримані дані як часовий ряд і вибирає найважливіший ряд як вибірку аналізу, що значно підвищує ефективність алгоритму. По-друге, він оцінює відсутні тимчасові та просторові значення розмірів. Цим двом вимірам призначаються різні ваги. По-третє, є дві стратегії боротьби з серйозною втратою даних, що підвищує застосовність алгоритму. Основний робочий процес моделі TSCA зображено на рис. 1.8.

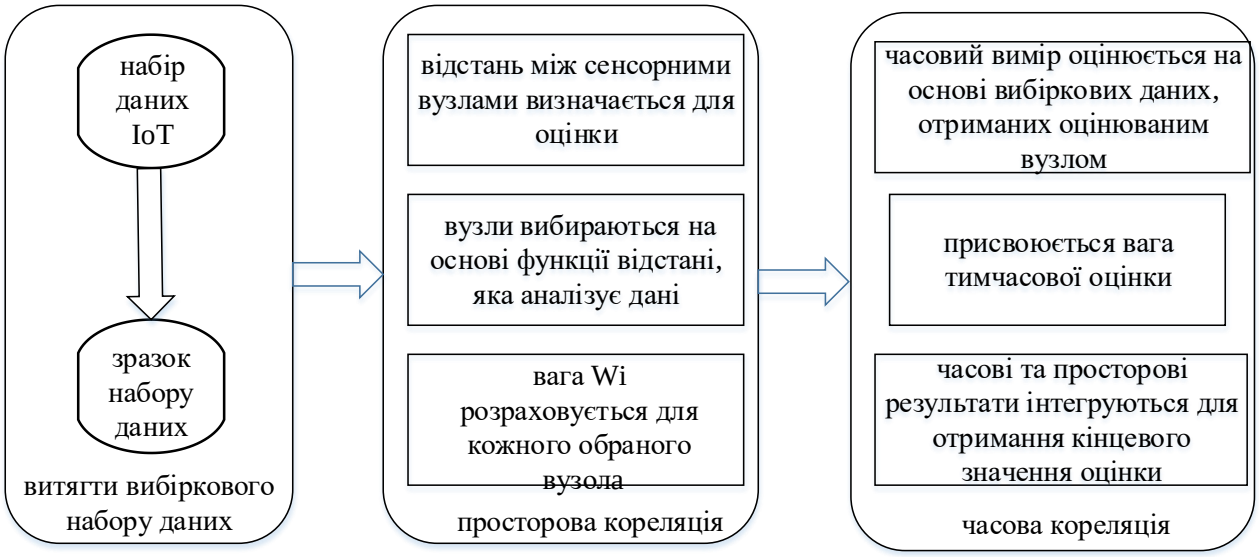


Рис. 1.8. Робочий процес алгоритму тимчасової та просторової кореляції

Модель передбачає, що всі сенсори підключені до одного діапазону зв'язку. Експеримент проводився на наборі даних про якість повітря. Першим кроком є цільовий набір даних, який отримується з вихідного набору даних. Встановлюється поріг відсутніх даних, який відрізняється від випадку до випадку. На наступному кроці встановлюють відсоток відсутніх значень. Якщо він перевищує поріг, то імпутація ігнорується; в іншому випадку здійснюється просторово-часова імпутація. На наступному кроці *n* сенсорів наближення обчислюють за формулою відстані Гаверсина. Співвідношення між *n* сенсорів наближення та сенсора з відсутнім значенням розраховують за допомогою Пірсона. Будується повний набір цільових даних і оцінюється на точність.

Автори роботи [40] запропонували новий метод імпутації найближчого сусіда для імпутації відсутніх значення на основі просторових і часових кореляцій між сенсорними вузлами. Дерево структури даних було розгорнуто для збільшення часу пошуку. На основі відсотка відсутніх значень, зважений дисперсії та зважені евклідові відстані були використані для створення kd-дерева. Рис. 1.9 ілюструє робочий процес просторово-часової моделі. Алгоритм, визначений у запропонованій моделі, встановлює поріг відсутнього значення як T. Відсоток P відсутніх значень обчислюється як обраний набір даних. Якщо P знаходиться в межах порогу, то він знаходить n наближених сенсорів через просторову кореляцію. Відсутні дані сенсорів обчислюється за показниками сенсорів наближення, що відповідають часу. Результат порівнюється з множинними

результатами імпутації. Точність оцінюється за допомогою середньоквадратичної помилки (RMSE).

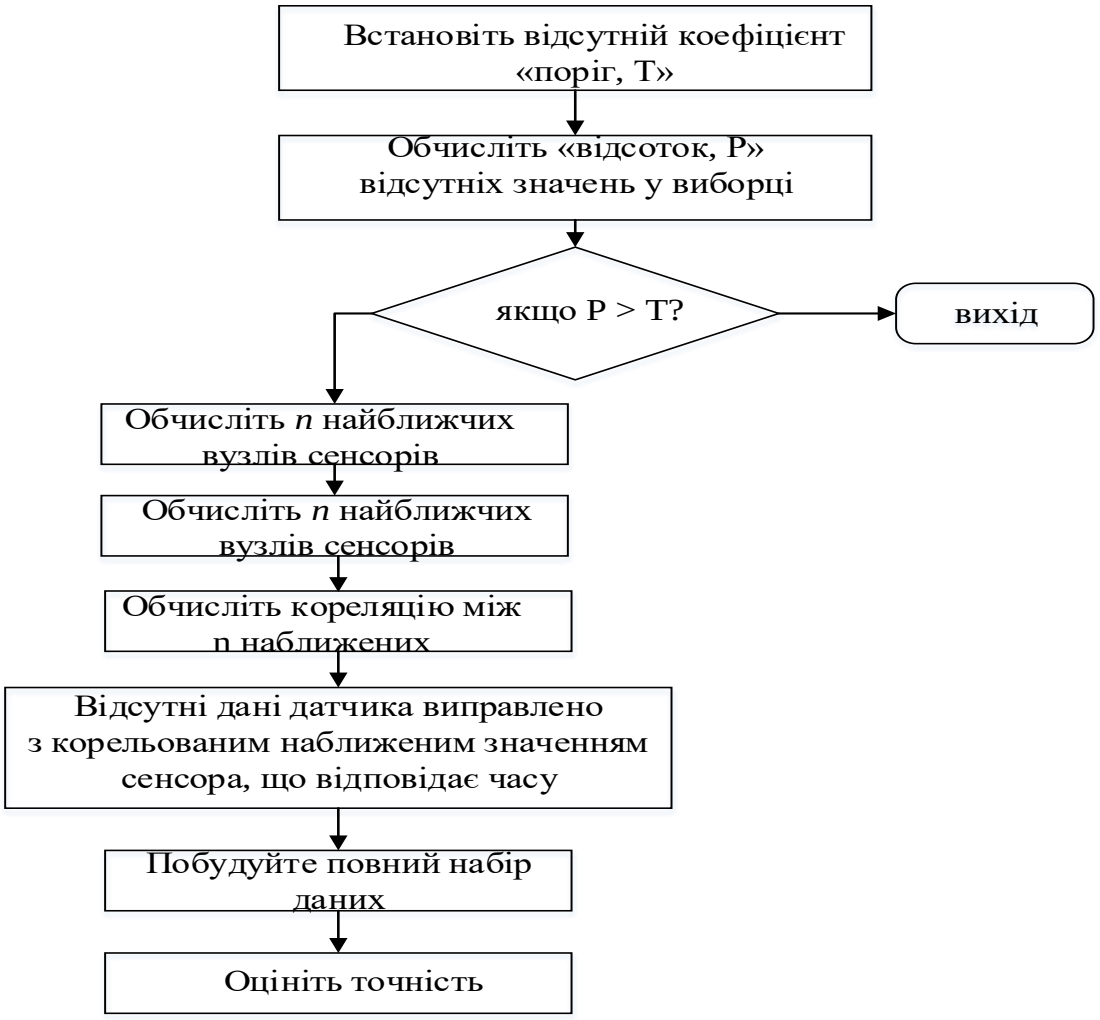


Рис. 1.9. Робочий процес просторово-часової моделі

Інкрементальна просторово-часова модель (ISTM). Інкрементальна модель, обговорена в [41], є моделлю, яка оновлює параметри існуючої моделі залежно від попередніх вхідних даних, а не створення нової моделі з нуля. Модель використовує інкрементальну множинну лінійну регресію до обробляти дані. Коли дані надходять, ця модель оновлюється після зчитування проміжної матриці даних замість доступу до всіх історичних даних. Якщо знайдено якісь відсутні дані, модель надає оціночне значення на основі історичних даних і спостережень найближчих сенсорів.

Факторизація ймовірнісної матриці: є дві основні переваги використання імовірнісної матриці факторизації (PMF) [42] для обробки відсутніх даних сенсорів IoT. По-перше, це зменшення розмірності, що є основною властивістю матричної факторизації. По-друге, вихідна матриця може відтворюватися за допомогою добутку факторних матриць. Цей метод використовується для відновлення відсутніх значень в оригінальній матриці. Факторизація виконується на попередньо призначених сенсорах. Дані сусідніх сенсорів перевіряються на схожість і групуються в інший клас кластерів за допомогою Алгоритм K-середніх. Алгоритм кластеризації K-середніх групує сенсори в окремі класи відповідно до їх вимірювання подібності. Аналіз шаблонів сусідніх сенсорів допомагає відновити відсутні дані сенсорів. Потім у кожному кластері реалізується імовірнісний матричний алгоритм факторізації. PMF стандартизує дані та обмежує ймовірнісний розподіл випадкових ознак. Алгоритм робочого процесу зображений на рис. 1.10.

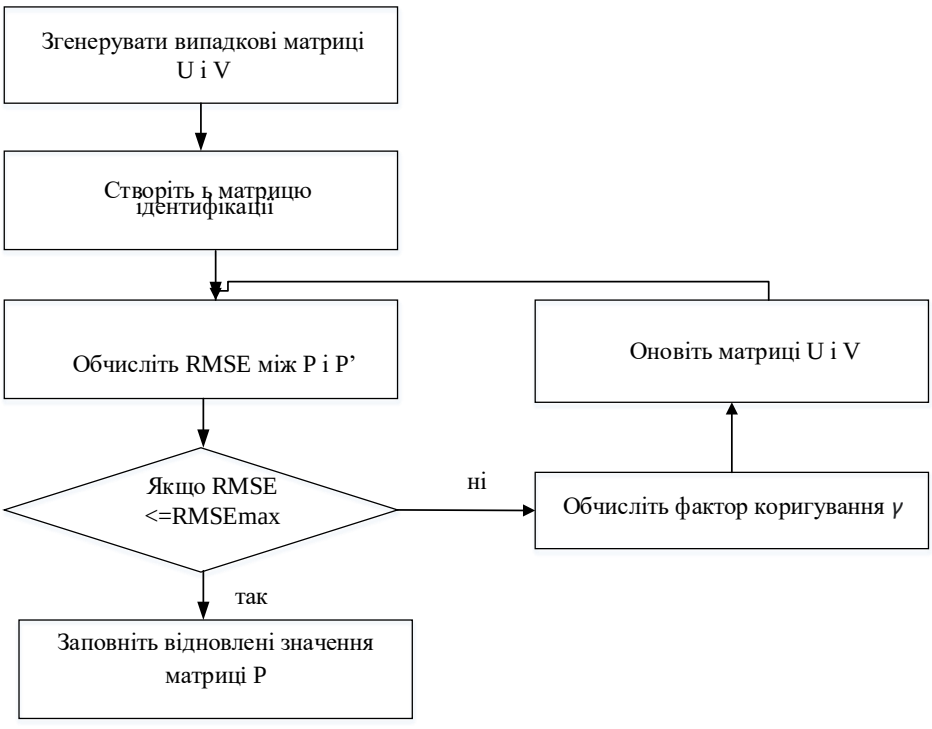


Рис. 1.10. Робочий процес в PMF моделі

Вихідна матриця представлена як P_{NxM}. Згенеровані дві випадкові матриці, U і V, такі, що $P' = U.V^T$, де U та V мають розмірність NxK та KxM відповідно. K - ціле число, яке представляє кількість векторів стовпців прихованих ознак в U та V. Відсутні точки даних у вихідній матриці представлені у вигляді ідентичної матриці,

і має той самий розмір, що й вихідна матриця P. Значення в матриці I_{ij} визначаються за допомогою рівняння (1.5):

$$I_{ij} = \begin{cases} 1, if\, P_{ij}\ is\ not\ missing. \\ 0, if\, P_{ij}\ is\ missing \end{cases} \quad (1.5)$$

Далі обчислюється середньоквадратична помилка (*RMSE*) між P і P', яка наведена рівнянням (1.6)

$$RMSE = \sum^{N}_{i=1} \sum^{M}_{j=1} I_{ij} (P_{ij} - U_i V^{T}_{j})^2 \quad (1.6)$$

Результат, отриманий за рівнянням (1.6), порівнюється з $RMSE_{max}$, що є максимально допустимою помилкою. Алгоритм завершиться, якщо $RMSE \leq RMSE_{max}$. Інакше значення U і V оновлюються за допомогою рівнянь (1.7) та (1.8).

$$U` = U_i + \gamma\, (\partial RMSE_{ij}) / \partial U_i \quad (1.7)$$

$$V` = V_j + \gamma\, (\partial RMSE_{ij}) / \partial V\,, \quad (1.8)$$

де γ позначає коригувальний коефіцієнт, який вирішує, наскільки значення U та V необхідно відкоригувати. Кроки з четвертого по шостий повторюються до тих пір, поки RMSE не стане менше або дорівнює $RMSE_{max}$. Велике значення γ може призвести до низької точності, тоді як занадто мале значення може призвести до занадто великої кількості ітерацій.

Автори роботи [43] розглянули питання про відсутнє значення в сенсорах IoT. У сенсорних мережах IoT, однорежимні збої викликають втрату даних через несправність кількох сенсорів у мережі. Автори запропонували множинний сегментований підхід до імпутації, в якому була прогалина в даних ідентифіковані та сегментовані на частини, а потім приписані та реконструйовані ітераційно з релевантними дані. Експериментальні результати продемонстрували значні покращення порівняно зі звичайною технікою середньоквадратичне.

1.2.3. Виявлення викидів даних

У сенсорній мережі IoT трапляються що неоднорідні вузли сенсорів. Це в реальному фізичному середовищі призводить до величезних збоїв згенерованих ними вихідних даних [44]. Тому важливо визначити такі викиди даних перед аналізом даних і прийняттям рішень. Для цього на основі просторової кореляції

даних виявлення викидів виконується і здійснюється за допомогою трьох популярних методик, а саме голосування більшістю, класифікатори та аналіз головних компонентів.

Голосування більшістю: у цьому методі визначається вузол сенсора, який функціонує ненормально, дає інший результат в зчитуванні в порівнянні із сусідніми сенсорними вузлами. За даними авторів [45] Пуассон розповсюдження - це звичайний метод генерації даних у різних мережевих сенсорних додатках IoT. В Інтернет сенсорній мережі створені набори даних складаються з викидів для короткострокових, неперіодичних і незначних зміни тенденцій даних. Прості та ефективні статистичні методи для виявлення викидів у методі Пуассона розподіл набору даних мережі сенсорів IoT є стандартним відхиленням. Так само в розподіленому середовищі оцінюють відхилення між даними, згенерованими несправним сенсорним вузлом з даними отриманими із сусідніх вузлів. Якщо оціночна різниця в даних перевищує певний пороговий ліміт, тоді дані, згенеровані цим вузлом, ідентифікуються як викиди. Хоча ця техніка проста і менш складна, вона надмірно залежить від сусідніх сенсорних вузлів. Крім того, точність у випадку розрідженої мережі низька.

Класифікатори: цей метод включає два кроки, перший - навчання даних сенсора IoT за допомогою стандарту модель машинного навчання. Другий - дані виявляються за допомогою алгоритмів класифікатора як будь-який нормальний або аномальний [46]. Зазвичай використовуваними алгоритмами класифікатора є машина опорних векторів (SVM). Половина простору пошуку даних навчена бути стандартними даними. Пізніше дані аналізуються і класифіковані через SVM для виявлення в рамках навчених даних стандартних даних або інакше ненормальні дані. Однак недоліки алгоритму класифікатора полягають у його високій складності.

У [47] автори розглянули питання виявлення викидів у даних сенсора IoT за допомогою машини Tucker і генетичного алгоритму. Різні сенсорні вузли, задіяні в сенсорній мережі IoT, демонструють просторові атрибути та дані зондування. Крім того, отримані дані сенсора є динамічними для часу. Як правило, великі дані

містять аномалії через збій режиму. Звичайні засоби виявлення викидів включають векторні алгоритми. Недоліки векторних алгоритмів турбують вихідну структурну інформацію даних сенсора і демонструють побічний ефект розмірності. Тензорне рішення для виявлення викидів з використанням факторизації Такера продемонстрували підвищення ефективності та точність виявлення викидів без порушення внутрішньої структури даних сенсора.

1.2.4. Агрегація даних

Метод агрегації даних називається методом збору та передачі даних інформацію в зведеному вигляді. Це можна використовувати для статистичного аналізу. В IoT неоднорідний дані збираються з різних вузлів. Надсилання даних окремо від кожного вузла призводить до високого рівня споживання енергії, а також потрібна висока пропускна здатність мережі, що скорочує термін служби мережі. Методи агрегації даних запобігають подібним проблемам шляхом узагальнення даних, що зменшує надмірну передачу даних, збільшує термін служби мережі та зменшує мережу трафіку. Агрегація даних в Інтернеті речей (IoT) допомагає зменшити кількість передач між об'єктами. Це подовжує термін служби мережі та зменшує споживання енергії [48]. Це також зменшує мережевий трафік. Методи агрегації даних в Інтернеті речей класифікуються наступним чином:

а) підхід на основі дерева [49-55] - цей підхід розгортає вузли в мережі у вигляді дерево. Тут ієрархічні та проміжні вузли виконують агрегацію. Агрегований потім дані передаються до кореневого вузла. Підхід на основі дерев підходить для вирішення проблеми енергії споживання та термін служби мережевих проблем;

в) кластерний підхід [56-60] - вся мережа організована як кластери. Кожен кластер буде містити кілька сенсорних вузлів з одним вузлом як головкою кластера, який виконує дані агрегація. Цей метод спрямований на здійснення ефективної реалізації енергоагрегації у великих мережах. Це допомагає зменшити споживання енергії вузлами з обмеженою енергією в величезні мережі. Це призводить до зменшення пропускної здатності через передачу та обмежену кількість пакетів. У

випадку статичних мереж, де кластерна структура не є зміщення протягом тривалого часу, кластерні методики є успішними.

с) централізовано [61,62,63] - усі сенсорні вузли в цій системі надсилають дані до вузла заголовка. Вузол заголовка відповідає за агрегування даних і відправлення одного пакета.

1.3. Підходи інтелектуалізації та їх використання у вимірювальних систем

Для підвищення точності потрібна інтеграція або злиття даних з кількох сенсорів додатків при відстеженні цілі у військовій системі або системі спостереження, відстеження точного місцезнаходження дорожнього транспортного засобу чи знаходження положення перешкоди у венах людського тіла, тощо. Застосування злиття даних у різних областях коротко пояснюється таким чином [64,65]:

- підхід до злиття даних із кількома сенсорами може застосовуватися на кораблях, літаках, фабриках тощо. У цих системах дані від електромагнітних сигналів, акустичних сигналів, сенсорів температури, рентгенівських променів, тощо, можуть бути інтегровані для перевірки та підвищення точності. Ця інтеграція підвищить точність і зміцнить довіру до системи, що корисно для своєчасного обслуговування діяльності, виявлення несправностей системи та віддаленого доступу виправлення, тощо;

- медична діагностика є складною системою, яка включає людський організм і використовується при діагностуванні захворювань, наприклад, пухлин, аномалій легенів або нирок, внутрішніх захворювань тощо. Застосовують ЯМР, хімічні або біологічні сенсори, рентгенівські промені, ІЧ, тощо;

- супутники, літаки, підземні чи підводні машини використовують сейсмічне випромінювання, електромагнітне випромінювання, хімічні або біологічні сенсори для збору точної інформації або виявлення природних явищ, моніторинг навколишнього середовища з дуже великих відстаней;

- військові та оборонні служби використовують цю техніку для спостереження за океаном, оборони, розвідки на полі бою, збір даних,

попередження, системи захисту, тощо, використовуючи ЕМ випромінювання з великих відстаней.

З'єднання сенсорів – це процес об'єднання двох або більше джерел даних таким чином, щоб забезпечити узгодженішу, точнішу та надійнішу оцінку динамічної системи. Ця оцінка дає кращі результати, ніж при використанні сенсорів окремо. Метою злиття сенсорів є мінімізація вартості, складності пристрою та кількості задіяних компонентів, а також покращення зондування точності і впевненості.

Джерелами даних можуть бути сенсори або математичні моделі, а станом системи може бути прискорення, відстань тощо. Чотири різні причини використання з'єднання сенсорів включають те, що це, по-перше, підвищує якість даних, по-друге, підвищить надійність, по-третє, можна вимірювати невиміряні стани і можна збільшити зона покриття.

Загалом, методи злиття даних можна охарактеризувати як ймовірнісні, статистичні, засновані на знаннях, а також методи висновків і міркувань. До імовірнісних методів належать байєсівські мережі, методи оцінки максимальної правдоподібності, теорія висновків, фільтрація Калмана, тощо. Методи включають коваріаційний, перехресний та інший статистичний аналіз [66]. На основі знань методи включають штучні нейронні мережі, нечітку логіку, генетичні алгоритми тощо [67,68]. В залежності від специфікації проблеми, слід вибрати відповідні методи злиття даних.

Байєсівський метод: злиття даних із кількома сенсорами є суттєвою властивістю байєсівської статистики. Тому всі невідомі величини вважаються випадковими величинами.

Основний закон Байєса:

$$P(\alpha,(\beta_1, \beta_2) \propto P(\alpha)L(\alpha; \beta_1)L(\alpha; \beta_2), \tag{1.9}$$

де попередня щільність на α представлена як $P(\alpha)$, ймовірність α через β представлена як $L(\alpha; \beta)$, такий, що ймовірність пропорційна $P(\alpha, \beta)$, а умовна ймовірність визначається як рівняння:

$$L(\alpha; \beta) \propto P(\alpha;\beta) \tag{1.10}$$

Це показує ймовірність отримання даних сенсора β з урахуванням апріорного значення α. Злиття двох даних різних вимірювань сенсорів α і β, задані неінформованим попереднім, таким як P(α), потім задаються

$$P(\alpha, \beta_1, \beta_2) \propto P(\alpha)L(\alpha; \beta_1)L(\alpha; \beta_2) \quad (1.11)$$

$$1 \times exp\ ½\ ((\alpha - \beta_1)/\delta_1)^2 \times exp\ ½((\alpha - \beta_2)/\delta_2)^2 \quad (1.12)$$

Метод фільтра Калмана: використовується для оцінки стану системи, коли його неможливо виміряти безпосередньо. Це ітераційний математичний процес, який використовує набір рівнянь і введених даних, виміряних у часі, містить шум і неточності. Він виробляє оцінки цих невідомих параметрів, які є більшими точніше, ніж результати вимірювання одного сенсора, за допомогою функції спільного розподілу над кожною змінною часового кроку.

Модель фільтра Калмана передбачає еволюцію стану в момент k від стану в $(k-1)$, відповідно до наступного рівняння.

$$x_k = Fx_{k-1} + Au_{k-1} + w_{k-1,} \quad (1.13)$$

де F - матриця переходу станів, яка застосовується до векторів попереднього стану x_{k-1}, а A - вхідна матриця керування, яка застосовується до вектора керування u_{k-1}. w_{k-1} - це вектор шуму передбачається, що це багатовимірний розподіл Гауса, отриманий з нульового середнього з коваріацією Q_k, де $w_{k-1} \sim N(0,Q_k)$. Вимірювання на кроці часу k спостерігається як

$$z_k = Hx_k + v_k, \quad (1.14)$$

де H - матриця вимірювання, z_k - вектор вимірювання, v_k - вектор вимірювання шум, який вважається багатовимірним гауссовим розподілом, отриманим із нульового середнього за допомогою коваріація R, тобто $v_k \sim N(0,R)$. Метою фільтра Калмана є надання оцінки x_k в момент k, з огляду на початкову оцінку x_0, серії вимірювань $z_1, z_2, \ldots, z_k$ та інформації системи, описаної F, A, H, Q і R. Три основні підходи можна використовувати для злиття або інтеграції мультисенсорних даних.

Пряме злиття: у цьому підході всі сенсори зв'язуються між собою для класифікації. Після цього відбувається інтеграція або злиття на рівні даних. Після того, як дані відповідного сенсора будуть інтегровані, функції даних витягуються. Цей процес прямого злиття також відомий як спільна декларація посвідчення, де

декілька ідентифікаторів асоціації сенсорів оголошуються спільно. Процес прямого злиття формально оформлений як показано у рівняннях (1.15)–(1.18).

$$O{:}S_i \rightarrow S_j \ \forall i \neq j \ \ i \in \{1, 2, 3, \ldots, n\}, \ j \in \{1, 2, 3, \ldots, n\} \quad (1.15)$$

$$P : f_{fe} \ (f_{df}(f_A(O))) \quad (1.16)$$

$$ID_{data\_extraction}(S_i) = g(P) \quad (1.17)$$

$$Q: JID_{declaration}(S_i) \quad (1.18)$$

Тут S_i представляє i-й сенсор, який розглядається в процесі злиття даних, а O є результатом i-го сенсора із відображення j-го сенсора. $f_A(.)$, $f_d(.)$ та $f_{fe}(.)$ - асоціація даних, внутрішній рівень злиття даних та функції вилучення функцій у повному процесі злиття даних. Дані окремих сенсорів ідентифікація досягається застосуванням функції оголошення ідентифікації (g) над виділенням результату ознак (P). Нарешті, Q є результатом спільної ідентифікації в підході прямого злиття з використанням функція $JID_{declaration}(.)$ та результату Q.

Вилучення ознак з подальшим злиттям: у цьому підході отримуються характеристики даних сенсора, потім асоціація даних на основі функцій. Ці асоціації полегшують злиття дані на основі декларації асоціації та ідентифікації об’єкта. Нарешті, дані на основі функцій декларація злиття та ідентифікації призводить до спільного декларування ідентифікації сенсора та інтеграції даних. Цей підхід формально представлено, як показано у рівняннях (1.19)-(1.22).

$$R : f_{fe}(S_i) \ \forall i \in \{1, 2, 3, \ldots, n\} \quad (1.19)$$

$$U : g(f_A(R)) \quad (1.20)$$

$$V : H(f_A(R)) \quad (1.21)$$

$$Q: I(U, V) \quad (1.22)$$

Тут R є результатом вилучення ознак з кожного сенсора. $H(.)$ - функція для інтегрування злиття даних на рівні функцій. Нарешті, результат злиття даних Q для спільної декларації ідентифікації обчислюється шляхом застосування паралельної функції інтеграції $I(.)$ до результату злиття на рівні ознак (U) та результату декларації особи (V).

Вилучення ознак з подальшим оголошенням ідентичності та злиттям для висновків або рішень високого рівня: підхід, дані сенсора спочатку витягуються.

Після цього дані окремих сенсорів ідентичності оголошуються з витягнутих ознак. Ці унікальні ідентифікатори допомагають асоціювати дані або в пошук необхідних даних. Для рівня оголошення використовується унікальна асоціація даних на основі ідентифікації злиття та декларація ідентичності. Обидва ці процеси призводять до спільної декларації особи. Цей підхід формально представлено, як показано у рівняннях (1.23)-(1.28).

$$R : f_{fe}(S_i)\ \forall,\ \in \{1, 2, 3, \ldots, n\} \tag{1.23}$$

$$U : g(R) \tag{1.24}$$

$$S_I^{Di} = J(U) \tag{1.25}$$

$$V: H(f_A\ (U, S^{ID}{}_i)) \tag{1.26}$$

$$W: g(f_A(U\ ,S^{ID}{}_i)) \tag{1.27}$$

$$Q : I(W,V) \tag{1.28}$$

Підхід до об’єднання даних із багатьма сенсорами також працює в багаторівневій архітектурі. У багарівневій архітектурі [69,70], підходи до злиття або інтеграції мультисенсорних даних працюють на різних рівнях.

1.4. Сучасні тенденції інтелектуалізації інформаційно-вимірювальних систем

Опрацювання робіт авторів [71], свідчить, що ключовими проблемами сенсорних мереж IoT є масштабованість і точність даних сенсора. Ці проблеми вирішуються за допомогою аналізу даних сенсора, а такі методи як збір даних, питання очищення, керування даними, виявлення знань, даних моніторингу є невід’ємною складовою цього процесу. У цьому аспекті моделі машинного та глибокого навчання відіграють життєво важливу роль у отриманні результатів, які включають формування знань і прийняття рішень.

Моделі машинного навчання: автори в [72] розглянули проблему, яка вимагає ефективного механізму отримання значущої інформації даних сенсорів IoT. У випадку сенсора IoT, аналітики даних, моделі машинного навчання повинні виконуватися в межах вбудованого сенсора. Це вимагає налаштування системних програм і ефективної структури даних для особливої обробки даних сенсорів IoT в

реальному часі. Таким чином, автори запропонували модель гаусової суміші (GMM) як рішення для роботи з різними функціями даних сенсорів.

1.4.1. Моделі глибокого навчання

В роботі [73] авторами запропоновано вивчення функцій для використання даних сенсорів IoT механізмом глибокого навчання. Дані сенсорів IoT надають незрозумілі функції, які потрібно зробити точними завдяки класифікації на основі механізму глибокого навчання в реальному часі. Однак моделі глибокого навчання є обчислювально дорогі з точки зору виконання на сенсорних платах з обмеженими ресурсами. Тому авторами і запропоновано механізм попередньої обробки в спектральній області, а тоді глибшим вивченням отримання моделі.

1.4.2. Нейронна мережа для обробки/аналізу сенсорів IoT

Штучні нейронні мережі добре підходять для задач апроксимації функцій і розпізнавання образів з використанням методів навчання під наглядом. Архітектура ANN включає вхідний рівень, вихідний рівень і один або більше прихованих рівнів. Згорткові нейронні мережі (CNN) складаються з мережевого рівня, який використовується для операції згортки застосовується до двовимірних або одновимірних даних сенсора [74]. CNN здебільшого підходить для даних зображень, які використовуються для аналізу даних зображення сенсора. Більшість пристроїв IoT, таких як дрони, розумні автомобілі, смартфони тощо, оснащені камерами. CNN приймає зображення/мову/сигнал, який є 2D або 1D, і функції високого рівня витягуються через ряд прихованих шарів. Приховані шари включають шар згортки та повністю пов’язаний шар об’єднання в кінці. Шар згортки має набір фільтри, відомі як навчальний параметр, який є ядром CNN, який фільтрує багатовимірні дані в нижчий вимір, що допомагає витягти найбільш відповідну функцію з вхідного зображення.

Важливим аспектом у інформаційно-вимірювальних системах, які реалізовані на базі технології IoT є використання штучних нейронних мереж (ANN) для ефективних обчислень на основі виміряних параметрів. Відомо, що сучасна ANN

являє собою одну із обчислювальних систем, яка містить блок навчання та блок прийняття рішень. ANN ще називають багатошаровим персептроном (MLP), який складається з вхідного шару, прихованого шару та вихідного шару (рис. 1.11) [75]. Ваги прихованих шарів оптимізовані з урахуванням характеристик даних та шаблонів. Шари можуть бути розширені, щоб легко обробляти складні дані та систему.

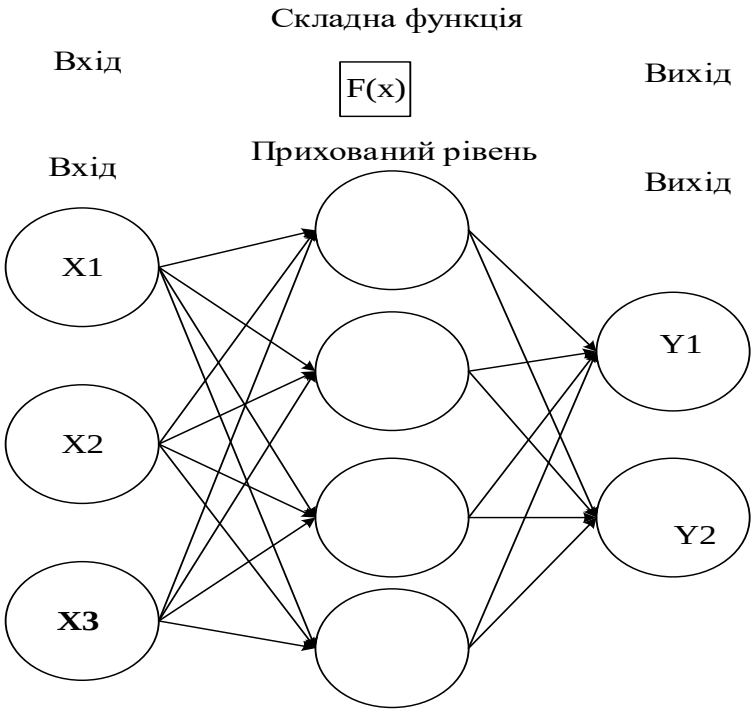


Рис. 1.11. Структура типової штучної нейронної мережі (ANN)
з алгоритм навчання

Звичайні сенсорні системи обробляють і аналізують лише вихідні дані системи. Однак, розробляючи ROIC, який може відображатись шляхом зворотного зв'язку інформації з аналізу, загальна продуктивність системи може бути ефективно покращена. Ця теза пояснює попередню реконфігуровану систему, систему сенсорів та АЦП. Для різних типів сенсорів необхідні відповідні інтерфейси зчитування, а також різні галузі застосування вимагають, щоб кожен інтерфейс відповідав властивим для конкретного застосування характеристикам. Однак проста інтеграція різних інтерфейсів сенсорів в єдину ROIC не має сенсу і не відображає різні вимоги. Ще однією тенденцією є те, що вони все частіше вбудовують додаткові функції, обробляючи дані, зібрані з декількох сенсорів.

Розробити відповідну ROIC непросто, і це може бути неефективним з точки зору загальної реалізації системи. Тому необхідно реалізувати багатофункціональну ROIC для зондування сигналу декількох сенсорів.

1.4.3. Мультисенсорні системи з використанням штучного інтелекту

Сьогодні мультисенсорні системи стають елементом дизайну з використанням штучного інтелекту Smart Home. Інновації в розумному домі існують вже деякий час, і гаджети, програмне забезпечення та програми для домашньої автоматизації здебільшого обертаються навколо конкретних будівель або приміщень. Наступним кроком в інтеграції будинку зі штучним інтелектом є можливість інтегрувати різні системи та постійно навчатися та адаптуватися за допомогою складного інформаційного опитування. Ця процедура навчання контролює єдину структуру, яка координує важливі структури в будинку, такі як освітлення, ізоляція, безпека, звук, жалюзі тощо. У цій ситуації розумний будинок нагадує екосистему, керовану центральним «мозком» за допомогою смартфон [76]. Домашня автоматизація дозволяє людині дистанційно або природним чином контролювати речі навколо будинку.

У загальній концепції таких автоматизованих систем ключову роль відіграє Alexa від Amazon – одна інновація штучного інтелекту, яка впливає на домашню автоматизацію [77]. «Мозок» Алекси можна координувати з різними гаджетами, які мають динамік і підсилювач, що робить її розумнішою, тож вона може виконувати нові завдання, такі як перегляд новин або зміна елементів керування в кімнаті.

У роботі [78] пропонується система домашньої автоматизації з мінімальними зусиллями та дистанційним керуванням (рис. 1.12). Платформа оновлює віддалену технологію Android, щоб забезпечити віддалений доступ зі смарт-мобіля. Конструкція замінює існуючі електричні вимикачі та забезпечує більший контроль над автоматичними вимикачами з технологією спрацьовування низької напруги.

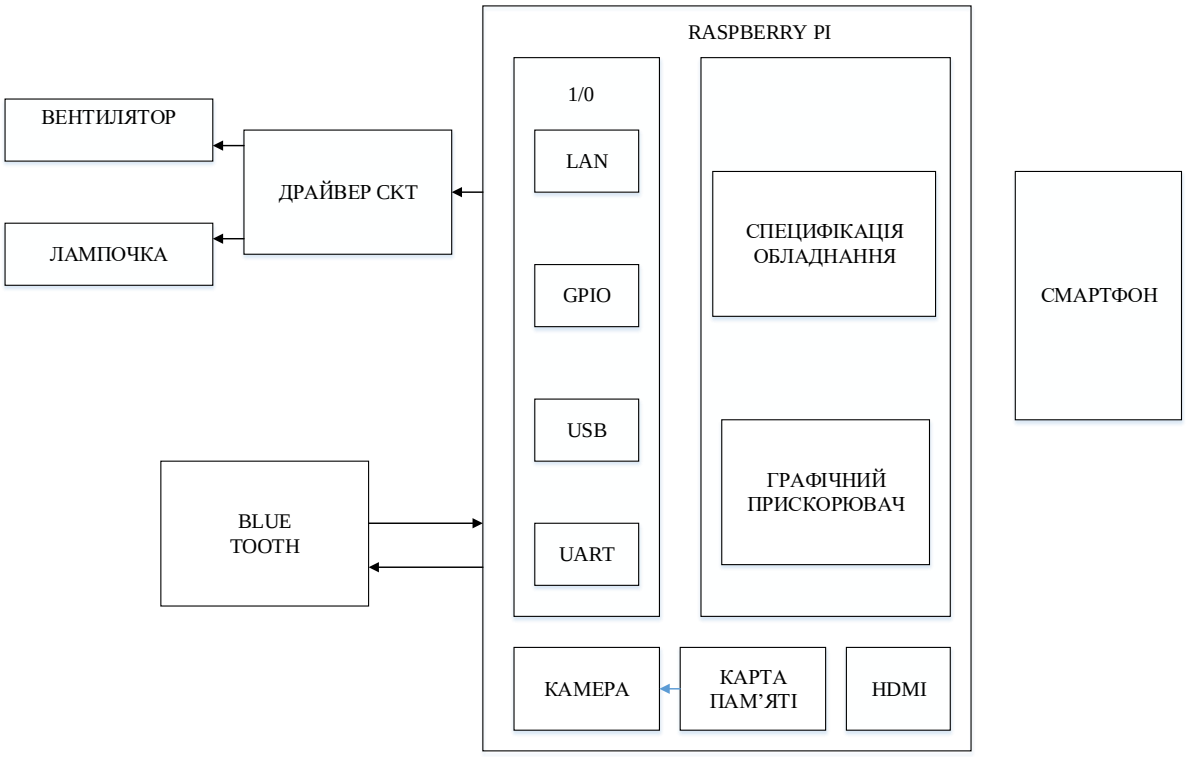


Рис. 1.12. Система домашньої автоматизації з мінімальними зусиллями та дистанційним керуванням

У системі використовується одноплатний комп'ютер Raspberry Pi B+. Контролер підключений до веб-сервера в Інтернеті, отримує необхідну користувачеві інформацію з будь-якої точки. Він оснащений 4 портами USB, одним портом ETHERNET, картою пам'яті, маршрутизатором, колонкою та інтерфейсами HDMI.

Оптимальний вибір архітектури MSDF є багатоцільовою задачею оптимізації, і симуляція системи може бути використана як стандартизований засіб представлення архітектури MSDF [79]. Інтеграція кількох сенсорів у систему, що використовується для оцінки явища, передбачає використання методів MSDF. Розподілена мультисенсорна система, як правило, складається з кількох розподілених вузлів зондування (сенсорів) і одного або кількох вузлів обробки (процесорів), усі взаємопов'язані. У розподілених мультисенсорних системах архітектура MSDF фіксує стратегічні рішення щодо розподілу завдань обробки між набором розподілених процесорів. Архітектура MSDF складається з трьох компонентів [79,80]: 1) комунікаційний граф (відображає мережевий зв'язок між

вузлами, тобто сенсорами та процесорами), 2) інформаційний граф (представляє детальний потік інформації між взаємопов'язаними вузлами), 3) інформаційний вміст (визначає, що саме передається між вузлами). Комунікаційний граф описує структуру, тоді як інформаційний граф описує поведінку.

Існують дослідження [81-85], які використовують різні критерії (набори) класифікації, що, ймовірно, свідчить про те, що жодне з цих досліджень не має повних висновків щодо класифікації типів архітектури MSDF.

Два критерії класифікації розглядаються в [86]. Перший критерій (C1) – це місце обробки композиції. Відповідно до C1, обробка треків може бути: 1) центрального рівня (спостереження надсилаються від сенсорів до віддалених процесорів для спільного формування глобальних треків), 2) рівня сенсора (локальні треки, сформовані локальними процесорами, приєднаними до сенсорів, надсилаються до віддалених процесорів, які виконують синтез із доріжки для отримання глобальних доріжок), і 3) гібридний (поєднання обробки центрального та локального рівнів). Другий критерій полягає в тому, де створюється та підтримується файл треку (база даних). Відповідно до C2, обслуговування файлів треків може бути: 1) централізованим (файл треків підтримується одним центральним процесором) і 2) розподіленим (файл треків підтримується кількома розподіленими процесорами). Відповідно до третього критерію (C3), архітектура може бути: 1) одноз'єднувальною (існує лише один шлях зв'язку від кожного сенсора до кожного процесора) і 2) багатоз'єднувальною (існує кілька шляхів зв'язку від принаймні одного сенсора до принаймні один процесор). В [87] проаналізовано вісім типів архітектури: 1) розподілене відстеження з відповідальністю за звітність, 2) композитне відстеження чистого центрального рівня, 3) практичне складене відстеження центрального рівня, 4) розподілене відстеження з центральним рівнем відстеження об'єднання, 5) розподілене відстеження з розподіленим об'єднанням доріжок, 6) розподілене відстеження з синтезом доріжок на центральному рівні та трекетами, 7) розподілене відстеження з розподіленим об'єднанням доріжок і трекетами, і 8) розподілене складене відстеження.

1.4.4. Моделі злиття даних в інтелектуальних системах моніторингу

Значна кількість робіт присвячена порівнянню системи моніторингу з існуючими комерційними системами, а також алгоритмам аналізу даних для покращення їх інтелектуальності та синтезу або об’єднання даних, розширенню поточної системи моніторингу на інші сфери застосування. Чотири рівні злиття відзначені для рівня вимірювання 0 (виведення функцій і шаблонів з вихідних даних і даних вимірювання; Рівень 1 (визначення параметричних і атрибутивних станів цільової сутності)); Рівень 2 (оцінка ситуації суб'єкта господарювання та його впливу на пов'язані суб'єкти господарювання); Рівень 3 (прогнозування майбутнього впливу на основі поточної ситуації).

Алгоритми злиття даних запропоновані в [88]. Щоб урізноманітнити їх застосування, існують різні способи класифікації алгоритмів синтезу. Один підхід базується на рівні абстракції, тобто на рівні сигналу, рівні функції та рівні прийняття рішення, і це зображено на рис. 1.13.

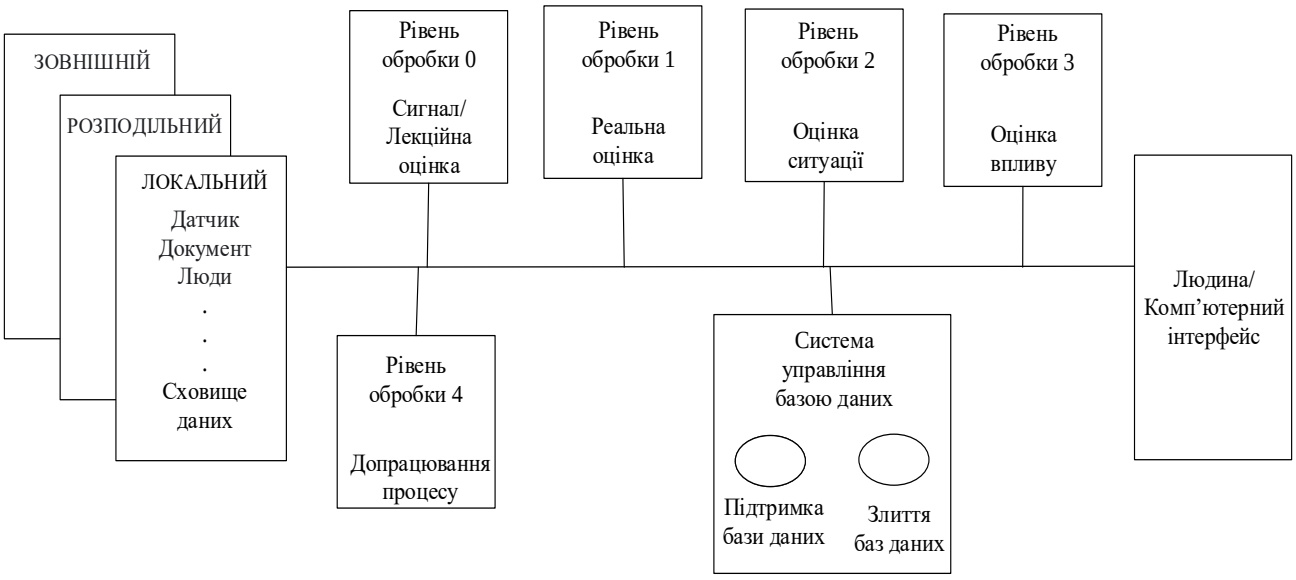


Рис. 1.13. Адаптована уніфікована модель об’єднання даних

Тут показано загальне застосування та виділено комбіновані результати. На рівні сигналу процес синтезу забезпечує точніші дані з точки зору певних показників якості даних, наприклад співвідношення сигнал/шум. Злиття на рівні функцій розроблено для отримання більш розрізнювальних функцій для інших

завдань. Рішення бере символічні представлення як джерела та об’єднує їх для отримання більш точного рішення [89].

Штучні нейронні мережі також добре підходять для апроксимації функцій і проблем розпізнавання образів з використанням методів навчання під керівництвом. Більшість пристроїв IoT, таких як дрони, розумні автомобілі, смартфони тощо, оснащені камерами. Багато додатків IoT, як-от повені або зсуви, прогнозують лісові пожежі за допомогою зображень дронів і трафіку управління, використовувати камери автомобіля. CNN приймає зображення/мову/сигнал, який є 2D або 1D, і функції високого рівня витягуються за допомогою ряду прихованих шарів. Приховані шари включають шар згортки та повністю з’єднаний шар об’єднання на кінці. Шар згортки має набір фільтрів, відомі як параметри, які можна вивчати, який є ядром CNN, який фільтрує високовимірні дані до меншого розміру, що допомагає витягти найбільш відповідну характеристику з вхідного зображення.

Рекурентна нейронна мережа (RNN) - це нейронна мережа, яка навчається на основі послідовностей або часу серійні дані [90]. Деякі завдання, наприклад, виявлення поведінки водія в розумних транспортних засобах, ідентифікація деякі моделі пересування або оцінка споживання електроенергії домашнім середовищем, може залежати від попередніх зразків для передбачення. У цих випадках RNN буде правильним вибором. RNN добре підходять для прогнозування даних сенсорів часового ряду.

Автокодери (AE) - це нейронна мережа, яка складається з однакової кількості вхідних і вихідних даних шари, з'єднані через серію прихованих шарів [91]. Генеративні змагальні мережі (GAN) містять дві нейронні мережі, а саме генеративну та дискримінаційні мережі, які в основному спрямовані на вироблення синтетичних та високоякісних даних [92]. Вони працюють за принципом ігор min–max, де одна мережа намагається максимізувати функцію значення, і інша мережа хоче мінімізувати його.

Застосування GAN включають формулювання локалізації та пошук шляху в мережах на основі графів [93, 94]. У цих програмах генератор мережі модуль GAN

оцінює всі можливі шляхи, які існують між двома вузлами. Модуль дискримінатора GAN виконує оптимальну ідентифікацію шляху між вихідним і цільовим вузлом. GAN широко поширені використовується в асистентних системах для людей з обмеженими можливостями.

Хмарний аналіз даних: автори в [93, 94] запропонували хмарну аналітику даних для Інтернет речей. Для ефективного аналізу даних потрібна ієрархічна та розподілена модель для систем обробки даних. Цей метод реплікується на кількох віртуальних машинах, які є асоціальними з віддаленим хмарним центром обробки даних. Потім, без будь-яких попередніх знань про дані, хмарна аналітика може впоратися з проблемами динамічної обробки даних і масштабованості.

Автори в [95] розглядають зберігання величезних кількостей високошвидкісних і складні дані зондування, які генеруються сенсорними системами IoT. Ці роботи представляли злиття даних за допомогою ефективних методів курації, які інтегрують Інтернет речей і величезні дані сенсорів у віддаленому хмарі сервер, що дозволяє системі надавати ефективні послуги для додатків IoT. Різні питання, такі як масштабованість сенсорної мережі, очищення даних, стиснення даних, зберігання даних, аналіз даних і візуалізації даних, вирішувалися за допомогою курації даних сенсорів IoT в хмару [96]. Автори в [97] розглянули питання збору даних розподілених сенсорів IoT з обмеженими ресурсами.

Слід зазначити, що дані сенсора IoT складаються зі складних даних часових рядів, і, таким чином, вимагаються ефективні механізми аналізу даних. Автори в [98,99] представили аналітичну структуру даних, яка включала багатовимірну модель обробки, вибору та вилучення ознак. Автори в [100, 101] мали на меті зменшити передачу та обробку даних у віддаленому хмарі за допомогою ефективної аналітики даних на основі даних сенсора туману. У роботі стверджується, що завантаження даних сенсорів у віддалену хмару через Інтернет не додає ніякої цінності через складність основної мережі.

Автори в [102] обговорювали поширений характер сенсорних мереж IoT який генерував величезний обсяг сенсорних даних. У даних сенсорів виявили

надмірність даних, які значною мірою погіршили загальну продуктивність сенсорних мереж IoT. Система аналізу даних, яка спрямована при рекурсивному оновленні та адаптації до динамічних змін в системи IoT вирішує в деякій мірі проблему.

Аналіз даних на основі туману: автори в [103–105] представили дані сенсорів IoT на основі обробки рівня туману. Тут витягуються й обробляються функції сеньор даних для класифікації різних сигналів з використанням нейронної мережі. Отже, на основі результатів класифікації нейронних мереж ідентифікація подій і прийняття рішень виконуються на рівні туману. Автори в [106,107] запровадити механізм оркестровки контенту, який оцінює опортуністичний туманний вузол для розвантаження даних. Механізм враховує доступну затримку пропускної здатності, вартість та показники якості обслуговування визначити опортуністичний вузол туману. Крім того, модуль прийняття рішень уможливлює періодичні звіти з вузлів, які служать вхідними значеннями для аналітичного процесу в реальному часі, і обчислює коефіцієнти для підвищення якості обслуговування.

Аналіз даних на основі периферій: автори в [108-111] для цієї мети пропонують багато рішень, таких як стохастичні та регресійні моделі виконується на граничному рівні сенсорної мережі IoT. При цьому аналіз даних здійснюється на вузлах сенсорів низького рівня. Автори [112] представляють рішеннях на основі IoT з точки зору охорони здоров'я. Це дослідження має на меті надавати ефективні послуги в додатку Internet of Medical Things, використовуючи периферійні обчислення парадигма. Запропонували трирівневу архітектуру IoT, яка включає граничні пристрої, туманна мережа та хмарні сервери. Сенсори діють як граничні пристрої та підключаються до туманних мереж для аналіз даних у реальному часі. Для високопродуктивних обчислень і збільшення об’єму пам’яті ці краї пристрої підключені до хмарних віддалених серверів. Семантичний аналіз даних сенсорів IoT: збільшення кількості сенсорних даних виникає в результаті створення даних та програм, легкодоступних та зрозумілих майбутнім користувачам.

Висновки до розділу 1

Розглянуто поширені мультисенсорні інформаційно-вимірювальні системи. Проаналізовано існуючі методи зв’язку в інтелектуальних вимірювальних системах, а також підходи до інтелектуалізації вимірювальних систем. Розглянуто основні тенденції інтелектуалізації інформаційно-вимірювальних систем, основні характеристики бездротових сенсорних мереж, які потребують поглибленого дослідження методів очищення даних, керування талантами на основі технології інтелекту. За допомогою цього методу процеси керування талантами в бездротових сенсорних мережах можуть реалізовуватися самостійно. Вдосконалення методу очищення даних може вирішити невідповідність даних при ідентифікації одного і того ж об`єкту і підвищити точність розпізнавання.

Показано, що із зростанням різноманітності пристроїв, підключених до IoT, традиційна централізована мережева архітектура повинна відповідати новим вимогам до послуг, а також ефективно ідентифікувати та надавати великі обсяги даних щодо безпеки, цілісності та конфіденційності. Тому є важливим розроблення методів для очищення даних у таких мережах. Для вирішення згаданих проблем і досягнення кращої якості обслуговування шляхом виконання операцій зберігання та обробки даних фізично поблизу джерела даних у розподіленій інфраструктурі важливо використовувати Fog/Edge обчислення. Крім цього, розроблення методу довіри на основі туманних обчислень для запобігання втручання третіх сторін при встановлені довірчих відносин в мережі з мультисенсорною конфігурацією, експериментальне підтвердження даного методу дозволить відстежувати стан довіри всієї мережі, виявляти приховані атаки на дані та відновлювати вузли неправильної оцінки при менших затратах енергії.

Внаслідок проведеного літературного аналізу виявлено недостатньо інформації щодо динамічного пошуку помилок у промислових протоколах інтернету, тому актуальним є необхідність використання методу «нечіткого інтелекту» для оцінки мультисенсорних комп’ютеризованих систем з двох аспектів: нечітка методологія необхідна, як система, що базується на знаннях, для виявлення причини невизначеності; під час оцінки інтелектуальних

мультисенсорних систем прослідковується неоднозначність, що обумовлюється недосконалим виконанням.

Таким чином, для досягнення поставленої мети – на основі комплексних теоретичних та експериментальних досліджень необхідно виконати такі завдання:

- покращити метод валідації промислових протоколів IoT для використання в мультисенсорній системі керування;
- покращити кодове покриття тестових випадків IoT для використання в мультисенсорній системі керування;
- підвищити ймовірність виявлення аномалій у реалізації протоколу IoT для використання в мультисенсорній системі керування;
- провести поглиблене дослідження методів очищення даних, керування талантами в бездротових сенсорних мережах на основі технології інтелекту;
- розробити метод довіри на основі туманних обчислень для запобігання втручання третіх сторін при встановлені довірчих відносин в мережі з мультисенсорною конфігурацією, а також експериментально підтвердити, що даний метод дозволяє зменшити споживання енергії, відстежувати стан довіри всієї мережі, виявляти приховані атаки на дані та відновлювати вузли
- запропонувати власні структурні та архітектурні рішення для мультисенсорних систем.

РОЗДІЛ 2. АНАЛІЗ І ВИБІР МЕТОДІВ ТА ЗАСОБІВ ПЕРЕДАВАННЯ ТА ОПРАЦЮВАННЯ ІНФОРМАЦІЇ З МУЛЬТИСЕНСОРНИХ СИСТЕМ

Необхідність вимірювання фізичних параметрів відіграє важливу роль у науці та техніці. У наш час для цих цілей широко використовуються сенсори. Важливо зауважити, що сенсори мають широке застосування і в повсякденному житті. Сенсор перетворює фізичні параметри в електричні сигнали. Ці пристрої зазвичай поєднувалися з складними електронними системами або комп'ютерами для моніторингу та контролю найрізноманітніших параметрів життєвого середовища. Поява мікроконтролерів дозволила замінити складні електронні системи та призвела до розробки простих і економічно ефективних платформ - бездротових сенсорних мер (БСМ) [113].

Мікроконтролер містить центральний процесор (CPU), оперативну пам'ять (RAM), постійну пам'ять (ROM), таймери, лічильники, аналого-цифрові (A/D) перетворювачі, вхід/вихід (I/O), порти та інші периферійні пристрої на одному чіпі [114]. Застосування мікроконтролерів є численним, і вони варіюються від простих додатків, таких як іграшки, до складних, таких як електромережі, медичні програми, робототехніка тощо. Удосконалення технології їх виготовлення призвело до виробництва внутрішніх компонентів невеликих за розміром, який не тільки займає невелику площу, але й забезпечує низьку ефективність споживання електроенергії, значні швидкості передачі даних та незначні затрати на виготовлення.

Розробка комунікаційних модулів і протоколів значною мірою підтримує впровадження систем дистанційного моніторингу в реальному часі на основі мікроконтролерів. Простий характер протоколів дозволяє легко реалізувати їх за допомогою мікроконтролера. До цих комунікаційних модулів належать Wireless Fidelity (Wi-Fi), Bluetooth, ZigBee та Global System for Mobile Communication (GSM) [115]. Ці модулі мають як переваги так недоліки. Так Bluetooth, RF і ZigBee можуть передавати дані лише між тими самими модулями або на комп'ютер. Для реалізації системи дистанційного моніторингу дані модулі зв'язку не можуть бути використані, оскільки вони мають обмеження радіусу дії. Разом з цим Zigbee,

Bluetooth або GSM не мають можливості надсилати дані безпосередньо на інтернет-сервер.

Таким чином, для реалізації системи потрібен головний комп'ютер і програмне забезпечення. Як показано на рис. 2.1, сенсорні вузли надсилають дані на материнську плату, яка збирає дані з усіх вузлів та надсилає їх на комп'ютер користувача через універсальну послідовну шину (USB).

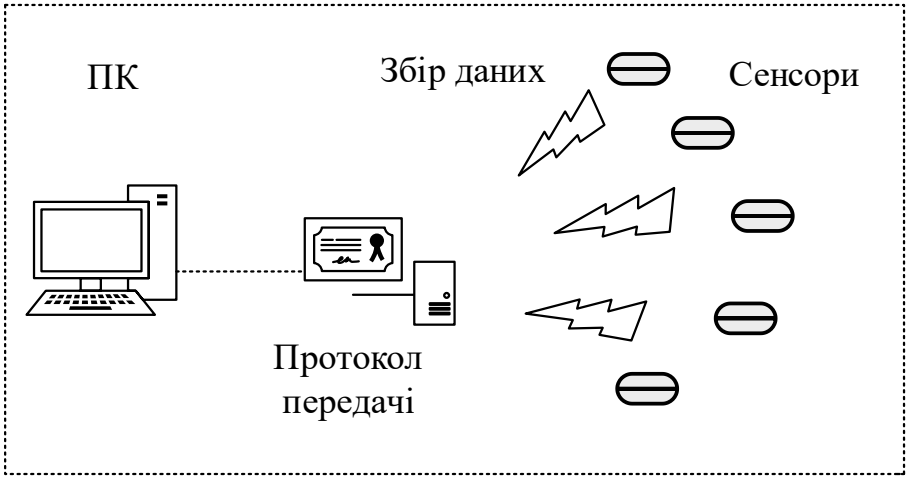


Рис. 2.1. Архітектура найпростішої бездротової сенсорної мережі [116]

За допомогою Інтернету дані можна надсилати в будь-яку частину світу. Тому використання Інтернету є найкращим варіантом для впровадження системи віддаленого моніторингу. Щоб надіслати дані на інтернет-сервер, потрібен комп'ютер разом із цими модулями зв'язку. За допомогою модулів *Wi-Fi* і *GSM* дані можна передавати безпосередньо на інтернет-сервер, з якого їх можна переглядати з будь-якої точки світу. Модулю GSM потрібна виділена лінія для передачі даних, тоді як модуль *Wi-Fi* можна використовувати з існуючою мережею або звичайним середовищем *Wi-Fi*. Завдяки високій швидкості передачі даних через *Wi-Fi*, передавати дані дуже легко. Це дозволяє легко реалізувати моніторинг у реальному часі. Отже, використання модуля *Wi-Fi* для передачі даних на інтернет-сервер є найбільш рентабельним, ефективним і застосовуваним методом у системі дистанційного моніторингу в реальному часі [117].

2.1. Методи мережного кодування

Як правило, кодування – це процес перетворення необроблених даних у форму, яку легко передавати, обробляти та зберігати. У зв‘язку з тим, що джерела інформації стають все об’ємнішими, проблеми передачі, обробки та зберігання постають все гостріше. Найбільшими джерелами даних є мережева інформація. До того ж значного розвитку досягли реляційні та нереляційні бази даних [116].

Обсяг доступних даних стає більшим, але в той же час з’являються одна за одною проблеми, найпоширенішими з яких є проблеми кодування даних. Кодування даних є важливим етапом опрацювання інформації в багатьох областях [117,118]. Для великого набору даних критерієм якості відповідних параметрів є точність отримуваних даних в процесі кодування.

Теорія кодування визначає три основні категорії кодування: вихідне кодування, канальне кодування та мережеве кодування [119-123]. Перший спосіб призначений для стиснення даних у джерелі, другий метод додає надлишкові біти в канал для підвищення надійності передачі даних. Третій тип кодування відбувається на проміжних мережевих вузлах і на різних рівнях мережевого стеку.

Сьогодні розвиток мережевого кодування в основному зосереджений на двох теоретичних підходах: лінійному кодуванні та китайській теоремі про залишки [124]. Фундаментальний принцип мережевого кодування полягає в тому, що проміжний вузол об’єднує кілька отриманих пакетів і передає їх наступним вузлам мережі [125]. Концепція мережевого кодування та обміну пакетами в бездротовій мережі з трьома вузлами проілюстрована на рис. 2.2. Вузли A і B спілкуються через вузол C (ретранслятор) (X_1, X_2).

Для бездротової мережі без кодування протокол передачі подібний до:

$$
\begin{aligned}
&t_1 \quad x_1 : A \to C;\\
&t_2 \quad x_2 : B \to C;\\
&t_3 \quad x_2 : C \to A;\\
&t_4 \quad x_1 : C \to B.
\end{aligned}
$$

У результаті для передачі повідомлень по мережі без кодування між вузлами A і B необхідні чотири кадри.

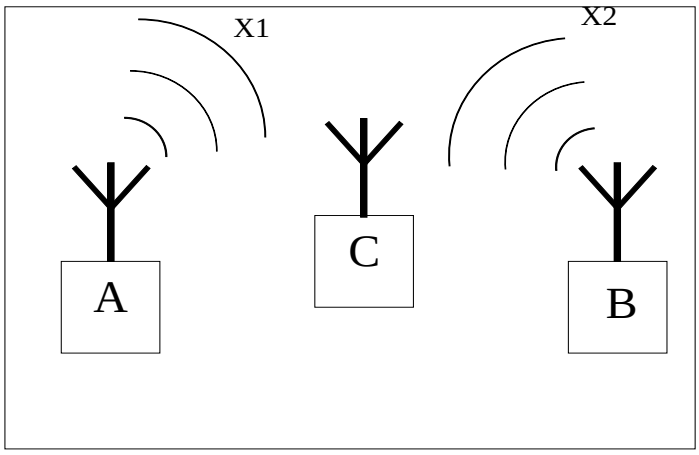


Рис. 2.2. Бездротова мережа трьох вузлів

Коли використовується мережеве кодування, повторювач зберігає пакети та створює їх лінійні комбінації. У цьому випадку протокол передачі буде схожий на такий:

$$t_1 - x_1 : A \to C;$$
$$t_2 - x_2 : B \to C;$$
$$t_3 - x_1 + x_2 : C \to A;\ C \to B.$$

В кадрі t_3 вузол *S* формує суму прийнятих пакетів за модулем та передає $x_1 + x_2$. Вузли А і В отримують повідомлення і обраховують необхідні пакети:

$$x_2 = (x_1 + x_2) + x_1 ;$$
$$x_1 = (x_1 + x_2) + x_2.$$

В результаті для передачі повідомлень між вузлами А і В необхідно 3 кадри.

Використання мережевого кодування дозволяє збільшити пропускну здатність і підвищити надійність мережі [127].

У мережевому кодуванні кожен переданий пакет через мережу є лінійною комбінацією пакетів, додавання виконується по полю $GF(2^s)$ [128]:

$$X = \sum_{i=1}^{n} g_i H^i ; \qquad (2.1)$$

де H^i – вихідні пакети одного або декількох джерел інформації; g_i – коефіцієнти поля $GF(2^s)$, символ g_i складається із *s* бітів.

Для того, щоб відновити закодовані пакети $(g_1, H^1), \ldots, (g_m, H^m)$, необхідно розв’язати систему *m* рівнянь;

$$\{X^j = \sum_{i=1}^{n} g_i H^i\}, j = \{1 \ldots m\} \quad (2.2)$$

в якій H^i є невідомими.

Щоб розв’язати систему з m рівнянь із n невідомими, це необхідно для того, щоб мати m > n. Однією з найбільш значущих завдань, пов'язаних з використанням мережевого кодування, є вибір лінійних комбінацій кожним вузлом мережі. Найпростіший алгоритм - кожен вузол має рівну, випадкову та незалежну ймовірність вибору елементів із поля. $GF(2^s)$. В [128] досліджуються різні методи вибору коефіцієнтів кодування та алгоритми декодування.

Методи лінійного мережевого кодування мають значний вплив на обсяг службових даних у пакеті протоколу, що означає, що вони не можуть бути використані в БСМ, які мають невеликі розміри пакетів і обмежену пропускну здатність. На відміну від методів лінійного мережевого кодування, метод мережевого кодування на основі китайської теореми про залишки зменшує обчислювальні витрати на кодування даних, а також обсяг службових даних у протоколах передачі даних [129].

У [128] обговорюються переваги мережевого кодування в системі залишкових класів при передачі даних «один до багатьох». У [129] пропонується використовувати набір взаємно простих модулів для кодування та передачі повідомлень на основі китайської теореми про залишки на прикладі багатоадресної розсилки для топології мережі «метелик».

У БСМ бездротові вузли зазвичай передають виміряні дані на базову станцію, що використовується за принципом «багато до одного». Проте методи збільшення пропускної здатності корисної інформації в БСМ при мережевому кодуванні в СЗК ще потребують вивчення. З огляду на все це залишається невирішеним питання розробки методів мережевого кодування в системі залишкових класів, що враховує особливості топології БСМ.

2.2. Перетворення та опрацювання даних в бездротових сенсорних мережах

Реалізація методів стиснення даних у БСМ дозволяє ефективно використовувати канали зв'язку з обмеженою смугою пропускання та зменшувати витрати енергії, пов'язані з передачею даних.

На сьогодні розроблена велика кількість методів та алгоритмів стиснення даних, серед яких найбільш поширеним є JPEG, який вже реалізовано в більшості мікросхем CMOS камер. Недоліком існуючих алгоритмів стиснення зображень є їх послідовний характер, це особливо помітно на пристроях з обмеженими ресурсами [130]. Кодування – це процес перетворення необроблених даних у форму, яку легко передавати, обробляти та зберігати. Кодування розглядається як перетворення пікселів зображення в систему постійних класів. При переході до представлення чисел у СЗК ми отримуємо числа, незалежні від малих розрядів (2-8 розрядів), це збільшує швидкість виконання арифметичних операцій. Кольорове зображення у форматі RGB візуалізується як набір кольорових пікселів,

$$M \times N \times 3, \qquad (2.3)$$

де M - кількість рядків, N - кількість стовпців, а кожен піксель - це 3-кортеж, компоненти якого відповідають кожному з трьох кольорів: червоний, зелений і синій.

Найпоширенішим підходом є представлення кожного компонента 8-бітним значенням, відповідно RGB-зображення має глибину 24 біти. Процес заснований на переході від представлення пікселів зображення в двійковій системі числення до їх представлення в системі залишкових класів [131]. Оскільки кількість пікселів у зображенні може приймати значення від 0 до 255, важливо вибирати в SZK модулі як прості, так і великі, які мають бути більше 255.

Отже, наступні набори модулів будуть обрані: {p1 =3, p2 =7, p3 =13}, де P= 3x7x13=273 або {p1 =5, p2 =7, p3 =8}, де P= 5x7x8=280. Для подальшої роботи оберемо модулі {5,7,8}, оскільки решта скорочених модулів у двійковому коді мають однакову швидкість передачі даних m=3. З цифрового фотоапарата

компоненти зображення RGB передаються в конвертер з двійкового коду в код СЗК, на виході якого ми отримуємо залишки. b_i від ділення значень пікселів на вибрану систему модулів p_i . Отримані масиви залишків поступають на кодер, який b_i , p_i потім виконує додаткову обробку (стиснення) візуального представлення залишків зображення за допомогою алгоритму арифметичного кодування [132]. З виходу кодерів стиснуті послідовності надходять на передавальні пристрої та передаються паралельними каналами [132].

2.3. Розумна архітектура моніторингу фізичних об’єктів у інформаційно-вимірювальних системах з мультисефнорною конфігурацією

Розглянемо загальний опису рівнів архітектури досліджуваної системи і кожен рівень цієї системи зокрема. БСМ в залежності від цільового призначення можуть використовуватися різні сенсори. Зокрема, для ведення спостережень, визначення температури, вологості, тиску, рівня шуму, визначення хімічного складу речовини чи повітря, наявності або відсутності певних видів об’єктів, руху, диму, виявлення швидкості, напрямку та розміру об'єкта, включена аудіо- та відеоінформація [133]. Архітектура фізичних об’єктів моніторингу базується на п’яти рівнях, як показано на рис. 2.3 [134]. Усі рівні пов’язані один з одним, щоб змінювати дані та задовольняти запити [135,136].

Рівень 1 - Фізичний рівень: це найнижчий рівень архітектури фізичних об’єктів, який означає певне середовище чи об’єкт з приладами.

Рівень 2 - Рівень сенсорів і виконавчих механізмів: він складається з: бездротових сенсорів, які визначають різні параметри в середовищі чи об’єкті, а саме температуру, вологість, тиск, освітленість, рівень шуму, задимленість, хімічний склад повітря, радіаційний фон, наявності або відсутності певних видів об’єктів, переміщення їх, визначення швидкості, напрямку і розміру об’єкту, аудіо- та відеодані і надсилають їх через мережу на локальний сервер (рівень керування даними). Виконавчі механізми, які виконують певні види керування, на основі запитів локального сервера.

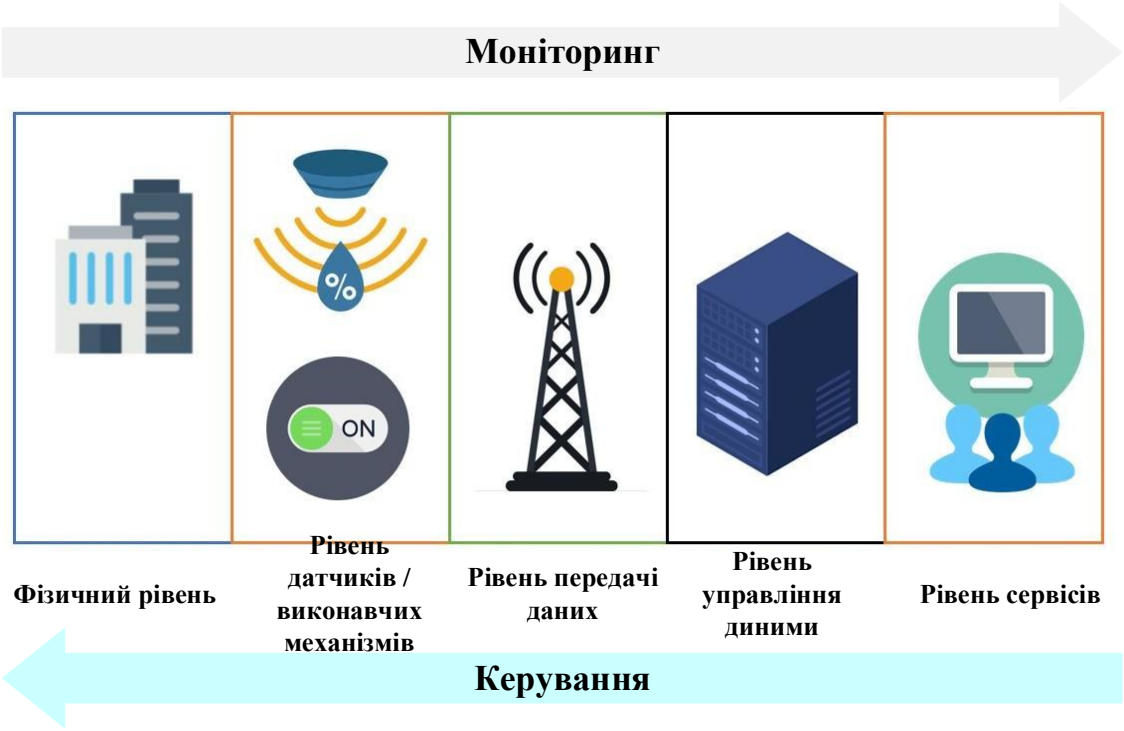


Рис. 2.3. Архітектура системи фізичних об’єктів моніторингу [137]

Рівень 3 - Рівень передачі даних: включає мережу та набір протоколів, які забезпечують зв'язок між сенсорами, виконавчими механізмами та локальним сервером, а також зв'язок між локальним сервером і хмарою.

Рівень 4 - Рівень управління даними: він є центральним елементом системи розумної будівлі. Він складається з локального сервера, який відповідає за зберігання, очищення та аналіз даних, надісланих сенсорами. На додаток до керування приводами.

Рівень 5 - Рівень сервісів: він визначає послуги для користувачів, такі як інформація про споживання, інформація про комфорт, безпеку, тощо та керування обладнанням.

Розрізняють два типи потоку даних: дані моніторингу і дані керування.

Дані моніторингу дозволяють контролювати вимірювальні параметри (комфорту, споживання, безпеки, тощо) фізичний об’єкт моніторингу (фізичний рівень) за допомогою сенсорів (рівень сенсорів/приводів), які надсилають дані через мережу. (Рівень передачі даних) на локальний сервер (Рівень керування даними), щоб їх аналізувати та зберігати. Інтерфейс користувача, розміщений на

локальному сервері, дозволяє кінцевим користувачам (рівень послуг) переглядати дані моніторингу в режимі реального часу. Сервісний рівень складається з інтерфейсу користувача та кінцевих користувачів (мешканці та менеджери).

Дані керування: дозволяють кінцевим користувачам керувати різними елементами, через інтерфейс користувача, який передає команду на локальний сервер. Локальний сервер визначає правильний ідентифікатор актуатора, потім транслює команду з ідентифікатором актуатора в мережу. Актуатор з тим же ідентифікатором виконує дію.

Фізичний рівень є першим рівнем системи розумної будівлі. Він складається з будівель та інфраструктури та забезпечує середовище для вивчення параметрів внутрішнього комфорту різних частин будівлі та моніторингу споживання мешканцями.

Рівень сенсорів і виконавчих механізмів спрямований на збір даних із навколишнього середовища за допомогою бездротових сенсорів і керування обладнанням будівлі за допомогою виконавчих механізмів. На рис. 2.4 показана архітектура цього рівня. Бездротовий сенсор складається з сенсорного блоку, який відповідає за фіксацію фізичної величини та перетворення її в цифрове значення, яке обробляється та зберігається блоком обробки та пам’яті. Коли дані готові, комунікаційний блок надсилає дані на локальний сервер і чекає на запити, які можуть бути надіслані користувачем, щоб змінити параметри сенсора, наприклад частоту передачі. Бездротовий привід складається з блоків керування, які виконують запити, що надходять від обробки для керування обладнанням будівлі [138].

Сенсори комфорту використовуються для кількісної оцінки комфорту середовища, а саме температури, вологості, хімічного складу повітря, освітленості тощо. Вони дозволяють досліджувати параметри комфортності у їх співвідношенні з іншими [141-147].

Рівень передачі даних забезпечує зв’язок між сенсорами та виконавчими механізмами та локальним сервером, а також зв’язок між хмарою та локальним сервером. Передача даних базується на протоколах малої дальності та дальньої дії.

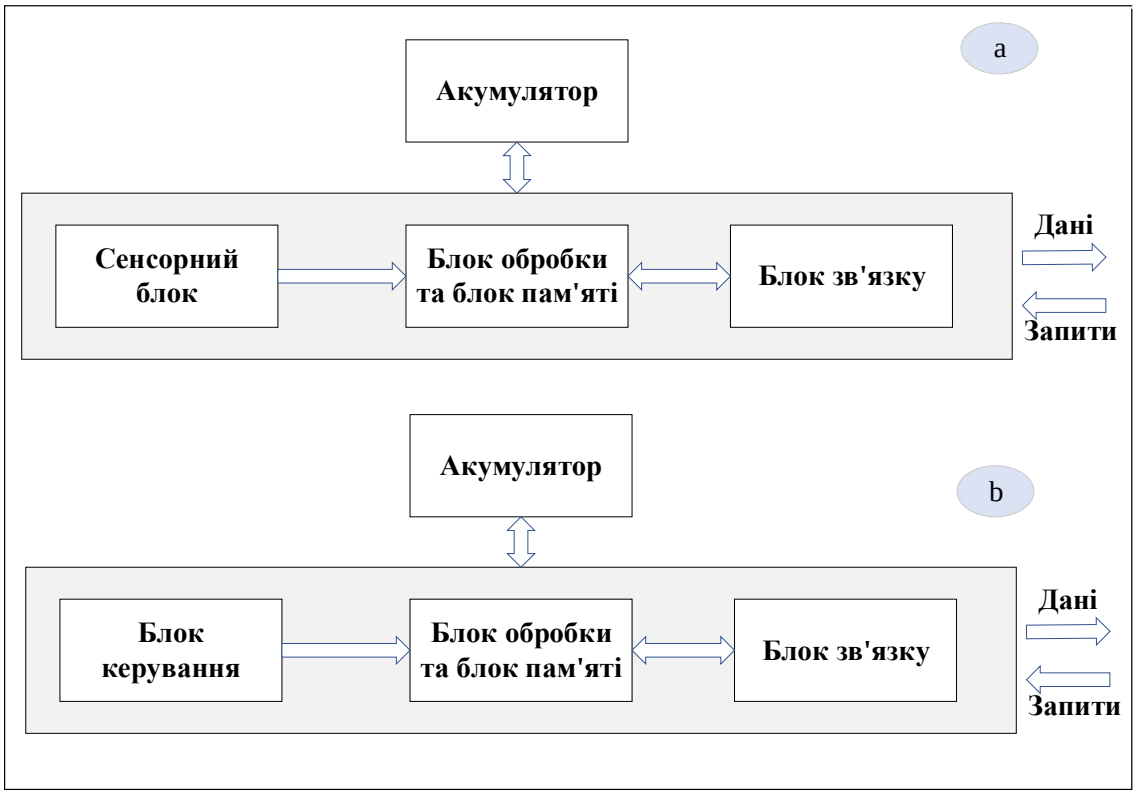


Рис. 2.4. Архітектура а) бездротового сенсора,
б) бездротового приводу сенсора [139]

Протоколи малого радіусу дії використовуються для зв’язку між сенсорами/виконавчими механізмами та локальним сервером, використовуючи низьке енергоспоживання та недорогі рішення. Використовуються такі протоколи:

- NFC : стандарт для безконтактного радіочастотного зв’язку на дуже короткій відстані (кілька сантиметрів), що забезпечує простий зв’язок між двома електронними пристроями (мітка та зчитувач). Кожен тег NFC має унікальний ідентифікатор і може містити невелику кількість даних.

- BLE: також відомий як Bluetooth Smart, це технологія зв’язку малого радіусу дії, яка використовує радіохвилі короткої довжини з мінімальною кількістю енергії. Він призначений для збору даних із сенсорів, які генерують дані з дуже низькою швидкістю.

- Z-wave: протокол бездротового зв’язку з низьким енергоспоживанням, призначений для пристроїв з живленням від батареї або електрики та широко використовується для розумних будівель, а також невеликих комерційних доменів.

- Wi-Fi: стандарт, який найчастіше використовується для бездротової локальної мережі (WLAN). Він поставляється з новим стандартом IEEE 802.11ah, який забезпечує більшу масштабованість, якість обслуговування та енергоефективність.

- Zigbee: технологія малого радіусу дії, що забезпечує низьке енергоспоживання, низьку складність і низьку вартість. Він використовує стандарт IEEE802.15.4 як фізичний рівень.

- EnOcean: протокол малого радіусу дії, низька складність і безпечний протокол, який використовується безбатарейними та бездротовими сенсорами.

- SWAP: легкий протокол з відкритим кодом і низьким споживанням. Використовується для зв'язку на короткій відстані.

Протоколи великої дії. Протоколи великої дальності використовуються для зв’язку між локальним сервером і хмарою або віддаленими сенсорами. Вони включають:

- LTE: протокол великої дії на основі мережі GSM/UMTS. Він охоплює швидкісні пристрої та надає послуги широкомовної та багатоадресної передачі. Він використовується для високошвидкісної передачі даних між мобільними телефонами.

- NB-IoT (вузькосмуговий Інтернет речей): широкозонне стільникове підключення для Інтернету речей, що забезпечує недороге рішення з низьким енергоспоживанням.

- Lora/LoraWane: бездротовий протокол великої дальності, який використовується в пристроях із довгостроковим живленням від батарей, де споживання енергії має першорядне значення. Він працює в багатьох діапазонах ISM залежно від регіону, в якому він розгорнутий, наприклад, діапазони ISM 433 МГц, 868 МГц або 915 МГц.

- Sigfox: телекомунікаційний оператор Інтернету Інтернету. Sigfox працює в діапазоні частот 868 МГц. Кінцевий пристрій (сенсори) може надсилати до 140 повідомлень на день із розміром корисного навантаження 12 октетів.

Рівень керування даними є основним рівнем архітектури розумної будівлі

та шлюзом між рівнем сенсорів/приводів і рівнем послуг. Він складається з локального сервера, який виконує наступні завдання: отримувати та зберігати дані, надіслані сенсорами; надсилати команди виконавчим механізмам; виявляти помилки сенсора; візуалізація даних; забезпечує зв'язок з центральним сервером.

На цьому рівні пропонується Raspberry Pi, що можна використовувати її як концентратор сенсорів (рис. 2.5) [148]. Raspberry Pi має переваги над ПК, що робить її ідеальним інструментом для взаємодії з широким асортиментом зовнішньої периферії. Подальше вивчення її ключових компонентів та характеристик дозволить оптимізувати роботу системи і покращити характеристики робото здатності системи на основі Raspberry Pi. Запропонована система заснована на протоколі SWAP і EnOcean. Зв'язок на великі відстані базується на мережі Sigfox, яка призначена для IoT. Він використовує мікроповідомлення (розмір: 12 байт) з інтервалом у 10 хвилин. Ця система буде використана в подальшому для забезпечення моніторингу в режимі роботи в реальному часі. Для цього в систему буде інтегровано більше типів сенсорів, таких як сенсор тиску, вологості та температури. Приведена система зарекомендувала себе як ефективний інструмент при проектування інформаційно-вимірювальної системи і покращення існуючих методів інтелектуалізації.

Програмну архітектуру реалізовано на локальному сервері. Він включає дві частини, базу даних та програмне забезпечення.

База даних є основним компонентом, який призначений для зберігання та отримання даних, зібраних із сенсорів. Основні типи баз даних:

- Реляційна база даних: база даних, яка містить елементи даних, пов’язані між собою заздалегідь визначеним чином. Ці компоненти розділені в набір таблиць, що складаються зі стовпців і рядків. Таблиці використовуються для зберігання інформації про об'єкти, які повинні бути представлені в базі даних.

- База даних NoSQL: це підхід до проектування бази даних, який може адаптувати широкий спектр моделей даних, включаючи формати з ключами, документами, стовпцями та діаграмами. Це особливо корисно для роботи з великими розподіленими наборами даних.

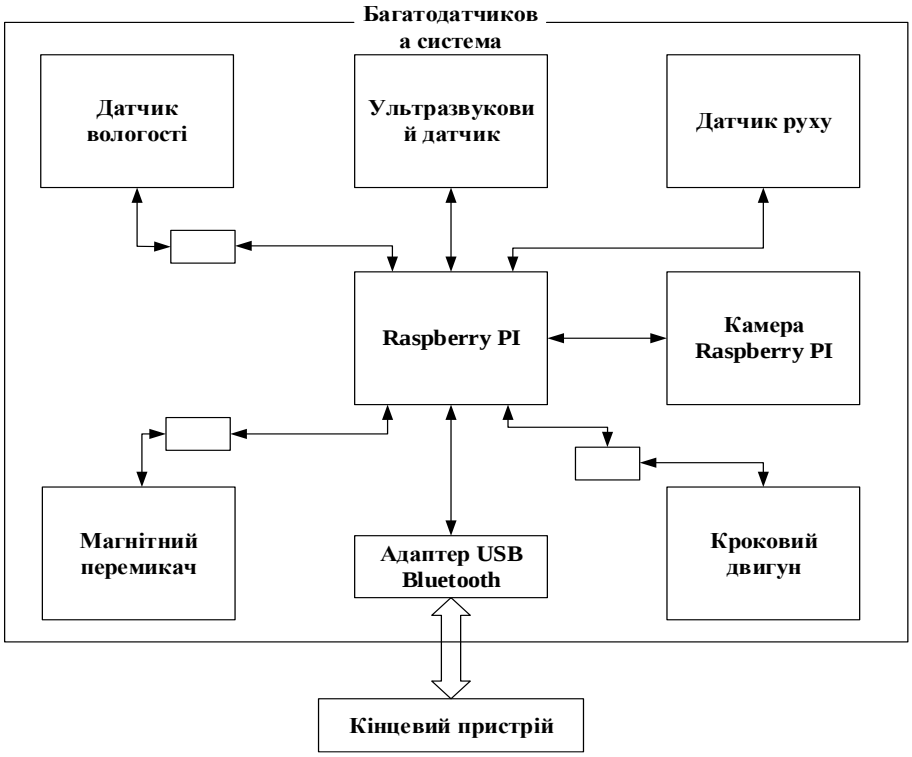


Рис. 2.5. Структурна схема інформаційно-вимірювальної системи [148]

Система заснована на реляційній базі даних, яка містить кілька таблиць даних. Ці таблиці з'єднані за допомогою спеціального ключа для організації даних у семантичній структурі, як показано на рис. 2.6.

Сервер локального сервера базується на середовищі з відкритим кодом nodejs, яке використовує javascript. Nodejs11 має високу продуктивність, масштабованість та асинхронне програмування, кероване подіями. Архітектура програмного забезпечення складається з модулів, кожен з яких має певну роль.

Сервісний рівень є верхнім рівнем архітектури. Він надає послуги користувачам (мешканцям і менеджерам), відповідає за візуалізацію даних, таких як зібрані (накопичені) дані та дані в реальному часі, і контроль обладнання будівлі. Крім того, він керує взаємодією між користувачами та локальним сервером.

Хмарна платформа Arduino IoT має рішення, яке спрощує створення мережевих проектів для розробників IoT.

HTTP REST API, MQTT, методики командного рядка, JavaScript і WebSocket, є деякі з методів взаємодії, які підтримуються платформою. Кілька пристроїв можна з'єднати разом і обмінюватися даними в режимі реального часу [150].

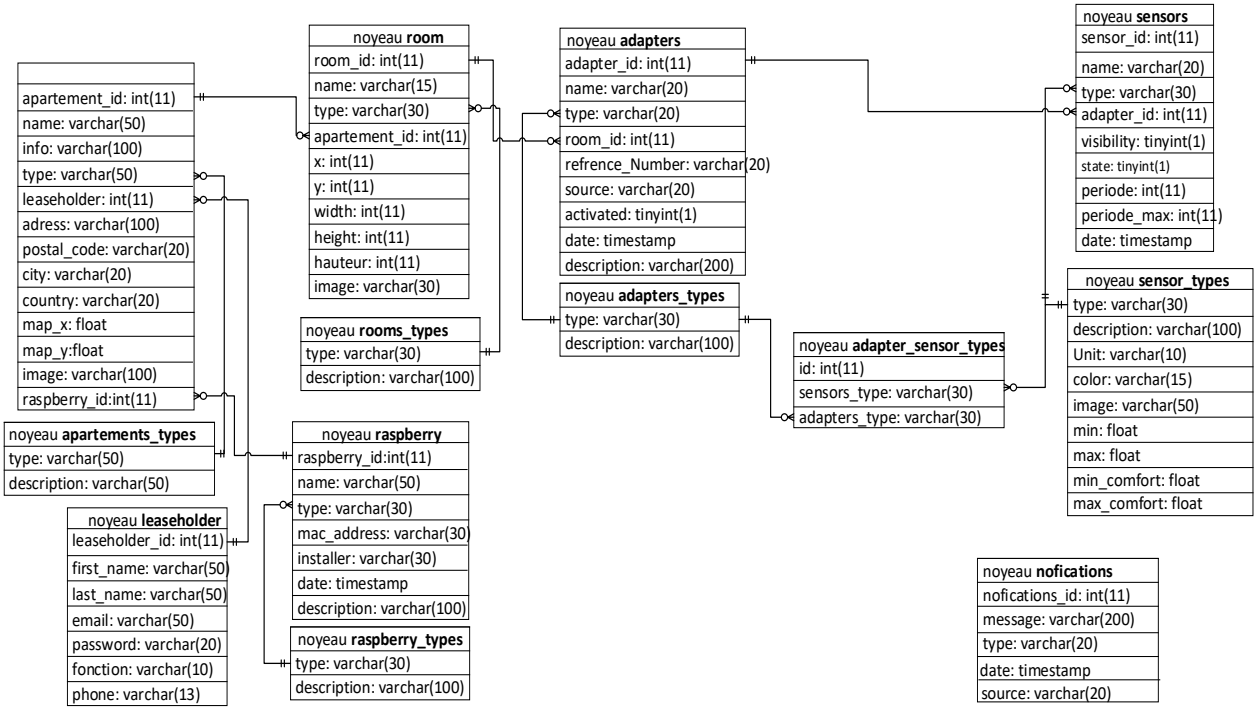


Рис. 2.6. Структура бази даних [149]

2.4. Методи та моделі статистичного моделювання для опрацювання експериментальних даних

Для обробки експериментальних даних використано систему програм періодичних випадкових процесів. При розробці цих систем виникла нова проблема їх тестування та оцінювання ефективності. Останнє виявилося складною задачею, яку довелося розв'язувати шляхом статистичного моделювання.

Як вказано у [151,152] під статистичним моделюванням розуміють відтворення за допомогою ЕОМ функціонування ймовірнісної моделі деякого об'єкту. Мета моделювання такого роду полягає в оцінюванні з його допомогою середніх ймовірнісних характеристик моделі об'єкту. Зазвичай це математичне сподівання величин, що характеризують систему, а також їх дисперсії та коваріації. У більш складних випадках оцінюються функції розподілу.

Існує багато робіт, присвячених моделюванню реалізацій випадкових величин, процесів і послідовностей з незалежними значеннями [152,153]. Виходячи з того, що математична модель не достатньо вивчена та мало відома в прикладному плані, виникає необхідність глибшого дослідження точносних характеристик та особливостей моделювання з використанням субгауссових випадкових

періодичних процесів [154, 155, 156, 157].

Розробка методів моделювання випадкових величин із заданим законом розподілу (наприклад, φ – серій [157] у випадку нестаціонарних процесів), що базується на класі субгауссових процесів, які в прикладних аспектах стосовно аналізу роботи ІВС ще повністю не вивчені. Очевидно, що для того, щоб виконати моделювання, треба мати певні початкові данні. За умови використання ЕОМ ми маємо справу із скінченними послідовностями. Скінченні послідовності випадкових величин задаються скінченновимірними функціями розподілу.

Моделювання інформаційних сигналів накладає певні обмеження на вибір функцій розподілу і на повноту їх опису. Як відомо, часто треба моделювати сигнали, коли нгемає в повному обсязі функцій розподілу вимірюваних сигналів (наприклад, невідомі деякі багатовимірні функції модельованого процесу). Тоді доводиться мати справу з процесами, для яких відомі лише якісь окремі характеристики.

В такому випадку, в першу чергу можуть бути використані конструктивні моделі процесів, наприклад, лінійні процеси або моментні функції, які можна отримати шляхом математичної формалізації опису структурної схеми роботи тієї чи іншої ІВС.

При оцінці точності моделювання випадкових величин та процесів методом довірчих інтервалів, а також при розв'язуванні різних задач пов'язаних з перевіркою шляхом моделювання та перевіркою статистичних гіпотез виникає задача оцінки величини ймовірності $p = P\{|\xi - \mathrm{M}\xi| < \varepsilon\}$, де 2ε – довжина довірчого інтервалу, а ξ – значення моделюючого процесу, які ще можуть залежати і від часу t або значень результатів вимірювання [152].

Щоб оперувати з наведеним вище виразом в ідеальному випадку треба мати функцію розподілу значень величини ξ і величину 2ε. На практиці використовують різного виду нерівності для оцінки величини p. При цьому найчастіше використовується нерівність Чебишева [152], яка дає оцінку знизу для величини p. Виходячи з цього, при використання нерівності Чебишева майже не накладаються ніякі апріорні обмеження на клас розподілів випадкової величини ξ, виникає ідея

розглянути задачу оцінки точності моделювання ввівши деякі неістотні з практичної точки зору обмеження, які б дали змогу уточнити оцінку, що отримана на базі нерівності Чебишева. Зокрема, дослідити цю задачу в класі періодичних субгауссових процесів. Слід зауважити, що для використання нерівності Чебишева необхідно знати дисперсію, що не завжди відома і існує лише для гільбертових випадкових величин і процесів за означенням [152]. Моделі, запропоновані у роботах [154, 155, 156, 157], були використані при моделюванні у даній роботі.

Висновки до розділу 2

Проаналізовано та здійснено вибір методів, засобів та методик передавання та опрацювання інформації з мультисенсорних систем як базу для використання у власних дослідженнях.

Предсталвнено методи мережного кодування, перетворення та опрацювання даних в безпровідних сенсорних мережах, розумну архітектуру моніторингу фізичних об’єктів у інформаційно-вимірювальних системах з мультисефнорною конфігурацією, методи та моделі статистичного моделювання для опрацювання експериментальних даних.

Представлено методи та засоби системи моніторингу і контролю в реальному часі на прикладі розумної будівлі. Система дозволяє надавати великий набір інтелектуальних послуг, таких як моніторинг комфорту в приміщенні. В розділі також охарактеризовані підходи до обробки експериментальних даних періодичних випадкових процесів.

РОЗДІЛ 3. РОЗРОБЛЕННЯ ТА ВДОСКОНАЛЕННЯ МЕТОДІВ ДИНАМІЧНОГО ПОШУКУ ТА ОЧИЩЕННЯ МЕРЕЖЕВИХ ДАНИХ ДЛЯ ІНТЕЛЕКТУАЛІЗАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ ПРОМИСЛОВОГО ПРИЗНАЧЕННЯ

3.1. Динамічний пошук помилок в промисловій системі інтернету

У розділі представлено теорію динамічної оцінки пошкоджень промислового Інтернет-протоколу, а також визначення необхідних даних для динамічного мультимодального зв’язку сенсора. Метод нечітких тестів у поєднанні з динамічною мультимодальною передачею даних відстежує виконання програми, знаходить поля введення, що впливають на умовні розгалуження за допомогою динамічного аналізу пошкоджень, і фіксує залежності умовних розгалужень, щоб належним чином контролювати генерацію граматики тестового прикладу, тим самим збільшуючи виконання коду глибокого рівня. Результати порівняльного експерименту демонструють, що метод має вплив на валідність тестів і швидкість покриття коду, крім того, він підвищує ймовірність виявлення недоліків у реалізації протоколу. Представлено результати перевірки методу тестування інформаційно промислових протоколів.

Виходячи з ідеї про необхідність «нечіткого інтелекту», для оцінки комп’ютеризованих систем були проведені необхідні дослідження. У цьому контексті вирішено проблему низького покриття коду через багаторазове виконання тестових прикладів одним і тим же шляхом, починаючи з рівня системної програми в реалізації промислових протоколів Інтернету, або передумова мати доступ до вихідного коду програми або двійкового виконуваного файлу. Описано процедуру, яка поєднується з динамічною передачею модальностей кількох сенсорів у часі в програмі нечіткої обробки. Процес забезпечує виконання програм, реалізацію протоколу та визначає поля, які впливають на умовне розгалуження, використовуючи динамічне виявлення невідповідності, і фіксує зв'язок залежності між умовними розгалуженнями, щоб керувати створенням тестів.

3.1.1. Структура промислової системи керування Інтернетом

Промислова система інтернету (ПСІ) була реалізована за допомогою різноманітних компонентів автоматизації для отримання даних, контролю, моніторингу та інших функцій. Типова промислова архітектура Інтернет-комунікацій, як правило, складається з трирівневої структури, від високої до низької, відповідно, мережі підприємства, мережі моніторингу та мережі системи керування [161,162]. ПСІ відноситься до системи, що складається з комп'ютерного обладнання та блоку управління промисловим виробництвом, який включає SCADA [163]. На рис. 3.1. зображено типову архітектуру промислової системи керування Інтернетом.

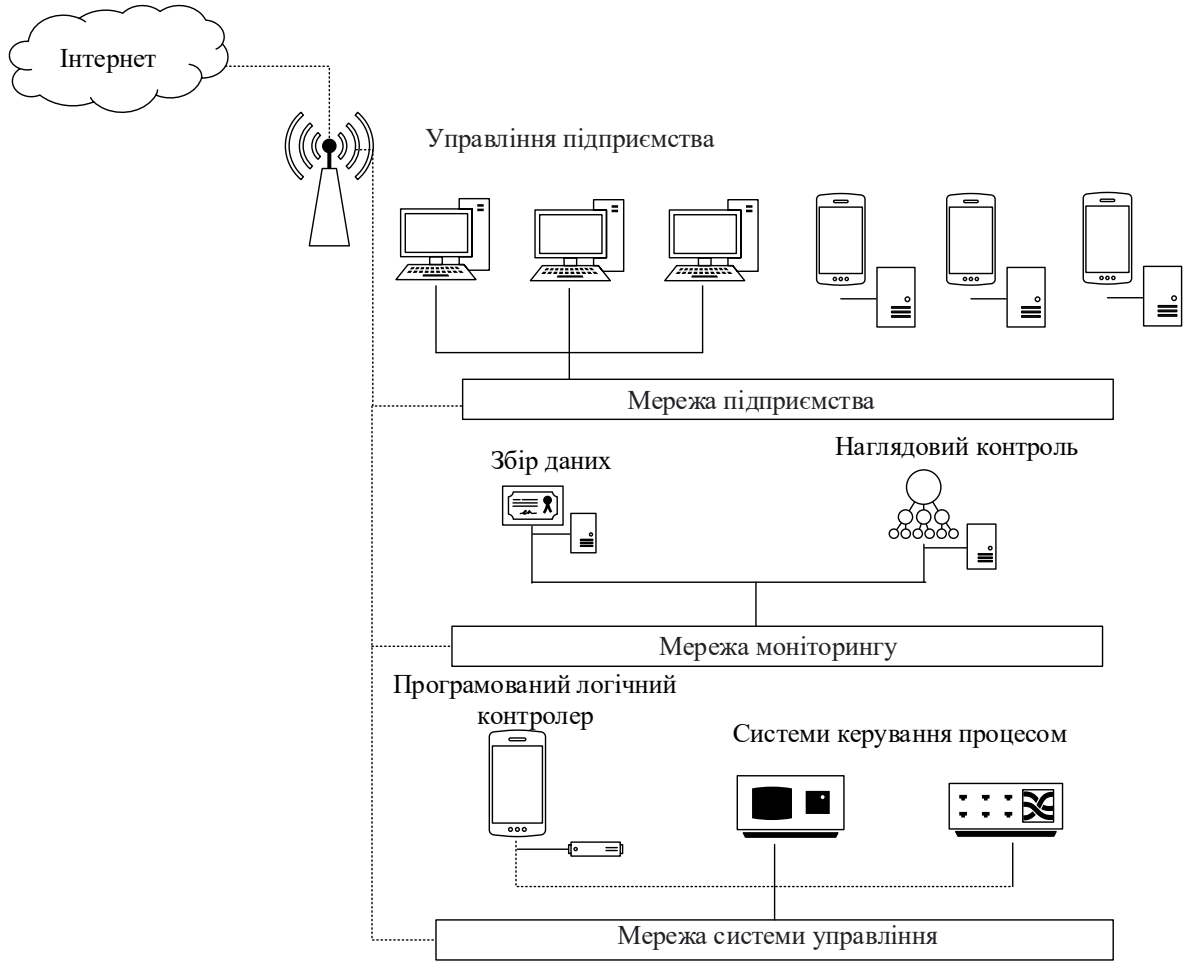


Рис. 3.1. Типова архітектура промислової системи керування Інтернетом

В електричному ПСІ кожна ланка (від виробництва електроенергії до її використання) має відповідний електричний термінал ПСІ, такий як (ПЛК)

програмований логічний контролер, інтелектуальне обладнання підстанції, пристрої моніторингу і керування та інші типи обладнання збору даних, генерування інструкцій, дистанційного керування тощо. Реалізується передача потоку даних та управління між терміналом ПСІ і головною станцією через промислові мережі Інтернету. Компонент програми, який працює в терміналі, відповідає за аналіз та обробку промислових Інтернет-протоколів.

За останні роки в умовах швидкого розвитку різноманітних інформаційних технологій індустріалізація та інформатизація тісно інтегруються. Сучасні інформаційні технології були застосовані до традиційних ПСІ. При цьому різноманітні стандартизовані протоколи зв'язку та архітектури комутації мережі популяризуються в ПСІ. Завдяки доданню передових інформаційних технологій та технологій комунікаційної мережі (таких як Ethernet), відкритість промислових систем керування Інтернетом була значно розширена, а також піддалася більшим ризикам безпеки.

У 2015 році українську електроенергетику атакувало шкідливе програмне забезпечення Black Energy [164], яке отримало віддалений доступ до керування системою, що спричинило збій хост-системи SCADA електромережі, а потім відключення електроенергії на великій території. У 2017 році виробник безпеки ESET анонсував інструмент win32/Industroyer, який безпосередньо атакує електричні ПСІ. Інструмент може викликати відключення трансформаторної підстанції, керуючи вимикачем. У 2018 році на віддаленій електричній платформі моніторингу ПСІ в Китаї з’явилося багато попереджувальних повідомлень про незаконний доступ до інтернету. Після аналізу з’ясувалася причина, яка полягає в тому, що виробник запускав і обслуговував сервер продукту віддалено, відкриваючи функцію обміну файлами, і таким чином надавав доступ до системи в загальнодоступній мережі на тривалий час, що спричиняло серйозні приховані небезпеки. Протягом останніх 10 років багато інцидентів з безпекою, потенційних і можливих небезпек, зробили ситуацію з безпекою ПСІ більш серйозною [165-167].

Під час реалізації протоколів промисловим Інтернетом термінальні коди, протокол поля, як правило, заслуговують на довіру, але зловмисник може контролювати виконання програми змінюючи значення даних полів за допомогою дефектів протоколів, які потім впливають на всю систему. Наприклад, параметри адреси призначення команди пропуску зазвичай надходять із надійних джерел даних у програмі, а не із зовнішніх недостовірних вхідних даних, таких як вхідні дані з промислового протоколу Інтернету. Однак зловмисник може перезаписати адресу призначення інструкції через уразливість системи, а потім контролювати запущений процес ПСІ. Практичні дослідження показують, що експерти не є досконалими і існують методи, які дозволяють нечітким системам моделювати продуктивність на рівні людини, включаючи мінливість.

Виникає необхідність «нечіткого інтелекту» для оцінки комп'ютеризованих систем з двох аспектів: (а) нечітка методологія необхідна, як система, заснована на знаннях, для представлення та причини невизначеності; (б) під час оцінки інтелектуальних систем необхідна неоднозначність і визнається недосконалим виконанням [168].

Таким чином, покращання методу валідності і охоплення кодом тестових випадків і підвищення ймовірність виявлення аномалій у реалізації протоколу ПСІ є важливим.

3.1.2. Аналіз безпеки протоколів промислового контролю

Промислова система контролю полягає у безпосередньому спілкуванні з базовим обладнанням або перетворювачем збору даних з використанням протоколу, погодженого сторонами, які спілкуються. Спочатку було впроваджено більшість протоколів промислового контролю тільки між закритою мережею та надійним програмним забезпеченням через виділений послідовний порт. Але для того, щоб задовольнити дедалі складніші вимоги промислового контролю систем, виділені лінії поступово замінюються TCP або бездротовими каналами. На початку проектування протоколи промислового контролю не враховували повністю необхідні умови для захисту безпеки користувачів, такі як шифрування та

автентифікація, і багато протоколів наразі покладаються на TCP/IP. У цьому випадку дані, що передаються через протокол, не можуть бути гарантовані від небезпеки. Зловмиснику потрібно тільки освоїти специфікації протоколу та проникнути в мережу промислового управління щоб підробити будь-які дані цільового пристрою.

Загальні недоліки безпеки. Протоколи промислового контролю аналізуються наступним чином:

1. Протокол Modbus не має механізму автентифікації. Встановлюється зв'язок на основі TCP/IP. Тому до тих пір, поки зловмисник отримує мережевий IP-адрес пристрою, він може успішно підключитися безпосередньо за допомогою порту 502. Якщо функціональний код, що передається блоком даних програми, підтримується Modbus пристрою, можна встановити легальний сеанс Modbus. Крім того, немає жодного повідомлення перевірити протокол Modbus/TCP. Контрольна сума формується на транспортному рівні, не на програмному, тому команду просто підробити. В той же час для будь-кого, якщо вони можуть підключитися до цільового пристрою Modbus, зловмисники можуть виконувати функції пристрою Modbus без дозволу. Крім того, дані, інкапсульовані в Modbus, передають в чистому вигляді текст, і зловмисник може отримати дані повідомлення за допомогою мережевого пакета інструмент захоплення. Нарешті, найнебезпечнішим аспектом Modbus є його програмовість. Зловмисники можуть впроваджувати шкідливий код в RTU або PLC, щоб отримати контроль.

2. DNP3 - це стандарт, розроблений IEEE PES на основі IEC. Його безпека виражається у відсутності механізмів авторизації та шифрування, і його коди функцій і типи даних були чітко визначені, що полегшує роботу зловмисників які мають намір втручатися в сеанс DNP3.

3. Ethernet/IP є більш сучасним, ніж Modbus, але все ще є проблеми з безпекою. До прикладу, при використанні протоколу UDP для трансляції даних в реальному часі в ньому відсутня вбудована мережа механізму рівня для забезпечення надійності зв'язку та цілісності даних. Нападники може легко вводити

неправдиві дані або використовувати керуючі повідомлення IGMP для маніпулювання шляхами передачі.

3.1.3. Метод fuzz-тестування промислового Інтернет-протоколу на основі динамічного аналізу

З огляду на серйозні збитки, завдані вразливістю ПСІ, дослідники запропонували багато технологій виявлення вразливостей, таких як динамічний аналіз, символічне виконання та fuzz-тест [169, 170]. Порівняно з іншими технологіями, fuzz-тест вимагає лише невеликих знань про ціль, але його можна легко розширити до великої прикладної програми з гарною можливістю повторного використання і, стати найпопулярнішим рішенням для виявлення вразливостей на даний момент, особливо в IIS. Що стосується мережевих протоколів, то найбільш репрезентативним фазифікатором є SPIKE [171]. Принцип полягає в описі протоколу як моделі послідовності блоків і створенні даних змін в блоках, а потім розділення структури даних повідомлення і автоматичне створення статистики довжини поля після зміни, таким чином значно покращуючи ефективність тестових випадків. Але принцип нечіткий і показує недостатню описову силу для обмеженого зв’язку в повідомленнях протоколу. Пізніше Zalewski F [172] і Peach [173] розширили модель даних, засновану на SPIKE, і додали більше описів зв'язку залежності між блоками даних. Щоб забезпечити більш гнучку та точну нечітку структуру, AFL [174] відстежував покриття шляху кожного входу за допомогою полегшеного інструментарію у вихідній програмі та випадковим чином на шляху розподіляв ідентифікатор базовим блокам, використовуючи механізм хешування, щоб визначити, де знаходиться будь-який згенерований новий шлях, а потім використовував вхідні дані, які генерують новий шлях, як початковий. У поєднанні з детальною інформацією в програмі, метод покращує швидкість охоплення коду, але метод Hash може легко мати випадки зіткнень і, таким чином, викликати проблему помилкових негативних результатів, навіть якщо вхідні дані досягли нового шляху. Gan та ін. запропонував CollAFL [175], який розподіляв значення ідентифікатора кожному базовому блоку за

допомогою жадібного алгоритму та інших методів, щоб забезпечити різницю значення хешування на кожній стороні, і таким чином уникнути зіткнення хешування та реалізовувати більш точне уявлення про покриття шляху.

Тестовий фреймворк компанії BlackPeer [176], ключ до створення аномальних даних це, за умови наведених вихідних зразків даних, визначення протокольних повідомлень за допомогою розширеної форми Бекуса-Наура та створення відповідної граматики тестових даних. Michael Toecker та ін. [177] здійснили fuzz-тест для верхнього комп'ютерного програмного забезпечення IIS, конструюючи аномальні дані шляхом виконання операції зміни відповідних повідомлень і збереження стану зв'язку протоколу шляхом підробки контрольних кодів і значень підрахунку. SecuriTeam додала протокол DNP3 в інструмент тестування bestorm [178] і виявила вразливість атаки відмови в обслуговуванні, характерну для DNP3 під час fuzz-тесту до Wireshark.

Отже існують проблеми у застосуванні методів нечіткого тестування до промислових протоколів Інтернету:

а) *Низьке покриття коду*: Багато помилок можуть виникати у випадку великого охоплення шляху. Хоча деякі методи використовують динамічний мультимодальний сенсорний зв'язок, дані, оброблені в програмі, не відображаються при формуванні тесту випадків безпосередньо, через що більшість випадків виконується багаторазово за одним і тим же шляхом, як вхідні дані і охоплює лише кілька шляхів.

б) *Немає уніфікованих моделей опису*. Протоколи, навіть одного типу, мають різні форми повідомлень, але поточні методи вимагають створення різних моделей даних, що збільшує навантаження на будівництво моделі.

с) *Недостатня відповідність тестових випадків.* Тестовим випадкам важко пройти перевірку програми, що спричиняє занадто багато недійсних тестів. Варіаційний дизайн стратегії, не враховуючи характеристики промислових Інтернет-протоколів, повне врахування може призвести до надмірності випадку у випадку великої кількості вибірки даних, а потім впливати на ефективність тестування.

3.1.4. Динамічний аналіз

Як зазначалося вище, динамічний аналіз забруднень, запропонований Newsome J., Song J. [179], є методом відстеження інформаційного потоку під час виконання програми, тобто відстеження передачі слотів даних, які підлягають аналізу в системі і отримання детального процесу обробки даних цільовою програмою. У запропонованому методі ми використовуємо технологію отримання динамічних даних мультимодального сенсора в програмі, щоб забезпечити основу для подальшого тесту. Метод динамічного аналізу невідповідностей включає дві частини, а саме ідентифікацію даних забруднення та моніторинг шляху передачі пошкоджень [180]. Основною частиною ідентифікації даних пошкоджень є визначення даних про невідповідність. Якщо джерело даних є підозрілим, генеровані ним дані є пошкоджені дані, які, як вхідні дані, повинні відстежуватися та аналізуватися в запущеному процесі двійкового коду, а потім дані повідомлень, сконструйовані в fuzz-тесті, можуть розглядатися як підозрілі.

Під час виконання програми передача даних про пошкодження завершується, інструкції та дані можуть брати участь в операції як вихідні операнди інструкції арифметичної операції або параметри інструкції переміщення даних, вихідні дані якої, як правило, пов'язані з даними відстеження пошкоджень і, таким чином, повинні бути ідентифіковані як атрибут невідповідності. На рисунку 1 показаний простий приклад процесу динамічного аналізу. На рис. 3.2 a_1 і a_2 є даними пошкодження; стрілка означає виконання операції; провідний кінець представляє вихідний операнд; і хвостовий кінець представляє вихід операції. Під час виконання програми на змінні V_1~V_8 впливають дані пошкодження a_1 і a_2. Якщо A, як набір джерела пошкодження для кожної змінної, порожній, ми можемо вважати, що змінна не зіпсована. На рис. 3.2 видно, що набір джерела пошкодження V_8, який є останнім виходом, є об’єднанням наборів джерел спотворення операндів V_3 і V_6.

Використовуючи метод динамічного пошкодження, ми прагнемо аналізувати потік даних і потік керування, також відомий як явний потік і неявний потік, під час виконання двійкових кодів.

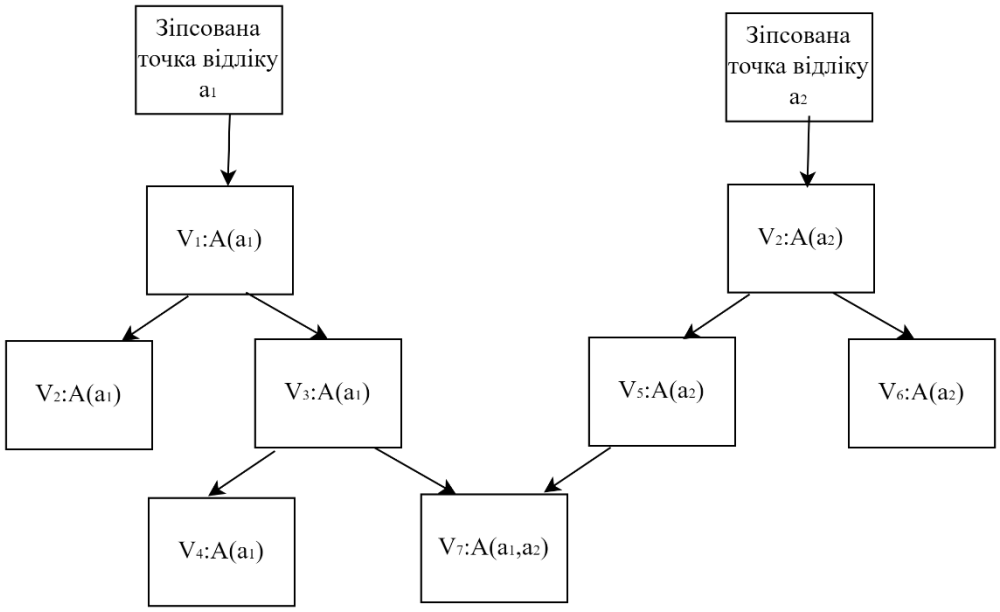


Рис. 3.2. Приклад динамічного забруднення

Перший стосується відношення залежності даних, тобто інформація про пошкодження змінної V_1 надсилається до V_2 безпосередньо за допомогою присвоєння або арифметичної операції, як показано на рис 3.2; зазначене відповідає зв'язку управління та залежності умовних гілок, тобто інформація про помилки змінної V_1 надсилається до V_2 опосередковано через пов'язаний вираз умови. З метою розуміння ми зробили простий аналіз, використовуючи приклад зразка коду, показаного на рис. 3.3.

У наведеному раніше прикладі функція реалізації коду переживає введення змінної *x*, генерацію повідомлення *msg* та передачу через повідомлення. У цьому процесі *x* ідентифікується як опорна точка, а відповідно до методу аналізу потоку даних, «*msg*» безпосередньо призначається як константа «a» або «b». Константа не буде зіпсована, тому «*msg*» ідентифікується як атрибут *untainted* (не пошкоджений). Однак значення «*msg*» також залежить від того, чи є умовні гілки *x*==a і *x*==b істинними чи хибними, тобто «*msg*» і *x* мають зв'язок контролю та залежності, що належить до неявного дефекту потоку. У практичному застосуванні такі коди мають ризик безпеки, коли вони використовуються для комунікаційної реалізації мережевих протоколів. Коли надіслане повідомлення, зловмисник може перехопити повідомлення, а потім вивести значення вхідних даних *x*, тому

повідомлення, безумовно, є опорною точкою, відтак, його слід фіксувати та контролювати. Будуть помилкові негативи без урахування зв'язку контролю та залежності повідомлення. Навпаки, значення *url* не залежить від гілок L_3 і L_7, тому немає ніякої шкоди, навіть якщо вона ідентифікована як незаплямована дана.

```
1.  begin
2. L1[ = get_input( ):
3. L2msg =uri = ` `:
4. L2 if (x ==`a`) {
5. L4  uri = `post` :
6.L5 msg = `a` :
7. L6}
8. L7 else if (x == `b`){
9.L8 urs = `post`:
10. L9  msg =`b` :
11. L10}
12. L11 send(uri, msg):
13. end
```

Рис. 3.3. Лістинг коду динамічного аналізу

Відповідно до програми реалізації промислового Інтернет-протоколу, отримуються відповідні динамічні мультимодальні дані комунікаційних сенсорів за допомогою динамічного аналізу невідповідностей, щоб направляти створення тестових випадків. На рис. 3.4 показаний конкретний процес тестування методу.

Запропонований метод використовує багато протокольних повідомлень, як вхідні дані, щоб уникнути проблеми недостатніх тестових випадків, викликаних єдиним зразком даних.

Динамічні інтерактивні поля: При виконанні даної програми, потім для умовної гілки x_i (і-е виконання умовної гілки x), існує $DIF(x_i) = \{F_j|F_j$ - це поле протоколу, що впливає на виконання $x_i\}$, у якому $DIF(x_i)$ - це набір поля протоколу.

Взаємозв'язок залежності та управління: під час виконання даної програми, тоді, якщо існує умовна гілка y_i, яка вирішує, чи виконувати *xi*, ми можемо сказати хі залежить від динамічного керування y_i і виражає його як $CDC(x_i) = \{y_i,T|F\}$, у

якому CDC(x_i) - це набір гілок, що відповідають умові, а обмеження (x_i) представляє обмеження умовної гілки x_i.

Блок-схема динамічного керування: для кожного введення програми, її виконання шлях може бути виражений за допомогою блок-схеми динамічного управління, в якій умовні гілка x_i — вузол, DIF(x_i) — динамічне інтерактивне поле вузла, а CDC(x_i) — це сторона, що представляє відношення залежності умовних гілок.

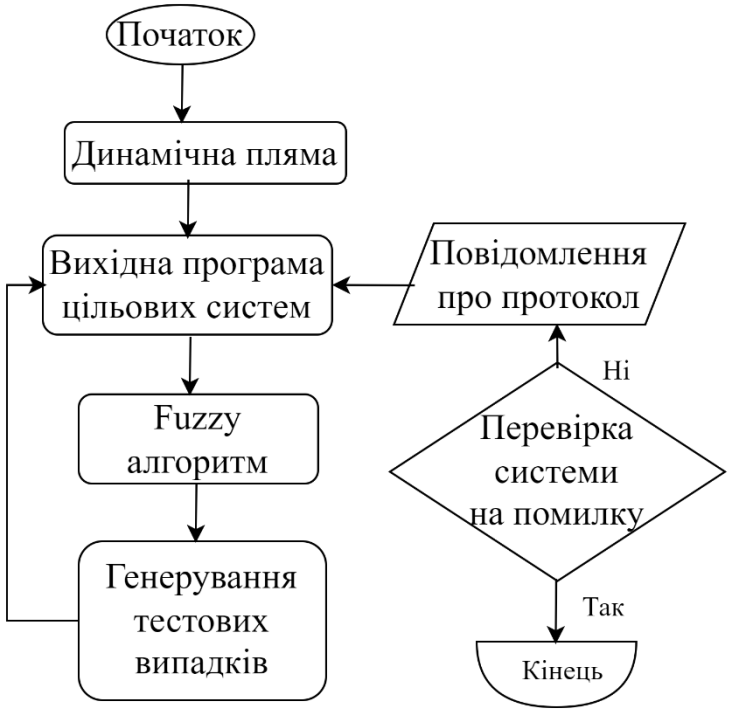


Рис. 3.4. Процес тестування промислового Інтернет-протоколу

Ми визначаємо поля вхідного протоколу, що впливають на умовні гілки через динамічний аналіз пошкоджень, виконуємо обробку ідентифікації пошкоджень для кожного входу протоколу поля та відстежуємо невідповідний потік даних під час виконання програми.

Що стосується залежності та взаємозв'язку управління програмою, то ми фіксуємо її за допомогою алгоритму, запропонованого в. Слід зазначити, що для промислових протоколів Інтернету поля, як правило, мають контрольну суму (наприклад, CRC), яка спричинить потік пошкоджених даних, якщо вони передаються в потоці керуючої інформації. Оскільки всі поля використовуються як контрольні суми, більшість умовних гілок буде ідентифіковано як пошкоджені

дані, і в цьому випадку важко безпосередньо знайти конкретне поле, яке впливає на умовні гілки. Крім того, через залежність та взаємозв'язок керування запропонований нечіткий метод непрямо розглядає порушення передачі в потоці керування. Тому, використовуючи метод динамічного аналізу невідповідностей, ми зосереджуємось лише на пошкоджені передачі в потоці даних, а не на відстеженні пошкодження передачі в потоці керування.

На рис. 3.5 зображено алгоритм 1 детально вводить нечіткий метод.

Основна функція *dynamic Fuzz* може виконувати програму *P*, протокол *G* та протокольне повідомлення I як вхідні дані, щоб обробити велику кількість вхідних даних протоколу. Рядки 1~2 алгоритму ініціалізують структуру даних, яка буде збережена. Тест випадки, які також будуть використовуватися як новий вхід, поміщаються в чергу для сховище для зручності виконання рекурсії.

Програма та вхідний протокол отримує набори *DIF*(x_i) і *CDC*(x_i) кожного умовного відділення в кожній програмі з використанням методу динамічного аналізу плям, і потім будує відповідну блок-схему динамічного керування, використовуючи *DIF*(x_i) as вузол і *CDC*(x_i) як сторону.

Рядок 6 виконує нечітку операцію з самого початку. Рядок 8 виводить обмеження (x_i), відповідну умову обмеження вузла x_i, як параметр тестового випадку алгоритм генерації.

Рядок 16 алгоритму означає, що після генерації тестового прикладу необхідно використати тестовий приклад як новий вхід для заміни значення поля протоколу в наборі *DIF* (x_i) оригіналу ввести та змінити усі поля, пов'язані з *Constraint* (x_i), щоб відповідати умові обмеження. Щоб переконатися, що тестовий приклад буде виконуватися на більш глибокому рівні в програмі та покращувати ймовірність пошуку нових шляхів виконання, решта полів протоколу, які не мають відношення до *DIF*(x_i) і *Constraint* (x_i), також повинні залишатися дійсними відповідно до протоколу граматики.

Наприклад, у протоколі введення значення двох полів початкова адреса і кінцева адреса має змінитися з початкових 1 і 2 на 3 і 6. Навіть якщо немає умови обмеження, відповідні поля адресних даних мають бути відновлені на основі

початкового нормального розміру (збільшення від 1 до 3), але, якщо дві адреси змінюються на недійсні, тестові випадки 6 і 3 після фазифікації (початкова адреса < кінцева адреса), поля, не пов'язані з *DIF*(x_i) і *Constraint* (x_i), все ще зберігають нормальні значення.

Алгоритм 1. Алгоритм нечіткого тесту в поєднанні з
Динамічні мультимодальні дані зв'язку з датчиків
Вхідні дані: Програма P, граматика протоколу
G та повідомлення протоколу I
Вихідні дані: Аномальна інформація
Функція dynamic Fuzz (***P,G,I***)

```
1. probedPath.inputQueue, checklist  ←  empty ( )
2. inputQueue.push(1)
3. while inputQueue.notEmpty( ) do
4.   input ←  inputQueue. pop ( )
5.  defg ←  execution Analysis ( P, I )
6.  node ←  defg.start(probedPath)
7.  while node ≠  NULL do
8.  C ← defg.gelConstraint (node.CDC)
9. if cheeklist.find(node.DIF, c)&&
10. node.true,node,false ≠  NULL. them
11. node ← defg.next( )
12. continue
13. end
14. tclist ← make TestCases (node.DIF, c , G)
15. for each te in icList do
16. input` ← makeInput ( input, te. value, c)
17. res ← executionMonitor (P< input`)
18. if (res.execption) then
19. return getExeptionInfo(res)
20. end
21.  if(probetPath.isNew(res.pathInfo)) then
22. inputQueue.push(input`)
23.   end
24.    end
25. cheeklist.add(node. DIF<c)
26. nodt ← defg.next( )
27. end
28. end
29. probedPath.add(defg)
```

Рис. 3.5. Алгоритм 1. Алгоритм Fuzz test у поєднанні з динамічними мультимодальними даними сенсора

Рядки 17~20 відстежують, чи є якийсь ненормальний стан, такий як збій або витік пам'яті під час виконання тестового прикладу. Незважаючи на те, що тестовий приклад використовується як новий вхід, з огляду на витрати, викликані виконанням програми, запропонований метод не отримує динамічні мультимодальні дані зв'язку сенсорів кожної умовної гілки повторно. Під час

виконання тестового прикладу рядки 21~24 зберігають пройдені кодові шляхи, поміщають умовні гілки без аналізу до черги введення, а потім визначають пріоритетний рейтинг відповідно до кількості нових гілок. Рядок 25 алгоритму зберігає динамічну мультимодальну послідовність даних сенсорного зв'язку кожного вузла для рядків 9~13, щоб судити, чи генерує поточний вузол тестовий приклад. Якщо в списку є динамічна мультимодальна послідовність даних сенсорного зв'язку та згенерований тестовий приклад, який відповідає $DIF(x_i)$ і обмеження (x_i) поточного вузла x_i, відповідно, тоді алгоритм пропустить генерацію тестового прикладу безпосередньо вузла. Рядок 29 алгоритму зберігає діаграми потоків динамічного керування як шляхи, які досліджуються в черзі після завершення шляхів коду програми.

Запропонований метод фокусується на технології генерації та створенні тестів для специфічного вузла x_i шляхом поєднання з відповідними динамічними мультимодальними даними комунікаційного сенсора. На рис. 3.6 зображено алгоритм 2, який описує цей процес.

Функція генерації тестового прикладу *makeTestCases* розглядає поля протоколу, тобто елемент у наборі $DIF(x_i)$ у визначенні 1, обмеження правила c і протокол G як вхідні дані, а вихідним є набір тестових випадків *tcList*. Ключом до функції є створення тестових випадків шляхом отримання ефективної граматики кожного вузла та зворотної граматики. Рядок 2 містить лише одну дійсну граматику для полів протоколу, тобто виводить граматику кожного поля протоколу в умові обмеження відповідно до параметрів як дійсну граматику вузла.

Можливість полягає в тому, що дійсна логіка, застосована до вузла x_i, безумовно є підмножиною протокольної граматики G, тому що згідно з визначенням 3.2, тобто в умові обмеження *Constraint* (x_i), дійсна логіка може застосовуватися до всіх полів протоколу в $DIF(x_i)$. Рядок 3 означає, що поле протоколу містить багато правильної граматики. Наприклад, для вузла x_i існує $DIF(x_i) = \{F_a, F_b\}$, а дійсна логіка, виведена в C*onstrain*t (x_i), є $F_a = (0|1)$, $F_b = 2$, а сукупна логіка дорівнює ($F_a = 0$, $F_b = 2$) і ($F_a = 1$, $F_b = 2$). Далі, рядки 4~9 алгоритму змінюють правильну логіка, що відповідає кожному з полів дійсної граматики в

передумові виконання умови обмеження *Constraint* (x_i), а потім об’єднує їх з правильною граматикою інших полів для нечіткої граматики та додає граматику до набору тестових випадків. Якщо в $DIF(x_i)$ не знайдено відповідної тестової граматики, дані регістра повинні генеруватися шляхом зміни таких методів, як випадкове переміщення бітів, екстремальна заміна та заміна граничних значень.

Алгоритм 2. Алгоритм генерації тестових кейсів у поєднанні з динамічними мультимодальними даними зв'язку з датчиків
Вхідні дані : Поля пртоколу **fields**, обмеження правил **c,** протокол граматики **G**
Вихідні дані: Набір або список тестових випадків **tsList**
Фунція: make TestCases (fields, c, G)

```
1. tcList ← empty ( )
2. validGrams ← extractGrams(fields, c, G)
3. tcList ← combination ( validGrams) ^ c
4. for each g in validGrams do
5. fg ← ~ g ^ c
6. fuzzyGram ← fg ^ (other g` in validGrams)
7.   teList.add(fuzzyGram);
8. end
9. return teList
```

Рис. 3.6. Алгоритм 2. Алгоритм генерації тестових прикладів у поєднанні з динамічними мультимодальними даними сенсорів

Використовується протокол Modbus як приклад. Поля протоколу Modbus включають ідентифікацію, довжину інформації, поле коду функції, початкову адресу, кінцеву адресу. Поле даних визначено відповідно до коду функції та CRC (циклічна перевірка надлишковості). Значення обчислення CRC та інших алгоритмів контрольної суми як механізмів, що завершують зв'язок протоколу, як правило, вбудовуються в специфікації протоколу для виявлення потенційних пошкоджених даних. Хоча механізм перевірки Modbus/TCP реалізовано в кадрі передачі TCP, а формати протоколу не включають CRC, перевірка CRC зазвичай існує в Modbus RTU та інших промислових протоколах Інтернету. Якщо CRC, отриманий програмою, не збігається з CRC, отриманим шляхом обчислення, дані відповідного випадку можуть бути проігноровані безпосередньо. Це важливий механізм для забезпечення безпеки та функціональності, але якщо значення CRC не оновлюється при зміні полів протоколу, fuzz-тест має бути заблокований, тому CRC також вважається важливим полем.

На рис. 3.7 зображено блок-схему управління та шляхи виконання першого входу. Це адинамічна блок-схема управління, побудована за допомогою дійсного входу, що виконує функцію запису, права половина якої показує шляхи, фактично пройдені введенням під час виконання прикладної програми, а пунктирна частина показує нові шляхи виконання, виявлені в кожному вузлі. Початковий вузол 3_1 представляє перше виконання умовної гілки, елементи в дужках представляють *DIF*(3_1), поле, яке впливає на умовну гілку; сторона з стрілкою представляє залежність і контроль відношення *CDC*(3_1). Блок-схема управління, показана на рис. 3.7, є тестовим прикладом, згенерованим шляхом вибору комбінації коду функції 0x05 і *n* полів даних у вузлі 24_1 випадку, що відповідає дійсності під час виконання першого входу і розглядається як другий вхід.

Припустимо, що початковий вузол 3_1 не залежить від жодного вузла, розглядає поле ідентифікації протоколу *I* як динамічне інтерактивне поле і має єдине істинне значення *0x0000*, а потім отримуємо відповідні дві тестові граматики після фазифікації з алгоритмом. Вузол 3_1 насправді є умовною гілкою для визначення того, чи є поле істинним чи хибним відповідно до протоколу, тому відповідно до кожної тестової граматики легко зробити висновок, що умова обмеження, яка робить вузол 3_1 істинною, дорівнює $I \neq 0x0000$.

Умова, що робить умовну гілку виконуваною, $I \neq$ 0x0000, тоді як обмеження, що робить умовну гілку хибною, $I = 0x0000$. У цьому випадку для наступного вузла 8_1, відповідно до його *CDC*(8_1). *Constraint* (8_1) є обмеженням, яке робить умовну гілку 3_1 хибною, тобто I = 0x0000. Подано циклічну надмірність як єдине поле перевірки граматики, застосовуване до всього поля, генерує дві тестові граматики $I = 0x000 \wedge CRC(I, L, F, S, E, D) = C$ і $I = 0x000 \wedge CRC(I, L, F, S, E, D) \neq C$ аналогічним чином. На основі цього отримуємо умову обмеження вузла 8_1, а потім аналогічним чином розв’язуємо кожен вузол.

Слід зазначити, що відповідно до опису алгоритму, тестова логіка, згенерована у вузлі 13_1, вирішила, чи є вузол 14_1 істинним чи хибним. Тому необхідності тестової логіки, згенерованої у вузлі 14_1, немає. Аналогічно пропустити вузли 21_1, 21_2, 22_2, 24_2 і 21_3. Відповідно до алгоритму 1, коли генерується тестова догіка,

динамічна блок-схема зберігається в черзі як досліджуваний шлях. У цьому випадку при пошуку шляхів на схемі потоків керування можна отримати вузол розділу, який генерує різні шляхи, порівнюючи шляхи та розглядаючи вузол як початковий. Усі вузли перед цим вузлом мають однакові динамічні мультимодальні дані зв'язку сенсорів і, таким чином, можуть бути вилучені безпосередньо, щоб уникнути повторного генерування тестових випадків.

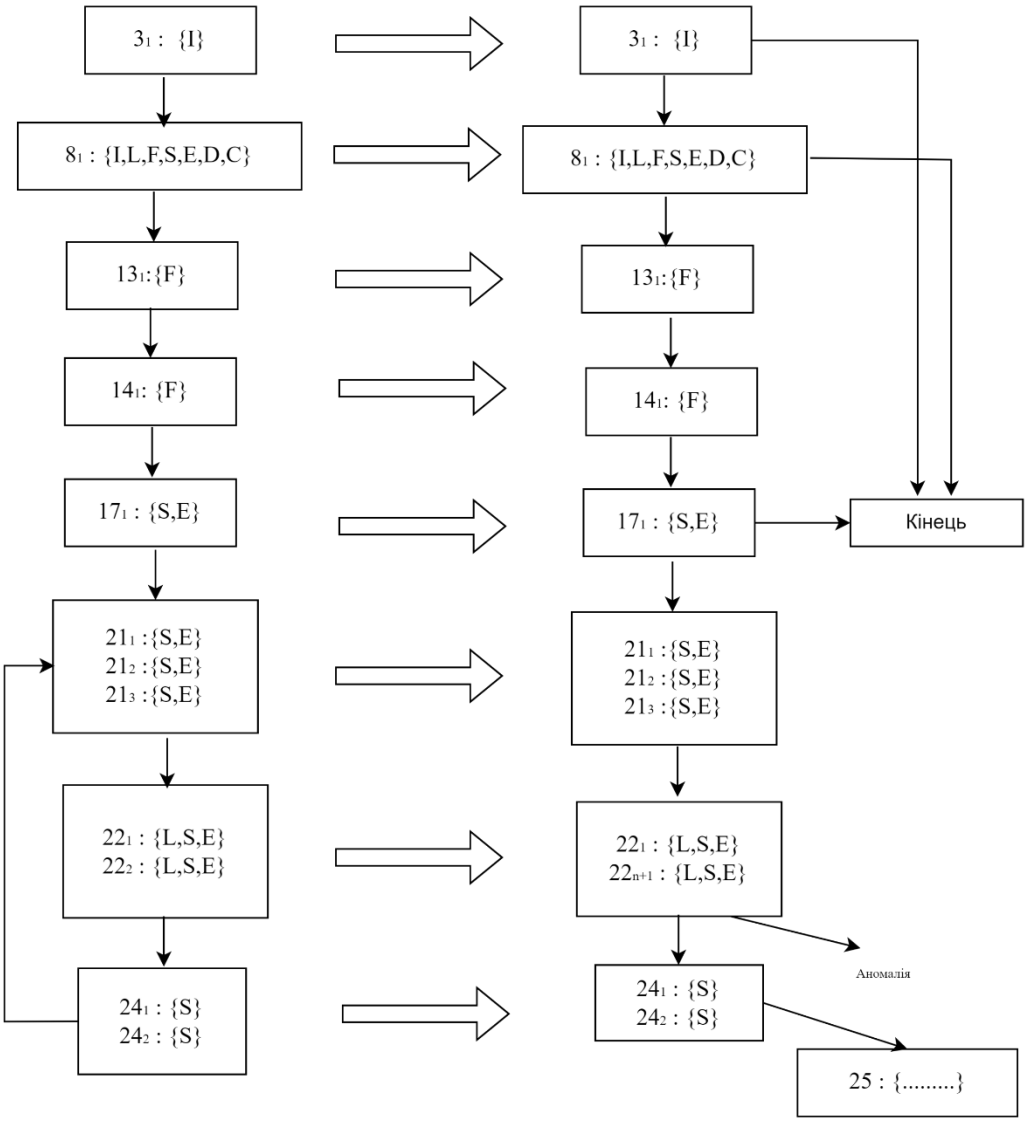


Рис. 3.7. Блок-схема управління та шляхи виконання першого випадку

У таблиці 3.1 наведено приклад граматики генерації тестового прикладу першого входу, отриманого за допомогою алгоритму 2. Експеримент на прикладі протоколу Modbus/TCP. Експеримент проведено з Ubuntu 14.04, тому вибрано бібліотеку з відкритим вихідним кодом libmodbus для реалізації Modbus. Зв'язок

протоколу TCP в системі Linux. Реалізовано запропонований нечіткий метод із Symfuzz інструментом, розробленим на основі BAP [183] з відкритим вихідним кодом для аналізу двійкової системи, який може конвертувати виконувані файли на проміжну мову, застосовну для аналізу програм, і в поєднанні з PIN-кодом для виконання динамічних інструментів двійкової системи до цільової програми для отримання динамічних інтерактивних полів і взаємозв'язків залежності та контролю, необхідних методу для генерування наступного тестового прикладу. AFL-fuzzer бкли вибрано як нечіткий інструмент. Підпорядкована станція Modbus чекає повідомлень запиту від інших головних станції. В експерименті ми встановили, що повідомлення головної станції надсилають 25, 30 і 35 повідомлень запитів, як тестовий вхід, до підпорядкованої станції, відповідно, і здійснюють нечітку обробку до програми введеної станції Modbus і контроль на аномалії.

Таблиця 3.1

Умови генерації тестових прикладів

Вузол xi	DIF (xi)	CDC (xi)	Обмеження (xi)	Дійсний логіка	Логіка після фазифікація	Тест логіка
31	I	Ø	ø	I = 0x0000	I ≠ 0x0000	I = 0x0000 I ≠ 0x0000
81	I~C	(31, F)	I = 0x0000	CRC (I ~ D) = C	CRC (I ~ D) ≠ C	CRC (I ~ D) = =C CRC (I ~ D) ≠ ≠C
131	F	(81, F)	I = 0x0000 CRC (I ~ D) = =C	F = 0 x 01	F ≠ 0 x 01	F = 0 x 01 F ≠ 0 x 01
141	F	(131, F)	I = 0x0000 CRC (I ~ D) = =C F ≠ 0 x 01	Пропустити	Пропустити	Пропустити
171	S, E	(141, T)	I = 0x0000 CRC (I ~ D) = =C F = 0 x 05	C ≤ E	C > E	S ≤ E C > E
211	S, E	(171, F)	I = 0x0000 CRC (I ~ D) = =C F = 0 x 05 C ≤ E	Пропустити	Пропустити	Пропустити
211	L, S, E	(211, T)	I = 0x0000 CRC (I ~ D) = =C F3 = {1}, F4 ≤ ≤F5	Вхідні дані розмір є дійсний	Вхідні дані розмір є дійсний	Вхідний розмір є дійсний

Щоб оцінити ефективність запропонованого методу fuzz-тестування, ми проводимо порівняння за кількістю тестових випадків, загальною кількістю сценаріїв виконання, швидкістю покриття коду та часом тестування з такою ж кількістю вибірок.

Експеримент складає статистику кількості тестових випадків, створених за допомогою двох методів нечіткої перевірки з 25, 30 і 35 вибірками даних. Кількість тестових випадків відноситься до загальної кількості зразків, згенерованих після збою програми або повного виконання зразків.

На рис. 3.8 видно, що із збільшенням кількості зразків, тестові дані, отримані за допомогою запропонованого методу, значно менші, ніж тестові дані, які генеровані за допомогою AFL-fuzzer, оскільки запропонований метод конструює тестові випадки, використовуючи технологію на основі генерації даних, і в цьому випадку створені тестові дані є меншими, ніж у AFL-fuzzer, що використовує стратегію варіації.

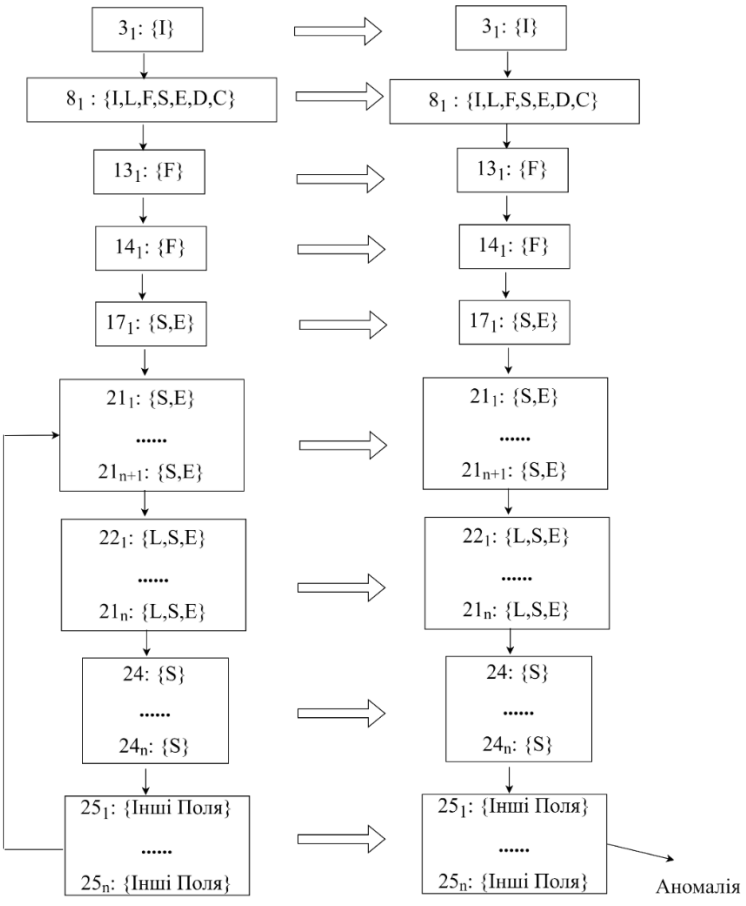


Рис. 3.8. Блок-схема управління та шляхи виконання другого випадку

Крім того, видно, що кількість тестових випадків, згенерованих AFL-fuzzer, зростає зі збільшенням кількості вибірок, оскільки Afl-fuzzer скорочує вхідні зразки з огляду на те, що користувачі можуть запропонувати початкові зразки низької якості і, таким чином, спричинити можливу надмірність даних у деяких типах варіації. Хоча запропонований метод залежить від кількості зразків, він використовує більш відповідну стратегію генерації.

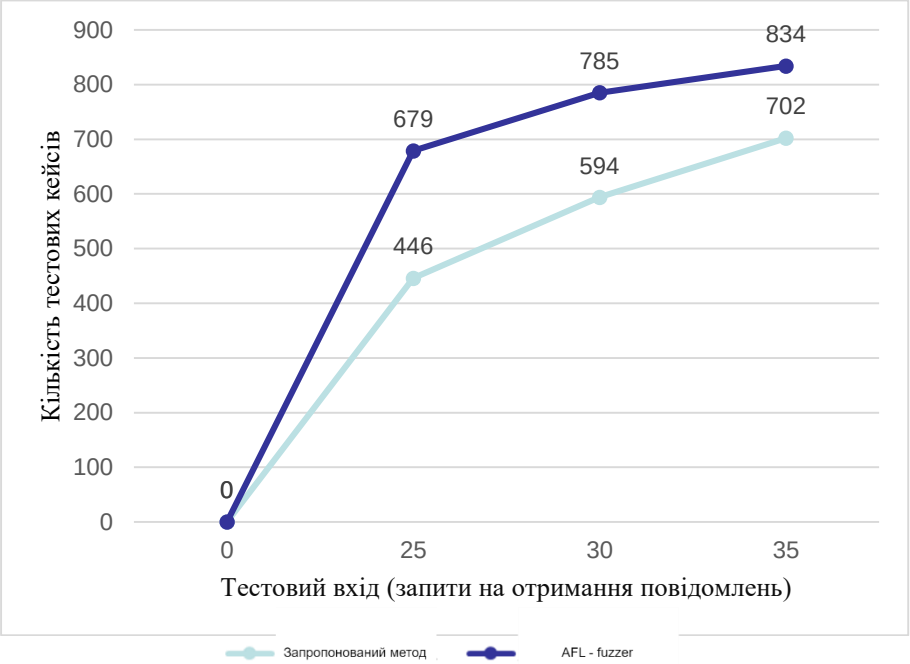


Рис. 3.9. Кількість тестів із різною кількістю зразків

На рис. 3.10 показана швидкість покриття коду. Принцип розрахунку полягає в тому, що прилади можуть допомогти фіксувати швидкість охоплення гілки та виявляти грубе ураження підрахунок виконання філії. Швидкість покриття коду не обов'язково має зв'язок з ймовірністю виявлення аномалій, але, безсумнівно, для тестового випадку шлях виконання, який не досягає умовної гілки програми на глибокому рівні, безсумнівно, не спричинить жодної потенційної аномалії.

Отже, ми бачимо, що запропонований метод реалізує більшу швидкість охоплення коду порівняно з AFL-fuzzer. Таблиця 3.2 показує статистику, коли цільова програма аварійно завершує роботу.

Генерація тестового прикладу шляхом поєднання з програмними динамічними мультимодальними даними комунікаційного сенсора платить ціну тестового часу для вирішення умовного обмеження розгалуження та генерування тестової граматики в порівнянні зі стратегією варіації.

Однак це значно збільшує кількість шляхів виконання та швидкість охоплення коду, що вказує на те, що під час роботи програми чим більше шляхів виконання знайдено умовними гілками, тим більшою є ймовірність того, що тестові випадки досягнуть глибокого рівня та спровокують аномалії, і тим сильніша доречність.

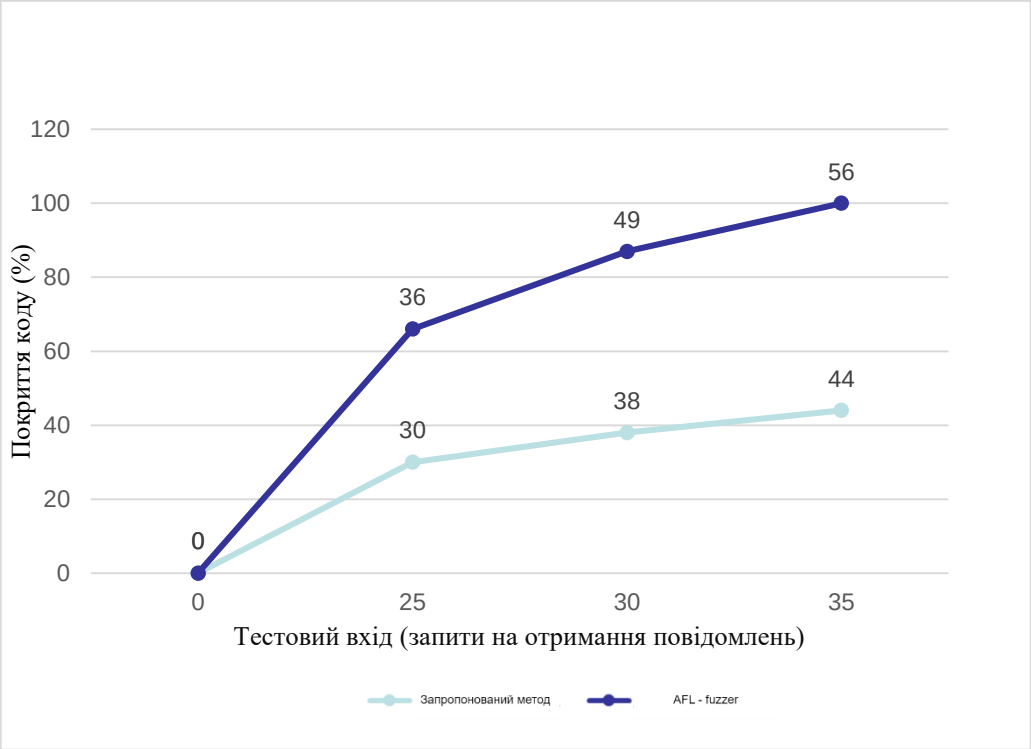


Рис. 3.10. Коефіцієнти покриття коду з різною кількістю вибірок

Таким чином, запропонований метод перевершує AFL-fuzzer з точки зору реалізації тесту протоколу Modbus-TCP.

Таблиця 3.2

Порівняння результати з експеримент

Метод тестування	Кількість тестових випадків	Кількість пройдених умов	Тестове покриття коду	Час виконання годинах	Аномалія
Method proposed	18492	3862	55,90%	4.19	1
AFL – fuzzer	46931	1787	39,85%	3.35	1

3.2. Метод очищення даних у бездротових сенсорних мережах

Бурхливий розвиток всесвітньої павутини призвів до експоненційного збільшення кількості онлайн-інформації. Джерела інформації є дуже об’ємними. Найбільшими джерелами даних є мережева інформація. В додаток

значного розвитку набули реляційні та нереляційних бази даних. Поширені проблеми – це проблеми інтеграції даних. Інтеграція даних є важливим етапом обробки в багатьох областях [188]. Для великого набору даних критерієм є якість відповідних параметрів [189]. Однак є деякі помилки, яких не можна уникнути під час інтеграції процесу. Загалом, основна причина проблем у процесі інтеграції полягає в тому, що серед баз даних немає узгоджених стандартів, а формат даних неоднаковий, що вплине на дані. Інтеграція спричинила певні перешкоди [190]. Тому при введенні великих обсягів даних завжди будуть деякі помилки та деякі суперечливі дані [191].

БСМ - це технологія отримання та обробки інформації, яка в основному складається з сенсорів, MEMS та мережевих систем [192]. Кожен сенсорний вузол може вимірювати та аналізувати сигнали в навколишньому середовищі за допомогою вбудованих сенсорів і отримувати необхідні дані [193]. У порівнянні з традиційною мережею, безпровідна сенсорна мережа зосереджена на даних, а не на передачі даних. Бездротовий сенсор має широкий спектр функцій, за допомогою яких можна не тільки виявляти навколишнє середовище та стан будівлі, але й керувати розумним будинком за допомогою певних технічних засобів [194]. З огляду на характеристики бездротових сенсорних мереж, буде проведено поглиблене дослідження методів очищення даних, керування талантами в бездротових сенсорних мережах на основі технології інтелекту. За допомогою цього методу дані керування талантами в бездротових сенсорних мережах можуть бути реалізовані самостійно. У разі його ефективності загальний розмір даних зменшується. Зменшення масштабу даних управління талантами може не тільки підвищити ефективність процесу аналізу, але й покращити якість результатів аналізу.

3.2.1. Структура бездротової сенсорної мережі

Бездротові сенсорні мережі в основному складаються з сенсорних вузлів, зон виявлення та серверів. Сенсорні вузли можна розташувати поблизу об’єктів, які потребують даних вимірювань за допомогою ручного розгортання. Після розгортання ці сенсорні вузли певним чином самоорганізовуються, і вони

будуть спільно сприймати навколишнє середовище та об’єкти, щоб отримати необхідні дані. Ця самоорганізуюча форма може сформувати відповідну мережу і передавати всі дані назад на головний вузол через режим ретрансляції. Нарешті, всі дані у всьому вузлі передаються на сервер через систему зв'язку. Коли користувачі використовують бездротові сенсори для отримання даних, вони можуть ефективно збирати необхідні дані за допомогою управління та контролю вузлів. Архітектура типової бездротової сенсорної мережі показана на рис. 3.11.

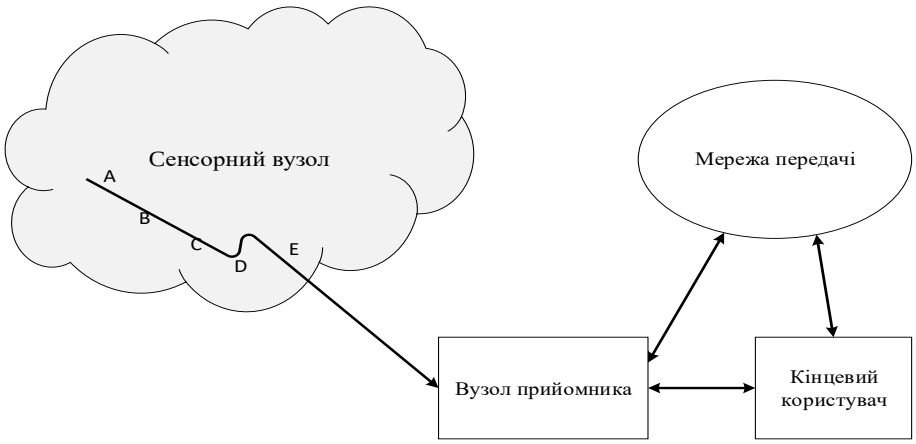


Рис. 3.11. Топологічна структура сенсорних мереж

Сенсорний вузол складається з сенсорного модуля, модуля обробки, модуля бездротового зв'язку та модуля енергопостачання. Сенсорний модуль відповідає за отримання інформації та перетворення даних. Модуль обробки контролює роботу всього сенсорного вузла, обробляє дані, зібрані ним самим, і дані, надіслані іншими вузлами, і запускає мережевий протокол для управління процесом зв'язку вузла; модуль бездротового зв'язку спеціально пов'язаний з іншими сенсорними вузлами. Дані надсилаються та отримуються; модуль живлення забезпечує енергією сенсорні вузли.

Для функції мережі кожен вузол сенсора в бездротовому сенсорі повинен враховувати як традиційні вузли мережі, так і маршрутизатори, не тільки для збору та обробки локальної інформації, але й для обробки даних, що передаються з інших вузлів.

У процесі передачі даних вузли повинні мати можливість працювати разом. Наразі апаратна та програмна технологія сенсорного вузла є центром дослідження сенсорної мережі, оскільки він має потужну здатність обробляти, зберігати та передавати дані вузла. Під’єднавши сенсор до Інтернету, можна конвертувати зв’язок між різними мережевими протоколами. У той же час він також може розподіляти завдання на всі вузли одночасно і передавати зібрані дані у зовнішню мережу. Для різних застосувань склад сенсора також різний, але майже всі мають спільні характеристики, вони, як правило, включають блок сенсорів, блок обробки, блок бездротового зв'язку і блок живлення, Компоненти сенсорного вузла зображені на рис. 3.12.

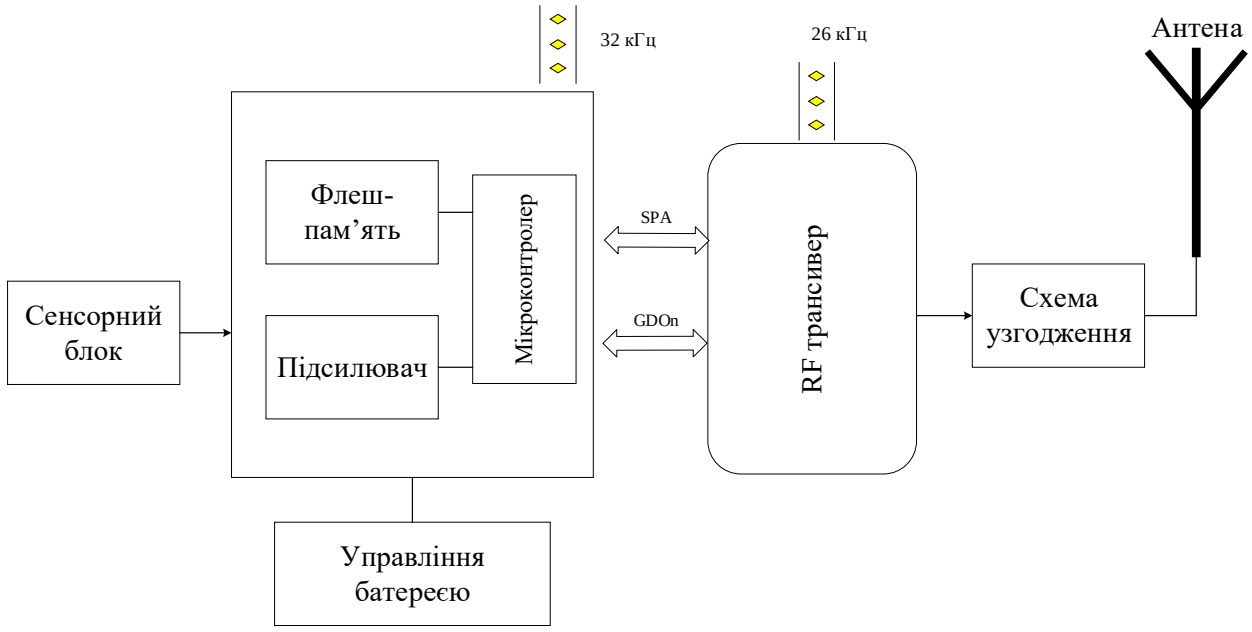


Рис. 3.12. Компоненти сенсорного вузла

Склад сенсорного блоку, як правило, відносно простий і в основному складається з сенсора і функціонального модуля аналого-цифрового перетворення, який в відповідає за перетворення даних для отримання інформації в зоні виявлення. Основною частиною процесорного блоку є вбудована система, яка в основному включає центральний процесор, пам'ять тощо, і в основному відповідає за керування вузлами всього сенсора та зберігання зібраних даних та обробку даних, отриманих іншими вузли.

Основна функція пристрою бездротового зв'язку полягає в тому, щоб завершити передачу даних без використання дротового пристрою.

Основною частиною блоку енергопостачання є модуль живлення, основною функцією якого є забезпечення енергією сенсорних вузлів. Є також деякі інші модулі, такі як системи позиціонування та мобільні системи. Завдяки співпраці цих пристроїв бездротова сенсорна мережа може працювати нормально. При використанні бездротової сенсорної мережі для отримання даних, необхідно налаштувати велику кількість сенсорних вузлів. Тому кількість сенсорних вузлів може бути дуже великою. Тому що кількість сенсорних вузлів відносно велика, а обсяг невеликий, і персонал у деяких площі не можуть прибути точно вчасно, сенсори не можуть бути доповнені заміною батареї. У цей момент є сенс розрахувати енергоспоживання вузлів сенсора. Основна енергоємна частина сенсорного вузла дуже енергоємна при передачі даних на модуль бездротового зв'язку. Модуль бездротового зв’язку має чотири стани бездротового зв’язку: надсилання, отримання, очікування та сплячий режим. Зв'язок між споживанням енергії бездротового зв'язку та відстанню зв'язку показано у формулі 3.1.

$$E = kd^n , \tag{3.1}$$

де E - це енергоспоживання бездротової мережі зв'язку, d - відстань, a та k - константи. Зі збільшенням відстані зв’язку збільшується енергія споживання.

3.2.2. Математична модель інтелекту

Математична модель аналізу мережі даних Байєсова - це модель, яка зазвичай використовується в процесі інтелекту даних, вираз якої показано у виразі (3.2).

$$P(x_1, \ldots . x_n) = P(x_1)P(x_2/x_1)P(x_n/x_1, \ldots . x_{n-1}) \tag{3.2}$$

Формула для ступеня довіри моделі наведена у виразі 3.3.

$$p(Mi/D) = \frac{p(D/Mi)p(Mi)}{p(D)} , \tag{3.3}$$

p – ймовірність відображення краю.

Інформаційний критерій Байєса (BIC) велика вибіркова апроксимація ймовірності краю. Використовуючи апроксимацію Лапласа, можна виконати наближення великої вибірки для *P*. Логарифмічна функція правдоподібності може бути розширена за допомогою оцінки максимальної правдоподібності, а потім обчислення може бути перетворено на багатовимірну функцію нормального розподілу в крайній точці сусідніх точок. Лапласівське наближення використовується для апостеріорної ймовірності, як показано в рівнянні (3.4).

$$p(D/m) = \int p(D/\theta, m)p(\theta/m)d\theta \qquad (3.4)$$

Завдяки встановленню моделі даних стає зручним використовувати технологію аналізу даних для їх обробки.

3.2.3. Технологія очищення даних на основі режиму кластеризації

Поточні дані управління талантами стикаються з проблемою раптового збільшення даних раптового збільшення даних. Ці великі бази даних зазвичай містять помилки або невідповідності через певні причини. Причини помилок включають неправильне введення, що спричиняє неправильні значення, оскільки введені дані непослідовні викликані різними форматами або використанням різної абревіатури не можуть повністю збирати інформацію про дані і призводять до втрати отриманих даних. Усі ці причини можуть призвести до неминучих відхилень підприємств та установ, коли вони приймають важливі ділові рішення, що призводить до величезних збитків. Процес очищення даних полягає у вирішенні поширених помилок і невідповідностей у великих базах даних, а деяка проста попередня обробка перед очищенням даних може покращити якість очищення даних. Блок-схема в очищенні даних зображено на рис. 3.13.

Рис. 3.13. Основний процес очищення та попередньої обробки даних

3.2.4. Алгоритм видалення запису реплікації на основі кластера

Об’єднання великих баз даних часто стикається з проблемами наприклад, неправильне введення даних, різні схеми або непослідовність форми скорочень. Ці проблеми викличуть об’єднану базу даних, щоб мати кілька записів, які представляють ту сама сутність, але мають дещо інший атрибут значення, що створює суперечливі дані. Після очищення і попередньої обробки, деякі прості помилки в базі даних очищаються. Однак тому, що об’єкт підлягає обробці є великою базою даних, кількість даних, з якою потрібно зіткнутися, дуже великий, тому він все ще містить багато помилок і непослідовних даних. Однак, оскільки об’єкт, який підлягає обробці, є великою базою даних, обсяг даних, з яким потрібно працювати, дуже великий, тому він все ще містить багато помилок і суперечливих даних. Показник точності є порівнянням кластеризації. Визначення чистої кластеризації відноситься до того, що всі записи, які містяться в кластері, представляють той самий суб'єкт.

Експериментальний метод використовується для оцінки даних у великомасштабній базі даних. Метою точності в процесі вимірювання є вся бази даних, а не лише одні дані в базі даних. Коли дані записуються за допомогою чистої кластеризації, представлення записів однакове. Якщо записи мають різні форми, ця

форма кластеризації не є чистою кластеризацією, що вказує на те, що метод кластеризації неточний. Використання алгоритму видалення записів реплікації на основі кластера може значною мірою вирішити проблему невідповідності даних. Цей метод дозволяє зменшити обсяг обробки даних і підвищити ефективність обробки даних. Основний процес попередньої обробки очищення даних показано в таблиці 3.3.

Таблиця 3.3

Основний процес очищення та попередньої обробки даних

Форма	Основний процес очищення та попередньої обробки даних
Очищення брудних полів даних	Основна мета цього кроку – усунути помилки введення даних. Деякі прості помилки під час виправлення записів даних за допомогою деяких зовнішніх функцій і зовнішнього джерела файли, як-от перевірка того, чи відповідає поштовий індекс місту та чи відповідають дата народження та вік. Це дозволить підвищити точність і стандартизацію даних, а також ефективно уникнути процесу кластеризації, оскільки помилка даних є занадто великою, щоб зробити запис одного і того ж об’єкта не в одному кластері.
Використовуйте уніфіковану абревіатуру	Відповідно до відповідного співвідношення абревіатури та повної назви всі дані обробляються стандартизованим способом, або уніфіковано абревіатуру або подання повної назви.
Перетворення даних	У цьому процесі ми в основному конвертуємо деякі дані в різні формати. У базі даних чоловік представлений в базі даних, а «1» виражається в іншій базі даних, що створює суперечливі дані. Процес перетворення даних полягає в перетворенні цих суперечливих даних у узгоджені дані. Цей процес також може перетворити таблицю даних у таблиці даних з безліччю різних структур відповідно до певних вимог.

Щоб зробити експеримент більш точним і мати можливість ефективно перевірити точність та ефективність роботи алгоритму, взято реальні дані кластеризації, які будуть використовуватися для аналізу, і визначаються конкретні значення використаних даних. Дані, використані в експерименті, є даними записів

управління талантами. Атрибути запису включали сім атрибутів управління талантами. Експеримент входить до 875 записів управління талантами. Через певну обробку копій, а потім за допомогою методу обробки випадкових помилок, отримано загальну кількість ефективних записів управління талантами - 2412. Загалом 218 кластерів, що містять більше двох записів, обчислюються вручну, з яких найбільший кластер містить загалом 14 записів.

Технологія виявлення кластеризації Canopy використовується для виявлення дублікатів записів. Є три основні параметри виявлення: пороги відстані T_1 і T_2 і постійний коефіцієнт *k*. Вибір T1 і T2 визначає розмір Canopy і ступінь його перекриття, тобто кількість даних, яку необхідно точно розрахувати. Вибір значення k визначає, чи можуть записи бути точно групованим. На початку обробки даних необхідно створити значення T_1 і T2.

Для їх встановлення використовується метод перевернутого виявлення двох значень. У разі різних T_1 і T_2 ($T_1 \leq T_2$), системна кластеризація повинна обчислити розраховану кількість відповідних пар для вимірювання якості T_1 і T_2.

Розрахунок різних значень T_1 і T_2 показано у таблиці 3.4. Згідно з експериментальними даними, наведеними в таблиці 3.2, при $T_1 = 0{,}75$ і $T_2 = 0{,}75$ точки даних, які необхідно точно розрахувати, є найменшими. Отже, $T_1 = 0{,}75$ і $T_2 = 0{,}75$ вибрано, що означає, що Canopy не перекривається. Відповідно до співвідношення отриманого числа кластерів і фактичної кластеризації, значення *k* показано в таблиці 3.5. При $k = 3$ коефіцієнт кластеризації найближче до справжньої кластеризації, тому експеримент вибирає поріг відстані $k = 3$. Однак коефіцієнт кластеризації все ще не досягає 1, оскільки міститься випадкова помилка в записах даних, які слід класифікувати в одному кластері.

Таблиця 3.4

Розрахунок різних значень T_1 і T_2

T_1	0.95	0.96	0.97	0.98	0.99
T_2					
0.95	2067	6632	8506	10,374	11,596
0.85	–	2043	2653	8450	10,846
0.75	–	–	2014	2788	10,895
0.65	–	–	–	2087	10,240
0.55	–	–	–	–	10,232

Домашня адреса таблиці записів управління талантами є складеним атрибутом. Метод перетворення даних програми розкладає місто, округ, конкретну вулицю, робочу одиницю та номер будинку адреси на підатрибути і в той же час виконує попередню обробку перед очищенням даних на основі використання зовнішнього вихідного файлу.

Таблиця 3.5

Коефіцієнт кластеризації різних значень k

k	5	4	3	2	1
Коефіцієнт кластеризації	0.786	0.9	0.981	1.073	1.26

Якщо місто, що відповідає домашній адресі, відповідає поштовому індексу, можна очистити деякі брудні дані. Зовнішній вихідні файли засновані на поштових індексах, виданих поштовим відділенням. Обсяг експериментальних даних дуже малий, що охоплює лише Гуансі, тому встановити 100% точність відносно легко під час налаштування зовнішніх вихідних файлів.

На рис. 3.14 показані результати експериментів із порівнянням попередньо оброблених і необроблених методів запису тестових копій, включаючи метод сусіднього сортування та метод запису тестової копії з використанням технології Canopy, у якому розмір вікна методу сусіднього сортування вибирано як два для порівняння ситуації. З рис. 3.14 видно, що точність методу виявлення попередньо обробленого дубльованого запису вища, ніж точність необробленого дубльованого запису методу виявлення, але оскільки використовувані експериментальні дані не великі, попередня обробка не може бути повністю відображена. У двох випадках ω=16 і ω=8 ω 16 є більш точним, ніж 8, оскільки найбільший кластер в експериментальних даних містить 15 записів. Якщо ω=8, це призведе до деяких дублікатів записів, які неможливо виявити. У випадку, коли ω= 16 потрібно зробити багато непотрібних порівнянь, що збільшує обсяг обчислень, але це може гарантувати більш високий рівень точності. У цьому експерименті попередньо оброблений метод ранжування сусідів ω=16 такий самий, як метод виявлення реплікаційного запису кластера Canopy, але метод Canopy має вищу швидкість

відклику, що вказує на те, що він може отримати більше записів реплікації.

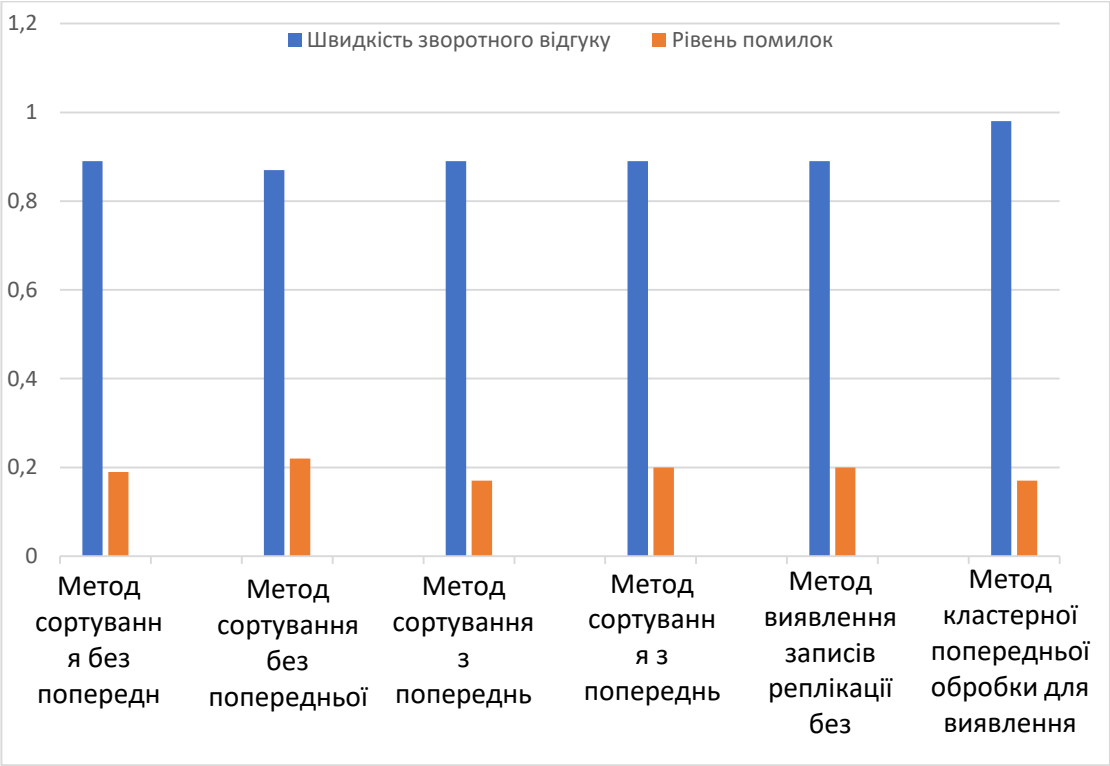


Рис. 3.14. Результати порівняння попередньої обробки тесту

Ця технологія використовується для дослідження методу даних управління талантами, очищення в бездротових сенсорних. Метод очищення даних є важливим дослідженням, яке може вирішити невідповідність даних при ідентифікації одного і того ж об'єкта і підвищити точність розпізнавання. При аналізі даних і прийнятті бізнес-рішень необхідно об’єднати деякі інформаційні дані, щоб легше знаходити цікаві шаблони. Причина неминуче призведе до отримання неправильних даних або суперечливих даних, тому приблизні повторювані записи з’являються в процесі злиття, що заборонено в базі даних, і ці повторювані записи необхідно видаляти [196].

Висновки до розділу 3

Запропоновано метод нечіткої обробки в поєднанні з динамічними мультимодальними даними сенсорів на рівні прикладної системи в промисловому Інтернеті. Цей метод відстежує виконання програм, а потім визначає поля, на які впливають умовні оператори, використовуючи динамічний аналіз виправлень, і

фіксує залежність умовних гілок, щоб належним чином контролювати генерацію граматик тестових випадків, тим самим збільшуючи здатність виконувати код на глибокий рівень. Результати порівняльного експерименту підтверджують, що метод певною мірою покращує валідність тестових випадків і швидкість покриття коду, а також підвищує ймовірність виявлення аномалій у реалізації протоколу.

Досліджено метод очищення даних управління талантами в бездротових сенсорних мережах на основі технології інтелекту. Проаналізовано конкретні форми застосування бездротових сенсорних мереж. Представлено характеристики структури бездротових сенсорних мереж та технологію очищення даних на основі моделі кластеризації. Запропоновано алгоритм видалення запису реплікації на основі кластерів та перевірено точність методів очищення даних. Отримані результати свідчать про ефективність використання досліджуваного методу.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ МЕТОДУ ІСТИННОСТІ МОНІТОРИНГУ ДАНИХ НА ОСНОВІ ТУМАННИХ ОБЧИСЛЕНЬ ТА ЗАСОБІВ ДЛЯ СИСТЕМ МУЛЬТИСЕНСОРНОЇ КОНФІГУРАЦІЇ

Нинішній підхід до IoT став поєднуватись з інноваційним машинним навчанням (ML) і штучним інтелектом (AI) на основі алгоритмів, для отримання цінної інформації, яка може бути використана іншими фізичними пристроями, створюючи КФС. Як зазначалося вище, КФС складається з фізичної частини та кіберчастини, з'єднаних через мережу зв'язку. Фізична частина в основному складається з сенсорів і приводів, які використовуються для збору даних і виконання завдань на основі зібраної інформації. В подальшому зібрані дані надсилаються через мережу до кіберрозділу, де вони зберігаються та обробляються за допомогою вдосконалених алгоритмів ML і AI для отримання цінної інформації, яку можна перевести в дії, які виконує фізична частина. Ця система актуальна для таких галузей як охорона здоров'я, логістика, видобування та переробка нафти та газу, транспорт, енергетику, гірничодобувну промисловість, металургію тощо. Застосування незначної кількості сенсорів дозволяю формувати промислові КФС (ICPS), які оптимізують функціонування галузей в цілому [196].

ICPS можуть створювати автономні машини для самообслуговування/відновлення та вдосконалення управління запасами за допомогою ML. Базуючись на промисловому Інтернеті речей (IoT), ICPS може збирати дані про транзакції та надсилати їх через мережу на хмарні сервери, де вони аналізуються та зберігаються, щоб бути доступними за потреби. Через швидке зростання та різноманітність пристроїв, підключених до IoT, традиційна централізована мережева архітектура повинна відповідати новим вимогам до послуг і викликам, а також ефективно ідентифікувати та надавати великі обсяги даних щодо безпеки, цілісності та конфіденційності та інших сфер.

Обчислення Fog/Edge використовувалися для вирішення зазначених проблем і досягнення кращої якості обслуговування (QoS) і досвіду (QoE) шляхом виконання операцій зберігання та обробки даних фізично поблизу джерела даних у розподіленій інфраструктурі [197,198]. Блокчейн – це ще один метод, який може

доповнювати системи IoT, забезпечуючи функціональність безпечних і надійних систем для зберігання та обробки даних [199,200].

4.1. Промислова кіберфізична система

Сьогодні, завдяки інноваціям, таким як застосування сенсорів та виконавчих механізмів, можна значно покращити промислове виробництво шляхом ведення моніторингу якості продуктивності. Промислова кіберфізична система (ICPS), заснована на IoT, яка з'єднує машині та промислові об'єкти, такі як транспортні засоби, генератори електроенергії та інші вузли, дає змогу збирати, обмінюватися, зберігати та аналізувати дані для передачі цінної інформації та розуміння, що дозволяє швидко приймати точні рішення. Таким чином, це покращує продуктивність в цілому та продуктивність промислових процесів зокрема [201,202]. ICPS поєднує потужність цих розумних фізичних машин з аналізом даних у реальному часі для досягнення вищої ефективності системи та швидшої реакції, що призводить до таких переваг, як економія часу та коштів, кращий контроль якості та управління енергією, моніторинг активів, прогнозування технічного обслуговування та зменшення відходів.

Розумна промислова індустрія (рис. 4.1) базується на поєднанні різних областей, у тому числі IIoT, робототехніки, штучного інтелекту, машинного навчання та комунікацій, щоб забезпечити більшу ефективність і достовірність. Це також впливає на безпеку та здоров'я працівників.

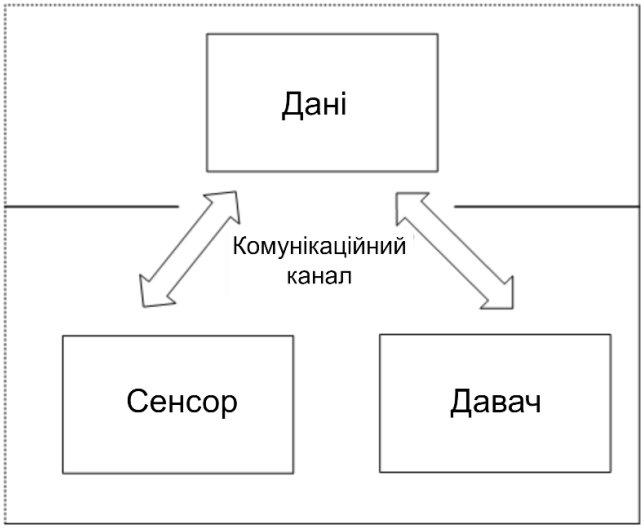


Рис. 4.1. Промислова кіберфізична система

Типові додатки IoT та IIoT базуються на бездротових мережах сенсорів, які збирають і передають дані в центр зберігання для обробки, аналізу та зберігання. Ці додатки, які розгорнуті на багатьох різнорідних пристроях, безперервно виробляють, обмінюються та споживають дані через мережу, що призводить до значного збільшення згенерованих даних. Основна вимога IIoT полягає в тому, щоб отримувати точні дані в режимі реального часу, а потім надавати швидкі та відповідні рішення, які забезпечують потрібну продуктивність [196-204]. Однак можливі і виклики IIoT.

Зберігання та маніпулювання даними. IoT-пристрої обладнані сенсорами, які мають мінімальні обчислювальні можливості та малий обсяг пам'яті. Зібрані дані надсилаються, зберігаються та обробляються у віддалених хмарах. Масштабування великої кількості пристроїв і швидка обробка величезних обсягів згенерованих даних є складними проблемами для поточних централізованих хмарних моделей, які використовуються в більшості рішень IoT. Нові рішення вимагають достатньої високоефективної обчислювальної потужності, щоб скористатися перевагами вдосконалених інструментів і механізмів аналізу, які обробляють і зберігають величезні обсяги даних і виконують великі програми. Існуючі централізовані хмарні парадигми не підходять для ефективного задоволення багатьох нових і майбутніх вимог для послуг, таких як доступність і ефективність системи, затримка та безпека в масштабованих мережах.

Якість обслуговування (QoS). Додатки IIoT включають різні типи даних, в тому числі і реагування на надзвичайні ситуації, відеоспостереження в реальному часі, комп'ютерне спостереження та автономне керування, і всі вони мають вимоги до QoS, які можуть змінюватися з часом, наприклад як затримка, пропускна здатність і надійність. IIoT має бути здатним адаптуватися до цих варіацій і надавати кожному пристрою необхідну послугу..

Безпека та конфіденційність. У додатку IoT мережева інформація, якою обмінюються, може бути конфіденційною, вимагати захисту та надавати обмежений і контрольований доступ до даних. Більшість пристроїв IoT є вразливими, оскільки вони мають обмежені можливості безпеки і можуть бути

відносно легко зламані хакерами шляхом доступу до збереженої інформації або надсилання неправильних даних у хмари. Захист системи IoT має вирішальне значення, і його слід гарантувати як на рівні мережі, так і на рівні системи зберігання.

Туманні обчислення можна вважати розширенням парадигми хмарних обчислень, представленої у багаторівневій структурі сервісу, як показано на рис. 4.2. Вони дозволяють здійснювати локальний моніторинг в реальному часі та оптимізацію для додатків IoT, тоді як хмара забезпечує глобальну оптимізацію та інші розширені послуги.

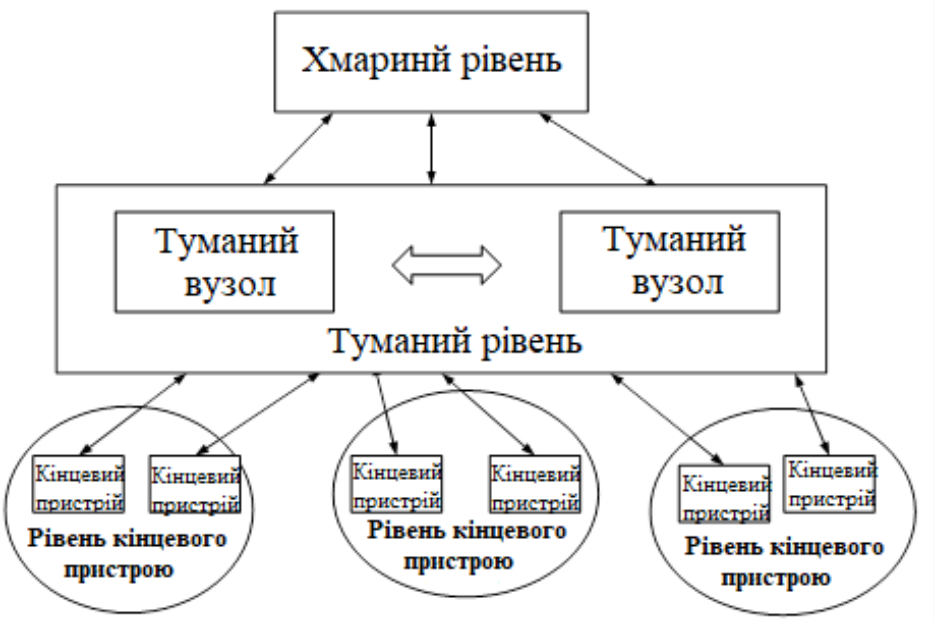


Рис. 4.2. Архітектура туманного обчислення

4.1.1. Рівні туманної архітектури

Архітектура туману складається з трьох рівнів:

- *Рівень кінцевих пристроїв*: включає пристрої кінцевих користувачів IoT, які керують генерацією даних. Їхнє головне завдання - сприймати навколишні об’єкти та події та передавати дані на верхні рівні для зберігання та обробки.

- *Рівень Туману*: як середній рівень, він має значну кількість вузлів туману (зокрема, серверів, маршрутизаторів, комутаторів), які розташовані на краю мережі та широко розподілені географічно. Усі ці пристрої з’єднані разом і співпрацюють,

щоб зберігати, обчислювати та обмінюватися отриманими даними. З'єднання між туманими вузлами і рівнем кінцевих пристроїв здійснюється за допомогою бездротових технологій (зокрема, WiFi, 4G, Bluetooth), і кожен вузол підключений до хмари через Інтернет-мережу IP. Вузли туману аналізують і зберігають отримані дані та надсилають лише ті, які вважаються цінними, на хмарний сервер для зберігання або наступної обробки.

•*Хмарний рівень*: він включає потужні можливості зберігання та обчислення для аналізу значного обсягу обчислень і постійного зберігання значного об'єму даних. Для оптимальної ефективності не всі обчислювальні завдання та завдання зберігання даних виконуються через хмарний рівень.

4.1.2. Багаторівнева архітектура модуля Fog

Представлено платформу, яка є багаторівневою архітектурою для аналізу даних, що надходять від інтелектуальних пристроїв на основі Інтернету речей, таких як інтелектуальні системи відеоспостереження. Три рівні складаються з хмарних обчислень, обчислень на межі туману та сенсорів, які працюють разом один з одним. Як бачимо, туманне обчислення – це в основному технологія віртуалізації, яка пропонує зберігання та обчислення між кінцевими пристроями та хмарним рівнем. Схематичний огляд нашої архітектури показано на рис. 4.3. Перший фізичний рівень складається з набору пристроїв IoT і змінних сенсорів. Основна робота сенсорів полягає в тому, щоб збирати всі дані та надсилати їх на межові шлюзи через розвантаження. Потім другий рівень, який складається з периферійних шлюзів і серверів edge-fog. Коли дані надходять до крайових шлюзів, дані потрібно відфільтрувати, попередньо обробити для подальшої обробки. У цьому процесі видаляється 30–70% безглуздих даних для їх аналізу. Завдяки цьому можна зменшити навантаження на передачу даних і збільшити швидкість обробки аналізу даних. Цей рівень працює як сервер. Обсяг даних надходить на периферійні сервери, а потім розподіляється між різними периферійними пристроями відповідно до вимог обчислення даних для зменшення затримки в сценарії реального часу. Роботи з розвантаження запитів повинні здійснюватися за допомогою запропонованого ефективного алгоритму планування завдань.

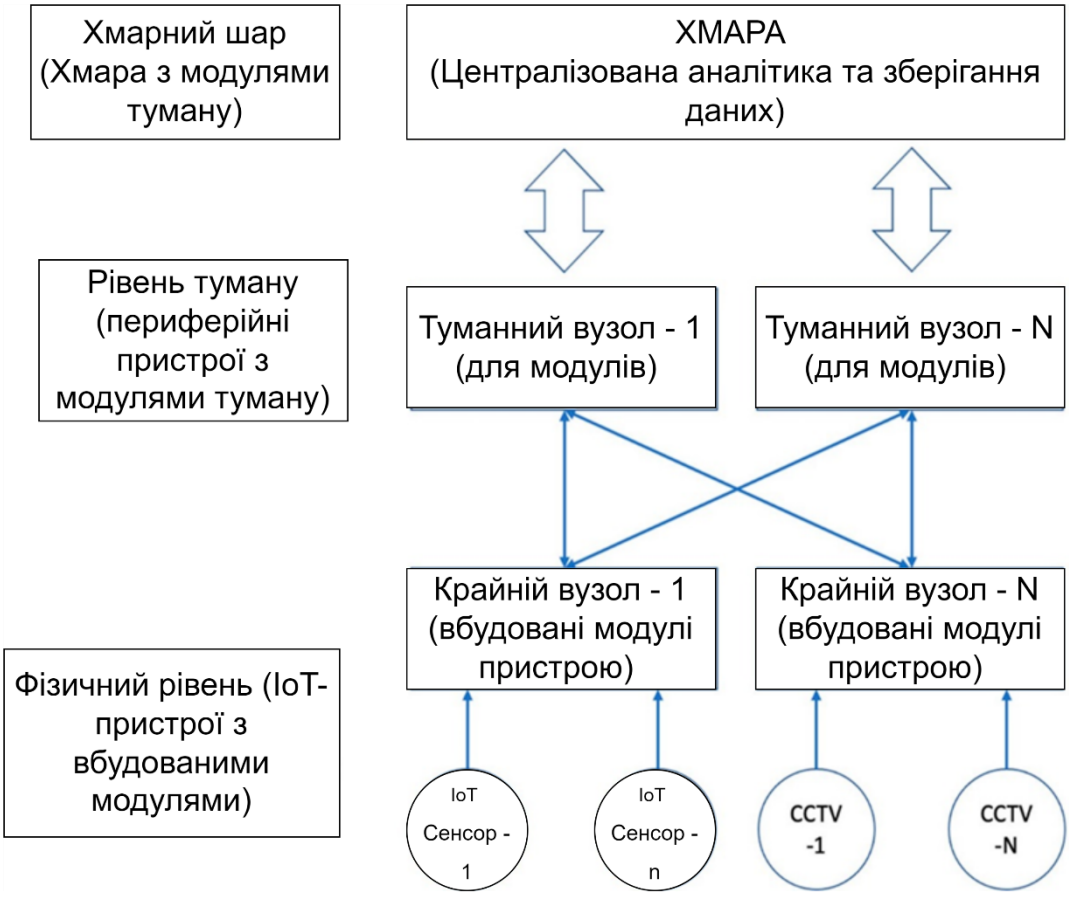


Рис. 4.3. Архітектура модуля Fog

Модуль шару туману

Компоненти модуля туманного шару мають ту саму концепцію, що й архітектура Open Fog, хоча поведінка пристроїв Edge у структурі відрізняється. Обчислення Fog полегшує попередню обробку даних ще до того, як вони потраплять у хмару, мінімізуючи час зв'язку, а також зменшуючи потребу у зберіганні величезних обсягів даних за допомогою фільтрації. Загалом, це відповідний підхід для додатків і послуг з IoT. Наш підхід тісно пов'язаний з архітектурою туманних обчислень. Граничні пристрої працюють в автономному режимі; отже, кожен вузол Fog (FN), який складається з периферійних серверів і периферійних пристроїв на рівні Fog, керує набором обчислювальних завдань. У даній архітектурі ми представили модуль аналізу даних і розвантаження для виконання наступних завдань:

- дані, згенеровані з пристроїв IoT, зібрані для аналізу, якщо це необхідно;

- ресурси завдання перевищують із даного периферійного сервера, тоді він виконуватиме розвантаження на інший крайовий сервер у приміщенні або в хмару.

Розподіл завдань і розвантаження є важливими діями під час процесу аналізу даних, оскільки від цього залежить завершення процесів.

Модуль туману [203] забезпечує методи аналізу даних і можливості для пристроїв Інтернету речей спілкуватися та співпрацювати з хмарним шаром і пристроями в шарі туману. Він надає можливості для передачі даних зі сторони туману в хмару та взаємодії з хмарою. Шлюз дозволяє кінцевим пристроям, які безпосередньо не підключені до Інтернету, мати доступ до хмарних сервісів. Хоча термін «шлюз» виконує певну функцію, орієнтовану на мережу, він також використовується для опису групи кінцевих пристроїв, які керують і обробляють дані від імені кластеризації пристроїв.

4.1.3. Аналіз даних у модулі Fog

На рис. 4.4 зображено аналітичну обробку даних поблизу джерела даних за допомогою модуля Fog за експоненціального збільшення обсягу та розмірів даних. Потокові дані локально аналізуються в модулі Fog, тоді як дані модуля Fog збираються та передаються в хмару для автономної аналітики та обробки даних. Модулі аналітики даних, розгорнуті в модулях Fog, періодично оновлюються на основі політик, прийнятих і переданих хмарною аналітикою. Оскільки необроблені дані попередньо обробляються, фільтруються та очищаються в модулі Fog, перед вивантаженням у хмарні модулі, кількість і розмір вивантажуваних даних менші, ніж дані, створені пристроями IoT. Крім того, аналітика в модулі Fog працює в режимі реального часу, а аналітика в хмарі – офлайн. Модуль Fog має обмежену обчислювальну потужність і потужність зберігання порівняно з хмарною стороною, однак обробка та керування на стороні хмари вимагає більшого часу затримки. Модуль Fog забезпечує високий рівень відмово-стійкості, оскільки завдання можуть бути передані іншим модулям Fog поблизу у разі збою. З появою ресурсного Інтернету речей, який дає змогу використовувати додатки з високою швидкістю передачі даних у реальному часі, кращим підходом здається переміщення аналітики до джерела даних і ввімкнення обробки в реальному часі. У майбутньому Fog-модуль може прийняти багато різних типів апаратних компонентів, таких як багатоядерний процесор, графічний процесор із високою

деталізацією порівняно з кластером подібних вузлів у хмарі.

Рис. 4.4. Аналіз даних на модулі Fog перед передачею в хмару

Розвантаження в модулі Fog

Проведені дослідження показують, що технологія обчислень на межі туману дає можливість подолати обмеження апаратного забезпечення для пристроїв кінцевого користувача, перевантажуючи інтенсивні обчислювальні завдання на розширені периферійні сервери для виконання. Виконання на периферійних серверах відповідає вимогам завдання та надає кінцеві результати кінцевим пристроям. Парадигма туманних обчислень наближає як обчислювальні, так і мережеві ресурси до користувача.. У модулі *Fog* розгортається мережева архітектура розвантаження завдань із кількома вузлами туману. Коефіцієнт продуктивності включатиме затримку кожного вузла туману. Ми запропоновуємо схему розвантаження, пам’ятаючи про вибір вузлів туману відповідно до метрики планування завдань, а потім розвантажимо завдання на вузли туману, які потребують мінімальної затримки у виконанні завдання. Кожного разу, коли обчислювальне завдання створюється на термінальному вузлі, кількість вузлів туману вибирається відповідно до вимог продуктивності та характеристик цих найближчих вузлів туману. Замість того, щоб обчислювати завдання локально, воно ділиться на кілька підзадач і розподіляються на ці вибрані туманні вузли для обчислення. Після цього результати обчислень передаються назад на термінальний

вузол. Очевидно, що найближчі туманні вузли з найпотужнішими обчислювальними можливостями будуть вибрані, коли ми прагнемо мінімальної затримки завдання за найкращої продуктивності. На рис. 4.5 зображено процес розвантаження інформації, зібраної в пристроях IoT [204] на основі нашого дослідження.

Хмарний шар головним чином відповідатиме за складні, ресурсозатратні завдання та оновлення правил для виявлення на шарі туману. Фаза збору даних є основним аспектом цих рішень, яка встановлює протоколи зв'язку між компонентами програмної платформи IoT. Глобальний аналіз даних і загальний моніторинг ресурсів. *Cloud* надає схвалену модель оплати за використання, коли модуль *Fog* стане атрибутом користувача. Залежно від програми Інтернету речей, у разі обмеженого доступу до живлення, *fog core* (туманного ядра) може працювати від батареї та має бути енергоефективним, тоді як хмара підтримується постійним джерелом живлення.

4.2. Проектування механізму довіри

При проектуванні механізму довіри розглянемо два випадки - пряма довіра та всебічна довіра між вузлами.

4.2.1. Пряма довіра між вузлами

Існує багато характеристик поведінки, за якими можна спостерігати, щоб оцінити стан довіри вузлів під час процесу взаємодії вузлів. Однак чим більше характеристик збирається, тим складнішою стає реалізація системи через деякі обмеження, які слід дотримуватися. До прикладу - споживання енергії, навантаження на мережу та інші. Ми вибираємо рівень втрати пакетів, рівень відмов маршруту та затримку пересилання як докази для оцінки стану довіри вузла.

Рівень втрати пакетів, $Trust_{packet}$, відноситься до співвідношення кількості пакетів даних, втрачених одержувачем, до загальної кількості пакетів даних у циклі зв'язку, що представляє тип доказу, який може вказувати на стан вузла або на те, чи вузол зламано. Частота помилок маршруту означає відношення кількості

пакетів маршрутизації, відкинутих одержувачем, до загальної кількості пакетів маршрутизації, надісланих відправником протягом певного проміжку часу, за яким можна судити про стан мережі. Затримка пересилання, $Delay_{forwarding}$, стосується інтервалу часу від отримання даних до пересилання даних, коли ретрансляційний вузол передає дані, що представляє тип доказу того, що вузол скомпрометовано або має серйозну несправність. Вихідний вузол може використовувати ці докази для встановлення прямих довірчих відносин на спільних вузлах. Більше того, спостережене значення поведінки вузла може коливатися зі зміною середовища та навантаження на мережу, тому значення історії довіри, $Trust_{history}$, додається до прямого обчислення довіри, щоб зменшити рівень неправильних оцінок звичайних вузлів і непотрібну витрату мережевих ресурсів. Формула прямої довіри представлена у вигляді (4.1).

$$Trust_{direct} = (w_1 Trust_{packet} + w_2 Trust_{history}) \times Delay_{forwarding}, \quad (4.1)$$

де $Delay_{forwarding}$ є свідченням серйозної проблеми безпеки, яка вважається тим, що вузол ретрансляції змінив дані. Коли часовий інтервал перевищує порогове значення, значення $Delay_{forwarding}$ встановлюється на 0, в іншому випадку - 1. Якщо виникає виняткова ситуація $Delay_{forwarding}$, значення $Trust_{direct}$ дорівнює 0. В іншому випадку значення $\mathrm{Trust}_{\mathrm{direct}}$ визначається $Trust_{packet}$ і $Trust_{history}$ на основі зваженого алгоритму.

Для зважених значень $w_1 + w_2 = 1$. Щоб зменшити енергоспоживання вузла при передачі даних, період виявлення довіри між вузлами максимізується в прийнятному діапазоні. У цьому випадку значення довіри може стати занадто старим, щоб справді відображати поточний стан довіри вузла. Отже, ми зменшили вагу $Trust_{history}$ за формулою (4.2).

$$w_2 = real_1 \times Period_{network} \times exp(-real_2 \times Period_{network}), \quad (4.2)$$

де $Period_{network}$ - це інтервал часу від останнього оновлення до теперішнього часу; $real_1$ і $real_2$ - два дійсних числа, які встановлюються під час ініціалізації.

Теорема 1. *Чим більше значення* $Period_{network}$, тим швидше воно сходиться до $Trust_{packet}$.

Доведення. По-перше, ми беремо похідну функції, щоб перевірити її тенденцію

зміни. Тоді знаходимо область спаду цієї функції. І на кінець, для налаштування ваги $Period_{network}$ вибирається відповідну область спадання.

$$(w_2)' = (real_1 \times Period_{network} \times exp(-real_2 \times Period_{network}))' = real_1 \times (1- real_2 \times Period_{network}) \times exp(-real_2 \times Period_{network})\ (w_2)` = 0 \quad (4.3.)$$

Then $Period_{network} = (1/real_2)$

Ми можемо встановити, що крива іде вниз, якщо $Period_{network}$ перевищує $(1/real_2)$. Крива різко падає в передній частині і стабільно знижується в останній частині. Щоб досягти ідеального результату, значення $real_2$ можна встановити в межах [0,7-1], а значення $real_1$ встановлюється відповідно до $real_2$, що може призначати різні зважені значення для $Trust_{history}$ відповідно до різних *Period*.

4.2.2. Всебічна довіра між вузлами

На цьому рівні вихідний вузол запитує значення рекомендацій у своїх довірених суміжних вузлів, коли він знаходить деякі винятки суміжних вузлів. Тим часом вихідний вузол також надсилає ці винятки на шар туману для аналізу стану довіри кожного вузла в цій області. Якщо ці аномальні вузли визначені як зловмисні, шар туману повідомить голову кластера, щоб ізолювати ці зловмисні вузли.

Винятки в WSNs поділяються на три категорії: виняток частоти помилок маршруту, виняток затримки пересилання та виняток значення різниці. Помилка маршрутизації є нормальним явищем у мережах *WSN*s, але вона вважається винятком, коли частота помилок маршруту досягає порогового значення за певний період часу. Коли затримка пересилання перевищує порогове значення, виникає виняток затримки пересилання. Виняток значення різниці полягає в тому, що значення різниці між новим значенням довіри та історичним значенням довіри виходить за розумний діапазон. Загальна формула розрахунку довіри до рекомендацій представлена формулою (4.4)

$$Trust_{recommendation} = \sum_{i \in set(neighbor)} w_{i(i,j)} \times Trust_{(j,k)}\ , \quad (4.4)$$

де *set(neighbor*) - набір довірчих вузлів вихідного вузла;

$Trust_{(j,k)}$ - значення довіри вузла *j* до вузла *k*.

Однак вихідний вузол має різні значення довіри для різних суміжних

вузлів. У цьому випадку повинні бути певні механізми для належного зменшення впливу вузлів з низькою продуктивністю. Тут ми сортуємо таблицю довіри вихідного вузла від малого до великого за значеннями довіри, а потім обчислюємо зважене значення кожного сусіднього вузла за допомогою арифметичної прогресії за формулою (4.5).

$$w_{i(i,j)=\frac{i}{\sum_1^n i}=2\times\frac{i}{n(n+1)}}, \tag{4.5}$$

де параметр *i* - це значення розташування вузлів у впорядкованій таблиці довіри, а параметр *n* - номер вузла *set*(*neighbor*).

Точніше, рекомендація $Trust_{recommendation}$ дає вихідному вузлу консультативний висновок, і остаточне рішення вихідного вузла базується на $Trust_{direct}$ і $Trust_{recommendation}$, і описується формулою (4.6). Зважене значення $Trust_{direct}$ більше, ніж $Trust_{recommendation}$, і $w_3 + w_4 = 1$.

$$Trust_{synthesis} = w_3 \times Trust_{direct} + w_4 \times Trust_{recommendation} \tag{4.6}$$

Теорема 2. *Невелика частка шкідливих вузлів рекомендацій не може прийняти рішення*

$Trust_{recommendation}$.

*Доведення***.** Найбільш надійна нота вихідного вузла має найбільше зважене значення, яке становить $\frac{2}{n+1}$. Значення різниці між зваженим значенням двох суміжних вузлів у таблиці довіри становить $\frac{2}{n(n+1}$. Коли *n* знаходиться в [*2,3,…n*], відповідне найбільше зважене значення в [$\frac{2}{2+1}$, $\frac{2}{3+1}$ … ,$\frac{2}{n+1}$].

Чим більше n, тим менше зважене значення кожної ноти.

4.2.3. Аналіз даних у шарі туману

Існує три типи аналізу даних у шарі туману. Перший тип відновлює помилкові вузли та виявляє атаки на приховані дані, що базується на таблицях довіри, історичних даних сенсорів і топології мережі. Другий тип перевіряє наявність зловмисних вузлів або зловмисних вузлів рекомендацій після отримання винятків від WSNs, що базуються на таблицях довіри, таблицях рекомендацій, історичних даних сенсорів і топології мережі. Третій тип

стосується довіри до крайових вузлів, яка базується на таблицях довіри та кореляції даних сенсорів.

Усі вузли сенсорів надсилають значення змін довірчої таблиці разом із даними сенсорів до шару туману протягом певного періоду, а вихідний вузол надсилає таблицю рекомендацій разом із даними сенсорів до шару туману після завершення обчислення довіри рекомендацій. Рівень туману періодично аналізує глобальний стан довіри кожного вузла та визначає, чи є вузли з неправильною оцінкою та приховані атаки на дані. Крім того, на основі цього глобального стану довіри вузлів ми можемо передбачити певний стан мережі, як-от навантаження мережі, залишкова енергія вузлів тощо.

Деякі шкідливі вузли надають неправильні дані сенсорів, щоб спонукати користувачів приймати неправильні рішення. Ці вузли важче знайти, оскільки вони поводяться нормально, коли спілкуються з іншими вузлами. У межах однієї області чи кластера спостерігаються певні явища кореляції даних, наприклад дані сенсорів з кількох вузлів в одному географічному положенні подібні, дані сенсорів з кількох вузлів у різних географічних позиціях демонструють поступовість, дані сенсорів із кількох вузлів рухаються разом мають кореляцію траєкторії. Шар туману може одночасно обробляти дані сенсорів з кількох вузлів і аналізувати наявність шкідливих вузлів за допомогою деяких індикаторів явища кореляції даних, таких як тенденція зміни, відмов-стійкість інтервал, подібна траєкторія тощо. Шар туману знаходиться ближче до *WSN*s, тому затримка виявлення атак на приховані дані прийнятна. Ми виконуємо багато-шляхову операцію для аналізу даних сенсорів з різних вузлів. Структура процесу показана на рис. 4.6. Тут ми в основному розглядаємо вузли, які реалізують ту саму функцію в тому самому географічному положенні. Залежності представлені системою (4.7).

$$Array = \begin{cases} Count_{crest} \cup degree, \frac{X2_i - X1_i}{Y2_i - Y1_i} > 0 \\ Count_{trough} \cup degree, \frac{X2_i - X1_i}{Y2_i - Y1_i} > 0 \end{cases} \qquad (4.7)$$

Тут ми використовуємо масив для зберігання даних про вершини/впадини.

$Count_{crest}$ вказує на вершину кривої даних сенсора, яка описується формулою (4.7). $Count_{trough}$ вказує на спад кривої даних сенсора і описується формулою (4.7). Градус - це значення різниці між двома сусідніми даними сенсора. Визначення вершини/впадини є безперервним двома негативними/позитивними значеннями після позитивного/негативного значення, а безперервний пік/впадина виникає, коли значення зміни даних сенсора дорівнює нулю після запису вершини/впадини. У кожен момент часу у масиві даних записується значення стану (1, -1, 0) і значення градуса. Крайові вузли мають менше зв'язків з іншими вузлами порівняно з внутрішніми вузлами. Ми встановлюємо виявлення коротшого періоду для крайових вузлів у *WSN*. Крім того, туманний шар буде сканувати та аналізувати стан довіри крайових вузлів за коротший період.

4.2.4. Встановлення довірчих відносин між SSPs і CSPs

Довірчі відносини між *CSPs* і *SSPs* поділяються на дві частини. Одна - це довірчі відносини між *CSPs* і *SSPs,* а друга - довірчі відносини між *SSPs* і *CSPs*.

Довірчі відносини CSPs з SSPs

Що стосується *CSPs*, вони очікують, що дані з *SSPs* повинні відповідати деяким вимогам, таким як відсутність підробки, цілісність, своєчасність і точність. Однак користувач послуг може не вимагати, щоб постачальники послуг задовольняли всі потреби, тобто постачальники послуг повинні відповідати лише конкретним вимогам користувача послуг. Таким чином, повинні існувати деякі спеціальні механізми рекомендацій для пошуку *CSPs*, які пропонують послуги в особливих аспектах. Туманні обчислення можуть добре впоратися з цими проблемами. Довірена третя сторона на основі туманних обчислень може забезпечити надійність *SSP* за допомогою трьох частин, як показано у формулі (4.8).

$$Trust_{SSPs} = w_5 Trust_{service} + w_6 Trust_{WSNs} + w_7 Trust_{CSPs,} \quad (4.8)$$

де $Trust_{service}$ - це значення довіри для параметрів служби. Перед транзакцією обслуговування *SSPs* і *CSPs* узгоджують стандарти параметрів обслуговування. Потім туманний шар відстежує ці сервісні параметри під час транзакції в режимі реального часу та порівнює ці сервісні параметри зі стандартними значеннями.

Якщо значення контрольованого параметра послуги знаходиться в прийнятному діапазоні, запис цього параметра дорівнює 1, якщо ні то дорівнює 0. Нарешті, $Trust_{service}$ обчислюється за допомогою різних зважених значень параметрів послуги. $Trust_{WSNs}$ - це значення довіри для *WSN*s, яке базується на записах інформації про винятки *WSN*s. Якщо WSNs має більше винятків у транзакції, йому буде надано нижче значення. $Trust_{CSPs}$ – це тип довірчого значення, яке обчислюється на основі інформації про записи інших *CSPs* у шарі туману. Є два кроки, які використовуються для прийняття рішення щодо вибору *CSPs*: розрахунок загальної рекомендації $R_{general}$ і розрахунок аналогічної рекомендації $R_{similar}$. Деякі *CSPs* вибираються в список кандидатів, записи про обслуговування яких містять усі запитувані параметри послуг. $R_{general}$ призначає всі вибрані *CSPs* до різних наборів, які класифікуються за кількістю надлишкових параметрів. Потім значення довіри кожного *SSPs* окремо обчислюється в різних наборах. Нарешті, деякі аномальні *CSPs* виключаються зі списку кандидатів відповідно до правила зміни значення довіри між різними наборами. $R_{similar}$ - це оптимальна стратегія вибору, яка має багато принципів, які можуть вибрати користувачі сервісу. $Trust_{CSPs}$ обчислюється за допомогою цих вибраних *CSPs*. w_5, w_6 *і* w_7 - це три зважені значення, які встановлюються під час ініціалізації залежно від різних вимог користувачів, а $w_5 + w_6 + w_7 = 1$. Значення $Trust_{service}$, $Trust_{WSNs}$ і $Trust_{CSPs}$ знаходиться в діапазоні від 0 до 1.

Для *SSPs* вони очікують, що послуги, які надаються *CSPs*, повинні відповідати певним критеріям, таким як надійність, безпека, зручність, керованість, стабільність тощо. Ці показники важливі для *SSPs* для встановлення довіри до *CSPs*. Шар туману може контролювати ці показники в реальному часі. Довірчі відносини між *SSPs* і *CSPs* складаються з двох частин, як показано у формулі (4.9).

$$Trust_{SSPs} = w_8 Trust_{service1} + w_9 Trust_{SSPs} \quad (4.9)$$

Де $Trust_{service1}$ - значення довіри параметрів служби *CSPs*, подібне до $Trust_{service}$. $Trust_{SSPs}$ обчислюється за допомогою деяких вибраних *SSPs*, процес вибору яких подібний до $Trust_{CSPs}$. У туманному шарі є деякі бази даних, які

використовуються для зберігання службових записів певного часу. w_8 і w_9 - два значення ваги, які встановлюються під час ініціалізації залежно від різних вимог користувачів, а $w_8 + w_9 = 1$. Значення $Trust_{service1}$ і $Trust_{SSPs}$ знаходиться в діапазоні від 0 до 1.

4.3. Моделювання оцінювання

Платформою експерименту була платформа MATLAB R2016b. Існує шість кластерних структур з більш ніж трьома сотнями вузлів, які випадковим чином розгорнуті на рівні бездротових сенсорних мереж. Кожен кластер розділений на три рівні, де зовнішній шар має більше вузлів, ніж внутрішній. У кожній структурі кластера голови кластера можуть отримувати пакети даних сенсорів від шести вузлів одночасно. Максимальний час затримки від *WSN*s до шару туману встановлюється як сім циклів зв'язку. Ці параметри наведено в таблиці 4.1. У шарі туману є деякі записи, близькі до реальних.

Таблиця 4.1

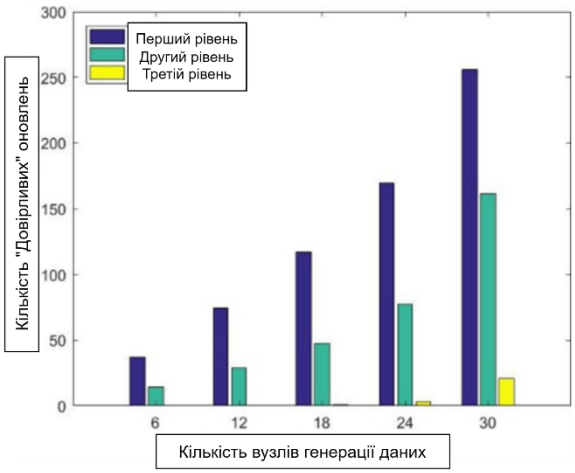
Параметри моделювання

Параметри	Значення
Мережевий протокол	Алгоритм сходової дифузії
Кількість кластерів	8
Кількість голів кластера	46
Кількість вузлів кластера	350
Кількість рівнів	4
Максимальна затримка	10

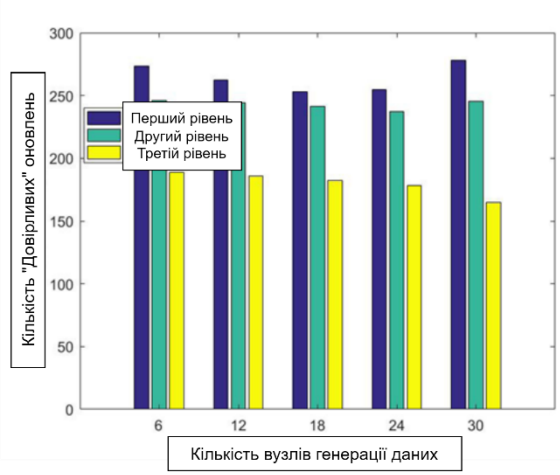
Існує два типи механізмів довіри: періодичне та неперіодичне оновлення. Для неперіодичного оновлення вузли оновлюють стан довіри своїх суміжних вузлів під час виявлення ненормальної поведінки. Є деякі недоліки в неперіодичному оновленні, зокрема, занадто мало уваги до крайових вузлів і старіші стани довіри вузлів, результат експерименту показано на рис. 4.6(а). Неперіодичне оновлення не може вчасно виявити шкідливі вузли. Для періодичного оновлення вузли оновлюють значення довіри своїх сусідніх вузлів після завершення періоду часу. Існують також деякі недоліки в періодичному оновленні, такі як використання забагато пам'яті та обчислювальних ресурсів, зниження продуктивності

мережевого зв’язку тощо, результат експерименту якого показано на рис. 4.6(б). Запропонований дизайн базується на періодичному оновленні.

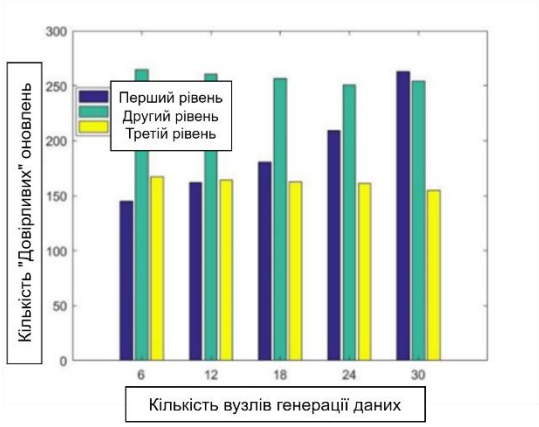
Було встановлено, що цикл оновлення довіри на зовнішньому рівні такий самий, як і періодичне оновлення, яке, очевидно, можна побачити на рис. 4.6(б) і рис. 4.6(с). Ми можемо подовжити цикл оновлення довіри на внутрішньому рівні за допомогою *Fog Computing*, уникнувши додаткових витрат ресурсів на періодичне виявлення, як показано на рис. 4.6(с). Три експериментальні результати розглядають кількість оновлення довіри на кожному рівні *WSN*, а навантаження на мережу збільшується на основі збільшення кількості вузлів генерації даних. На рис. 4.6(д) показано загальну кількість разів оновлення трастів за коротший час тестування. З цих експериментальних результатів ми можемо отримати три результати.



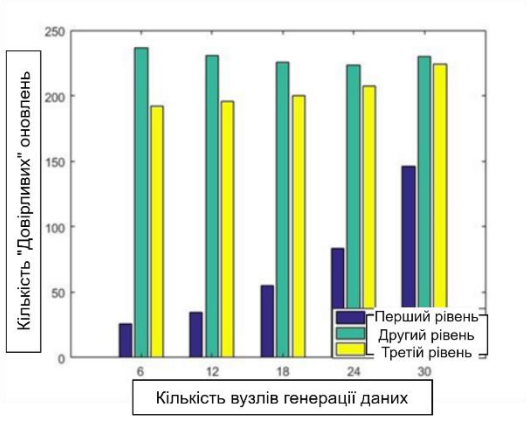
а). Неперіодичне оновлення



б). Періодичне оновлення



с). Запропонований дизайн



д). Результат після обробки даних

Рис. 4.6. Порівняння трьох схем

1. Для неперіодичного оновлення кількість часу оновлення довіри поступово збільшується з більшою кількістю випадкових вузлів, вибраних для передачі даних.

2. Для періодичного оновлення він підтримує стабільний стан. Однак на осі абсцис є невелике скорочення з 6 до 24, причина в тому, що пряме довірче оновлення зменшує кількість періодичних оновлень. Для нашого дизайну ми можемо отримати більшу перевагу, коли цикл оновлення подовжується. Коли *WSN* перевантажується, пропускна здатність мережі зменшується, а час оновлення довіри збільшується через часті перебої маршрутизації. Порівняно з періодичним оновленням, наш дизайн може економити мережеву енергію та підтримувати продуктивність мережі за рахунок скорочення кількості періодичних оновлень.

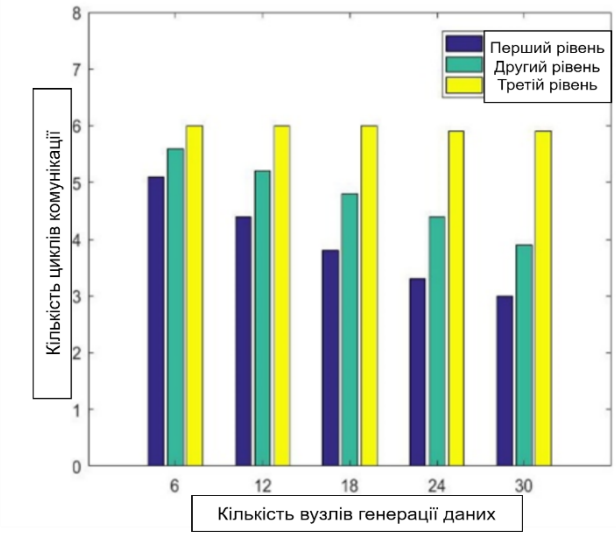
4.3.1. Швидкість виявлення шкідливих вузлів

Немає потреби в надто частому оновленні довіри, оскільки внутрішні атаки відбуваються в певний час, і часте оновлення займе більше передавання та обчислювальних ресурсів. Ми порівнюємо швидкість виявлення шкідливих вузлів між нашим дизайном і періодичним оновленням.

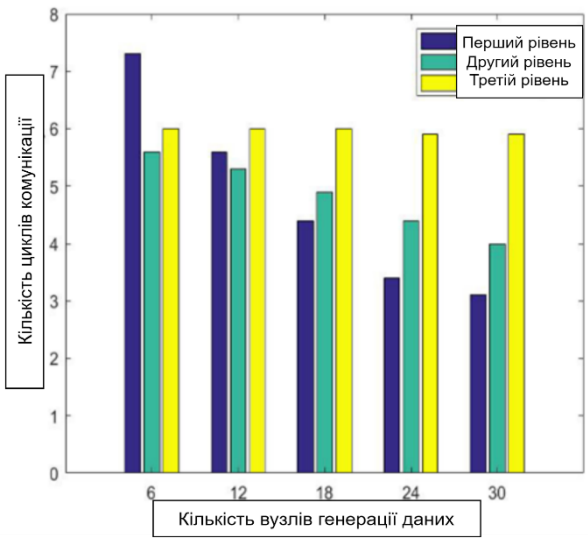
Шкідливі вузли можна виявити з двох частин: рівня бездротової сенсорної мережі та рівня туману. Оскільки час затримки в шарі туману трохи довший, ніж на рівні бездротової мережі, виявлення шкідливих вузлів у шарі туману вводиться як допоміжне виявлення. Швидкість виявлення періодичного оновлення показано на рис. 4.7(а), а на рис. 4.7(б) показано швидкість виявлення нашого дизайну.

В експерименті розміщено шкідливі вузли на різних рівнях мережі при ініціалізації, які витрачають більше часу, ніж у процесі роботи механізму. Результати експерименту показують, що швидкість виявлення шкідливих вузлів зростає зі збільшенням навантаження на мережу, за винятком зовнішнього рівня, оскільки стан довіри вузлів оновлюється частіше, коли навантаження на мережу стає більшим. На рис. 4.7 (с) випадковим чином розміщені шкідливі вузли на трьох рівнях, що вказує на більш інтуїтивну тенденцію до зниження. Незважаючи на певні проблеми із затримкою виявлення швидкості, ми можемо скористатися перевагами *Fog Computing* (туманні обчислення), щоб отримати повний стан

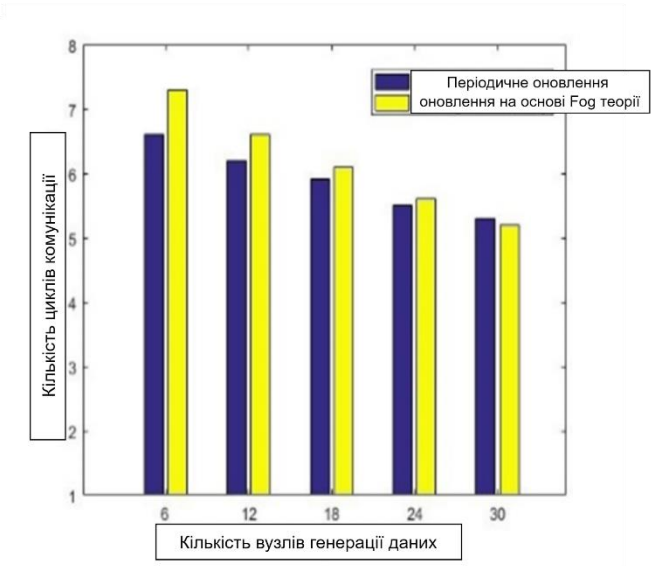
надійності *WSN*s за допомогою деяких аналізів даних, таких як виявлення атак на приховані дані та виявлення аномалій значень монітора.



а). Періодичне оновлення



в). Запропонований дизайн



с). Результат після обробки даних

Рис. 4.7. Порівняння двох схем

4.3.2. Відновлення вузлів неправильного оцінювання

Сервісні можливості вузлів можуть коливатися зі зміною навколишнього середовища та власним споживанням електроенергії. Таким чином, деякі вузли демонструють незвичайну поведінку в деяких випадках, наприклад, найбільша зміна середовища, низький заряд батареї та тимчасові перебої. Майже всі механізми довіри не розглядають ці випадки, ми вирішуємо ці проблеми в шарі туману шляхом аналізу даних.

В даноиу експерименті кількість вузлів генерації даних становить 18 в одному циклі зв’язку, і ми додаємо одну умову кожні п’ять циклів зв’язку після ініціалізації мережі. Як показано на рис. 4.8, є деякі шкідливі вузли, додані до мережі під час ініціалізації (1). Після шести циклів зв’язку періодичне оновлення та наш розроблений дизайн одночасно знаходять шкідливі вузли. Ми додаємо загальні шкідливі вузли в (2). Після (2) шкідливі вузли виявляються обома механізмами довіри після меншої кількості циклів зв’язку. Програма очищення шкідливих вузлів виконується в (3), а середовище змінюється в (4).

Після зміни середовища вузли, чутливі до середовища, демонструють аномальну поведінку, яку можна неправильно оцінити як шкідливі вузли. У цей період шкідливих вузлів фактично немає. Для цих вузлів неправильної оцінки шар туману аналізує, чи є ці вузли справжніми зловмисними вузлами за таблицями довіри, таблицями довіри рекомендацій, топологією мережі та довгостроковими даними сенсорів. Після прийнятної затримки наша схема може відновити вузли помилкової оцінки перед програмою очищення в (5).

Вибір для SSPs

Перед обчисленням $Trust_{SSPs}$ існують певні механізми підрахунку балів, наприклад, оцінка знижується на одну оцінку, коли параметри послуги на одиницю менші за параметри обслуговування вимог. У таблиці 4.2 відображено деякі важливі параметри, які показано як приклад. Вимога до послуги зберігає записи параметрів послуги *CSP*, якілися у попередній транзакції. Запис взаємодії зберігає *SSP*s та їхні значення довіри, які надали дані CSP. Запис рекомендацій зберігає *CSP*s. Час реєстрації зберігає час існування *CSP* у шарі туману.

Шар туману вибирає записи про послуги, які знаходяться в прийнятному діапазоні часу, як показано в таблиці 4.2, де цілісність і своєчасність знаходяться в різних розмірах наборів вимог до обслуговування, таких як set1 (*CSP*1, *CSP*2), set_2: (*CSP*3, *CSP*4), set_3: (*CSP*5). Існує певний потенційний зв’язок відповідності, який є *SSP*. 1) *SSP*1 (без втручання, цілісність). 2) *SSP*2 (цілісність, своєчасність). 3) *SSP*3 (без втручання, своєчасність). 4) *SSP*4 (без втручання, цілісність, точність). 5) *SSP*5 (цілісність, своєчасність).

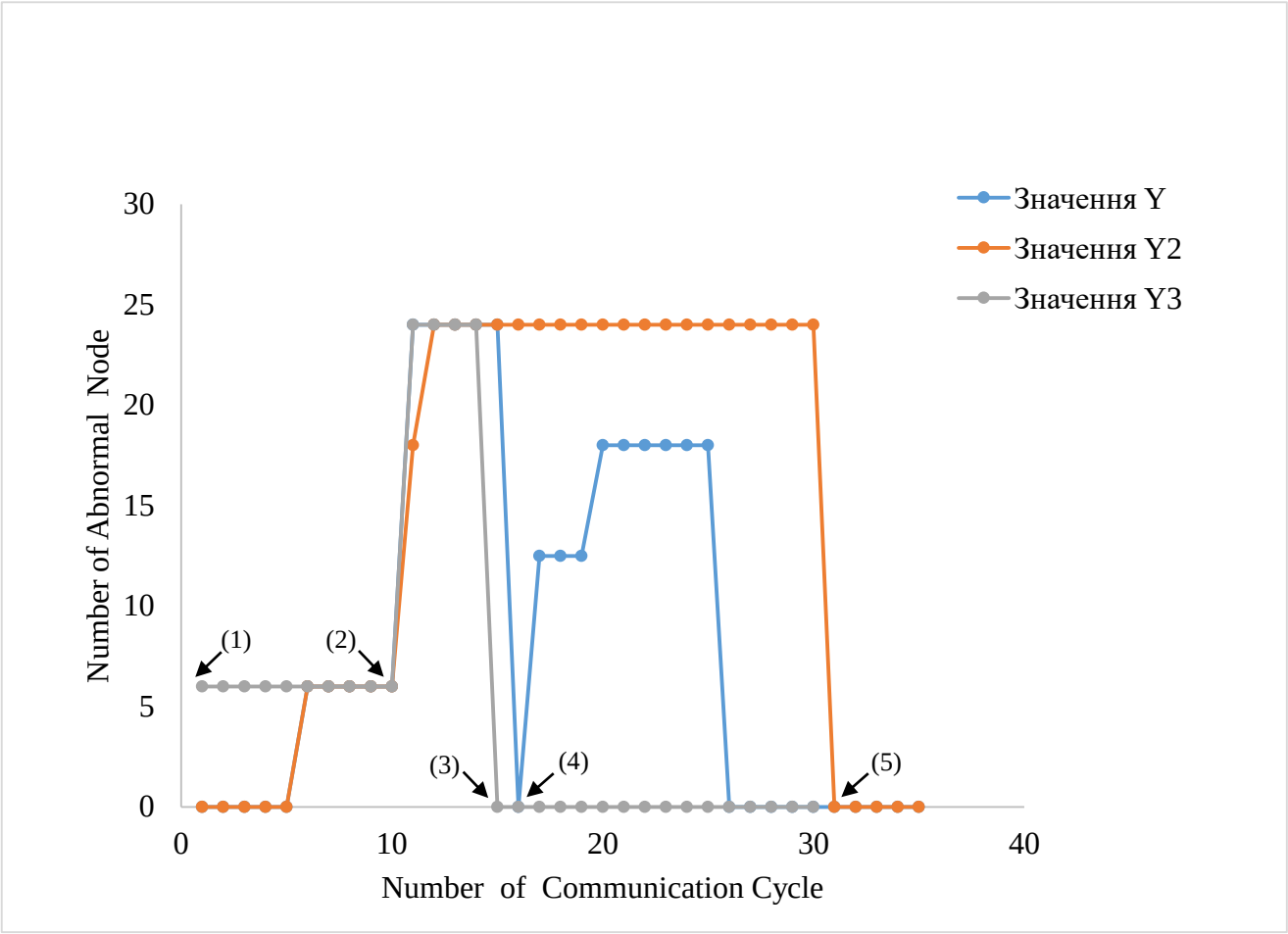


Рис. 4.8. Відновлення вузлів неправильного оцінювання

Таблиця 4.2

Інформаційний запис SSPs у шарі туману

	Сервіс вимога	Взаємодія записів	Рекомендований запис	Кількість прийняти	Час очікування
CSP1	Точність цілісності	SSP1 (605 SSP2 (90) SSP4 (96)	CSP2 CSP3 CSP4	96/97	15
CSP2	Точність цілісності	SSP2 (90) SSP4 (94) SSP5 (93)	CSP1 CSP3 CSP5	60/80	9
CSP3	Без втручання точність цілісності	SSP3 (75) SSP4 (94) SSP5 (85)	CSP1 CSP2 CSP4 CSP5	68/72	17
CSP4	Точність без втручання	SSP1 (86) SSP3 (72)	CSP1 CSP2 CSP5	26/29	7
CSP5	Без втручання цілісність своєчасність точність	SSP1 (74) SSP2 (72) SSP3 (72) SSP4 (85)	CSP3 CSP4	72/73	2

Під час розрахунку $R_{general}$ середнє значення довіри кожного *SSP* обчислюється в різних наборах, наприклад [*SSP*1(60), *SSP*2(90.5), *SSP*4(96.5), *SSP*5(92)] у наборі 1, [*SSP*1(84), *SSP*3(71), *SSP*4(93), *SSP*5(82)] у наборі 2 та [*SSP*1(77.5), *SSP*2(70), *SSP*3(70), *SSP*4(82)] у наборі 3. Значення довіри одного *SSP* у меншому наборі повинно бути більше або дорівнювати значенням у більшому наборі. Таким чином, за цим правилом ми можемо знайти деякі ненормальні оцінки, наприклад *CSP*1 може бути неправильним вибором.

Після видалення нестандартного рекомендованого виразу розглядається оптимальний вибір, наприклад знайомство, популярність і ризик. На знайомство впливають записи про рекомендації, розташування *SSP* та репутація. Популярність відноситься до прийнятої кількості рекомендацій одного *CSP*. Ризик зосереджується на тому, чи будуть певні втрати, якщо обрано молодого рекомендованого виразу. Наприклад, *CSP*3 має більше сервісів рекомендацій, ніж *CSP*2, і його кількість прийнятих рекомендацій більша, ніж *CSP*2. Крім того, *CSP*3 залишається в шарі туману довше, що свідчить про більшу достовірність, ніж *CSP*2. Нарешті обираються остаточні рекомендованого виразу.

Вибір для CSPs

Для вибору для CSPs його таблиця запису інформації подібна до вибору для SSPs, яка також містить вимогу до обслуговування, запис взаємодії, запис рекомендації, кількість прийнятих і час реєстрації. Для Trustservice1 він також генерується шаром туману на основі порівняння контрольованих параметрів обслуговування та стандартних параметрів обслуговування. Для Trust SSPs процес відбору подібний до процесу SSPs [205].

Висновки до розділу 4

Оптимізовано метод довіри на основі туману для компенсування вказаних недоліків та вирішення існуючих проблем споживачам цих мережних ресурсів для застосування у системах з мультисенсорною конфігурацією. Показано, що довіра щодо поведінки між вузлами встановлюється на рівні бездротових сенсорних

мереж, а довіра даних вузлів і об’єктів – у шар туману. Завдяки детальнішому аналізу даних у шарі туману ми можемо відстежувати стан довіри всієї мережі, виявляти атаки на дані та відновлювати вузли неправильної оцінки. Крім того, шар туману може бути побудованим як надійна третя сторона. Результати експерименту показують, що запропонований нами механізм довіри має певні переваги в деяких відношеннях, а саме - зменшення споживання енергії, забезпечення довірчого стану граничних вузлів і мережі, виявлення деяких прихованих атак на дані та відновлення вузлів з неправильною оцінкою.

Результати експерименту показують, що запропонований метод довіри має певні переваги в деяких відношеннях, таких як зменшення споживання енергії, забезпечення довірчого стану граничних вузлів і мережі, виявлення деяких прихованих атак на дані та відновлення вузлів з неправильною оцінкою.

РОЗДІЛ 5. ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДІВ ТА ЗАСОБІВ ІНТЕЛЕКТУАЛІЗАЦІЇ У ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМАХ

Запропоновані у роботі методи та засоби забезпечення передачі інформації у бездротових сенсорних мережах мають ряд практичних застосувань в різноманітних інформаційно-вимірювальних системах.

5.1. Системи для визначення положення тіла людини

При проектуванні системи було застосовано метод динамічного пошуку помилок в інформаційно-вимірювальній системі.

Спроектовано та досліджено систему для визначення положення тіла людини у віртуальному світі. Система складається з двох незалежних підсистем: підсистеми збору, обробки та передачі даних про положення тіла людини та підсистеми візуалізації цих даних у віртуальному світі. Загалом система вимірює положення тіла людини, обробляє ці дані, формує набір команд для віртуального світу та може надсилати ці команди на шолом віртуальної реальності для обробки. Запропоновано схеми системи та алгоритм її функціонування. Досліджено систему при використанні кватерніонів та кутів Ейлера. Найеефктивніше система працює при використанні кватерніонів.

Багато робіт зосереджені на створенні кіберфізичних систем для віртуальної реальності [206] з великою кількістю давачів для тренування [207], проектування роботизованих механізмів [208], імітаторів кисті рук [209], для медичних застосувань [208], у машинному навчанні [210], тощо. Однак найбільш поширено ці розробки зарекомендували себе при взаємодії з віртуальними комп’ютерними системами.

Зважаючи на все вище сказане, нами спроектовано та досліджено безпровідну систему для визначення положення тіла людини у віртуальній реальності. Запропонована система проводить виміри положення тіла у реальному світі, обробляє дані та проектує їх візуалізацію у віртуальний світ.

Структурна схема спроектованої системи для визначення положення тіла людини у віртуальному світі зображена на рис. 5.1. Вона складається з двох

незалежних підсистем: перша підсистема – несе відповідальність за збір, обробку та передачу даних про положення тіла людини, а друга підсистема – за візуалізацію цих даних у віртуальному світі.

Рис. 5.1. Структурна схема проектованої системи для визначення положення тіла людини у віртуальному світі

Для побудови системи збору, обробки та передачі накопичених даних використано мікроконтролер ESP8266. Визначення положення частин тіла людини базуватиметься на показниках гіроскопа та акселерометра, що знаходяться на модулі GY-521. Ці дані вимірюються та передаються на мікроконтролер, де оброблятись за певним алгоритмом. З метою підвищення точності показників, здійснюється просте фільтрування за методом Калмана. Обробивши та накопичивши ці дані, мікропрограма упакує ці дані. На наступному кроці дані відправляються на шолом віртуальної реальності за допомогою технології Wi-Fi.

Підсистема, яка відповідає за відображення отриманих даних у віртуальному світі керується власним центральним процесором, а також забезпечена графічним процесором та модулем безпровідного з’єднання. В підсистемі збору та передачі даних розташовані всі необхідні сенсори: акселерометри та гіроскопи. Окремим блоком є шолом віртуальної реальності,

який надаватиме можливість візуалізації рухів користувача для перегляду їх у віртуальному світі.

Для розробки програмного забезпечення підсистеми визначення положення частин тіла було використано спеціалізоване середовище розробки програмного забезпечення Arduino IDE для сімейства програмно-апаратних платформ Arduino. Для розробки програмного забезпечення підсистеми візуалізації отриманих даних використано середовище розробки 3D застосунків Unity3D. Це середовище дозволить перенести рухи людини в реальному світі у віртуальний шляхом заміни кінцівок на відповідні їх 3D-моделі. Також це дає можливість взаємодіяти із віртуальним об'єктами та перенестись у віртуальний світ.

Алгоритм роботи проектованої системи зображено на рис. 5.2.

Рис. 5.2. Граф-схема однієї ітерації алгоритму для визначення положення тіла людини

Відповідно до приведеної блок-схеми ітерації алгоритму після ініціалізації, система починає зчитувати покази з акселерометрів та гіроскопів модулів визначення положення тіла людини. Потім відбувається формування пакету даних, що буде передано до шолома. Це здійснюється у декілька етапів, перш за все дані фільтруються відповідно до попередніх даних. Для цього використовується примітивна реалізація фільтру Калмана та функції, які обмежують діапазон. Відфільтровані дані розміщуються у певному порядку, що дозволяє швидко ними оперувати на стороні приймача. Наступним кроком є передача пакету даних на шолом віртуальної реальності шляхом безпровідного інтерфейсу. Кожному пакету присвоюється ідентифікатор, щоб розташувати його у правильній послідовності. Задачею програми, яка запущена на шоломі віртуальної реальності, є прийом пакету даних, де кожному із показників буде присвоєний відповідним віртуальним моделям. А сам кут повороту кожної осі модуля, що закріплений на реальні частині тіла людини, буде призначено віртуальній моделі-аналогу частини тіла люди у віртуальному світі. Потім програма на шоломі віртуальної реальності візуалізує положення тіла людини у віртуальному світі.

Для можливості передачі даних через вбудований на плату мікроконтролера Wi-Fi та отримання доступу до мережі інтернет використано бібліотеку ESP8266, яка дозволяє досить просто та швидко створити інтернет з’єднання, відправляти та приймати байти інформації, що саме й потрібно у проектованій спеціалізованій системі для визначення положення тіла людини у віртуальному світі.

Встановлено, що розроблений алгоритм займає лише 19 % вільного простору в постійній пам’яті мікроконтролера та 20% оперативної пам’яті. Це означає що вибрана апаратно-програмна платформа Wemos D1 mini повністю задовольняє необхідну кількість ресурсів, які необхідні для виконання алгоритму керування спеціалізованою системою для визначення положення тіла людини у віртуальному світі.

Як зазначалося вище, для візуалізації положення тіла людини використовується середовище Unity3D. Для коректної роботи створено нову сцену та додано на неї об’єкт, який буде відповідати за приймання та передачу даних із

системою позиціонування частин тіла людини. Результат цих дій показано на рис. 5.3.

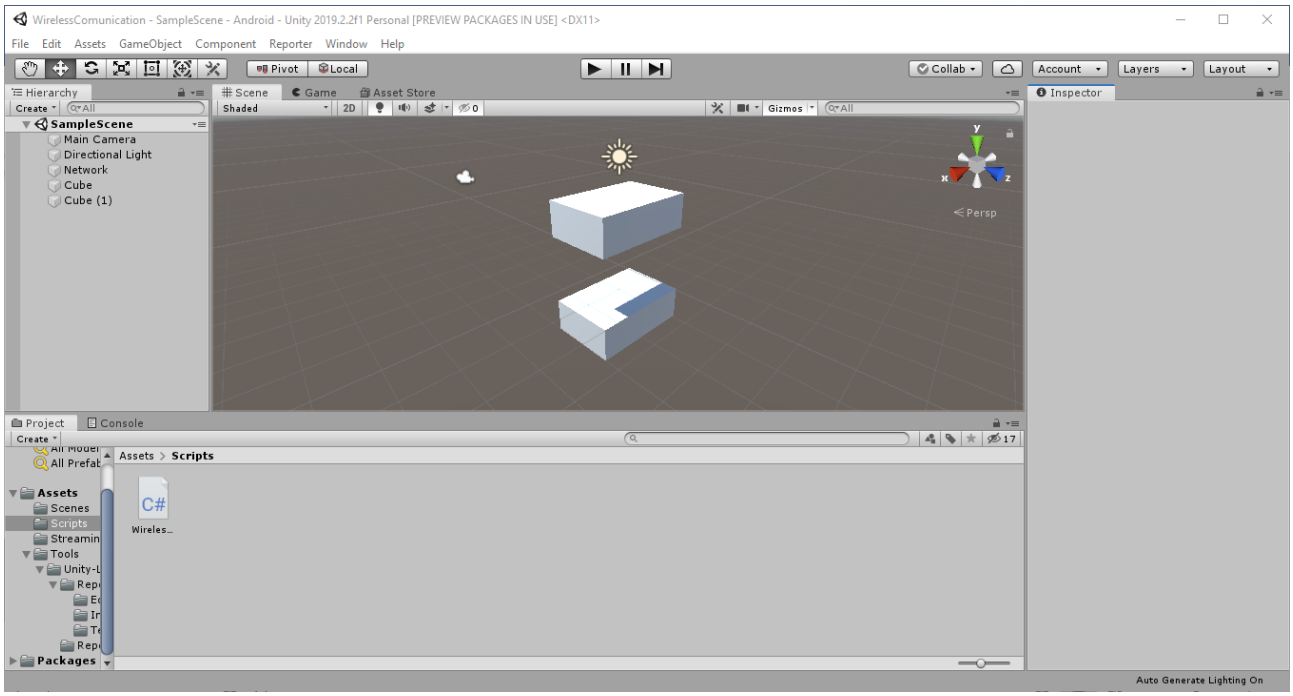


Рис. 5.3. Додані елементи в редакторі Unity3D

Після цього створено новий файл розширення «cs» та описали в ньому поведінку для отримання та передачі даних. Так як передача відбувається через Wi-Fi, то буде зручно використовувати сокети та клієнт-серверну архітектуру.

Загалом, цілісна розроблена система складається з шолому віртуальної реальності та системи для визначення положення тіла людини. Таких підсистем може бути декілька. Оскільки уся комунікація відбувається через безпровідне з'єднання, додаткових контактів не потрібно. Для коректної роботи шолому також потрібно використати штатні контролери. Безпосередньо до модулю визначення положення, за допомогою стандартних роз'ємів, підключене живлення. Живлення шолому віртуальної реальності вбудоване.

Для перевірки працездатності системи, достатньо подати живлення і запустити додаток на шоломі віртуальної реальності. Після цього система одразу почне передавати положення частини тіла, на якій закріплена. У конфігураторі можна змінити частину тіла, яку потрібно відстежувати, та одягнути систему для позиціонування на неї, якщо це потрібно. Для перевірки валідності роботи системи

було проведено декілька тестів з різними частинами тіла для перевірки поведінки системи при зміні розташування модулів позиціонування.

Модуль позиціонування - це один із ключових елементів спеціалізованої системи для визначення положення тіла людини у віртуальному світі. Він складається із двох модулів GY-521 та мікроконтролера. Для коректної його роботи цей модуль повинен правильно розміщуватись на частині тіла людини. Основна ідея полягає в тому, що будь-яку кінцівку людини можна умовно розділити на вектори та визначати їх повороти один відносно одного. Таким чином, можна отримати кінцеве положення останнього вектору відносно корпусу тіла людини. Систему модулів позиціонування можна розміщувати і на інших частинах тіла людини. Для цього потрібно змінити конфігурацію у додатку візуального відображення даних на шоломі віртуальної реальності.

Для зчитування та передачі даних з модулів позиціонування можна використовувати кватерніони або ж кути Ейлера. Існує теорія, що використання кватерніонів більш точніше, але потребує більших обчислювальних ресурсів. Це можна пояснити тим, що кватерніон це набір із чотирьох параметрів, які визначають вектор і кут повороту навколо нього, а кути Ейлера це набір лише з трьох параметрів, що визначають вектор повороту. У кватерніонах параметри одиничного вектора множаться на синус половини кута повороту. Четвертий компонент - косинус половини кута повороту. У кутах Ейлера параметри одиничного вектора і є кутами повороту навколо осей.

Використання кутів Ейлера має суттєвий недолік - складання рамок. Блокування обертання (складання рамок) – це термін, що відноситься до області гіроскопів та інерційної навігації. Для вільного гіроскопа в двовісному підвісі термін описує подію, яка може відбуватися в тому випадку, коли внутрішня рамка гіроскопа повернеться на 90 градусів щодо зовнішньої рамки, і при цьому вектор кінетичного моменту буде спрямований по осі зовнішньої рамки. При такому положенні гіроскоп втратить свою основну властивість - зберігати напрямок в інерціальному просторі, яке задається вектором кінетичного моменту. Точність

роботи системи при використанні кватерніони та кутів Ейлера зображено на рис. 5.4.

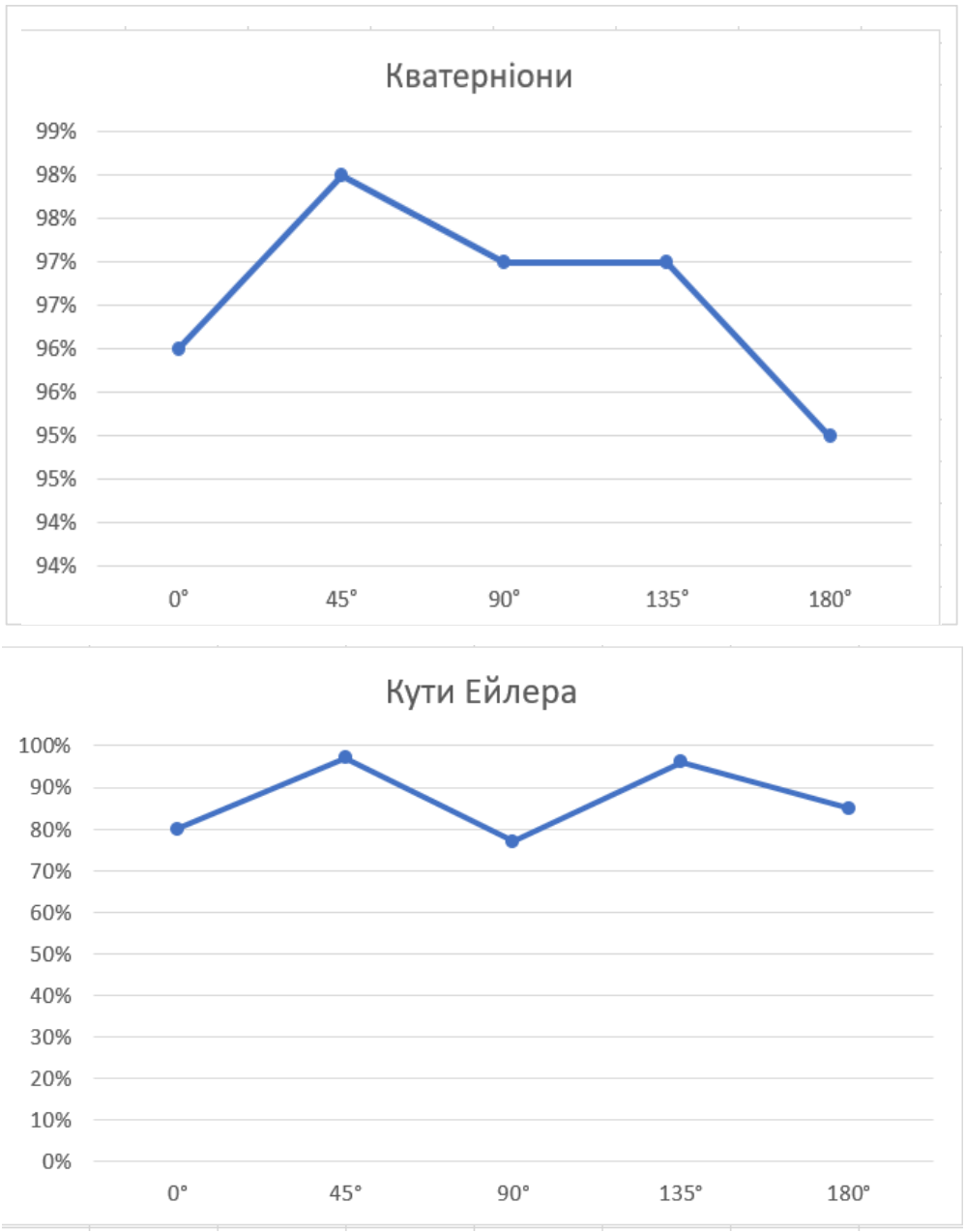


Рис. 5.4. Точність системи при використанні кватерніонів та кутів Ейлера

Як видно із наведених залежностей, використання кватерніонів більш прийнятне, адже при використанні кутів Ейлера точність системи страждає у критичних точках, коли осі гіроскопу близькі до ефекту складання рамок. Використання кватерніонів повністю нівелює цей ефект і на практиці не відбувається швидкої незапланованої зміни кутів повороту віртуальної частини тіла людини.

Щодо аналізу швидкості роботи кватерніонів та кутів Ейлера, то різниці не було виявлено. Це пов’язано із дуже малою величиною часу на одну ітерацію роботи алгоритму (приблизно 0,1 мс). Додавати час декількох ітерацій не можна через те, що час на візуалізацію роботи значно більший (приблизно 10 мс), що призведе до накопичення великої похибки. Тому часом обробки кватерніонів або ж кутів Ейлера у загальному часі обробки даних із модулів позиціонування можна знехтувати, що робить використання кватерніонів більш прийнятним.

Для перевірки системи використовувалось розміщення системи на руці. Відеоряд змінювався залежно від положення тіла людини. Під час використання кватерніонів мертвих зон виявлено не булою. При використанні кутів Ейлера були ситуації з так званим шарнірним замком, що вплинуло на сприйняття віртуальної реальності. Тобто використання кватерніонів більш перспективне у широких масах [210].

5.2. Інформаційно-вимірювальна платформа для визначення рівня радіаційного фону в режимі реального часу

При проектуванні системи було застосовано модифікований метод очищення даних в бездротовому сенсора.

Спроектовано, імплементовано та досліджено мобільну платформу для визначення рівня та локації радіаційного фону з одночасним контролем температури та атмосферного тиску потенційно небезпечних територій в реальному часі. Система здатна обробляти дані, формувати досліджений маршрут та надсилати інформацію на віддалене сховище для подальшої обробки. Запропонована структурна схема системи містить дві підсистеми з необхідним підключенням усіх необхідних компонентів. Системою можна управляти за допомогою радіоканалу зв’язку та мережі Інтернет. Вона має здатність формувати досліджений маршрут на віддаленому хмарному сервісі, що дозволяє переглядати накопичені дані за весь період роботи системи та здійснити загальну оцінку забрудненості певної території. Досліджено роботу системи у реальних умовах. Показано, що досліджувана система працює коректно, зібрані показники майже миттєво передаються на віддалене сховище в хмарному сервісі. Це все відбувається

у режимі реального часу з використанням реляційної бази даних. Також передбачена можливість локального моніторингу за системою.

В сучасному світі завдання визначення стану забрудненості навколишнього середовища є дуже актуальним, адже з розвитком техніки у вільному користуванні та доступі перебуває все більше пристроїв, які можна використовувати як у корисних так і шкідливих цілях. Найбільшу небезпеку становить підвищений радіаційний фон, адже радіацію відчути і виявити досить важко, не використовуючи спеціальних вимірювальних приладів. У військовій сфері, де існує велика ймовірність, що противник використав ядерну зброю, проблема визначення шкідливості навколишнього середовище ще актуальніша. Для уникнення шкідливого впливу на здоров'я людини, спочатку, необхідно провести розвідувальну операцію. Для виконання такого важливого, та надзвичайно небезпечного завдання необхідно використовувати автономні мобільні системи. Такі системи можуть самостійно зібрати необхідні показники навколишнього середовища та дозволять уникнути людських жертв.

Крім цього, подібні системи можуть застосовуватися у сільському господарстві, харчовій промисловості для оцінки опромінення загалом, визначення розподілу ймовірностей розташування та інтенсивності джерела опромінення, тощо. Значна увага дослідників зосереджена на розробці різноманітних мобільних систем як наземного, так і повітряного призначення для визначення дози опромінення, накопичення інформації за певний період, а також подачі тривожних сигналів, коли швидкість еквівалентної дози перевищує норми. Більшість з них орієнтовані на моніторинг радіаційного фону.

Таким чином, важливо було спроектувати систему визначення рівня та локації радіації з одночасним контролем мікрокліматичних параметрів та розмістити її на мобільній платформі. Система буде проводити виміри радіаційного фону на заданій території, обробляти дані, формувати досліджений маршрут та надсилати на віддалене сховище для подальшої обробки. Основна цифрова частина системи буде відповідати за зчитування показників рівня іонізуючого випромінювання із дозиметра, встановленого на автономній платформі, давачів температури,

атмосферного тиску та відслідковувати координати поточного місцезнаходження за допомогою GPS модуля, збирати ці дані та передавати по безпровідному каналу зв’язку GSM. Керування системою буде здійснюватися двома способами: беспосереднє управління за допомогою додатку на смартфоні або попереднє задання маршруту, який необхідно дослідити.

Структурна схема системи зображена на рис. 5.5. Вона містить основні функціональні частини (елемент, пристрій, функціональну групу, функціональну ланку) їх взаємозв'язки та призначення. Програмований мікроконтролер є основою мобільної платформи для визначення рівня та локації радіаційного фону. Проектована система складається з двох незалежних підсистем. Перша несе відповідальність за збір, обробку та передачу накопичених даних до хмарного сервісу, а друга відповідає за управління переміщенням мобільного агента за заданою територією та реалізовує систему безпосереднього управління.

Для побудови системи збору, обробки та передачі накопичених даних буде використано мікрокомп’ютер Raspberry Pi 3B. Визначення рівня радіаційного фону буде базуватись на показниках лічильника Гейгера. Ці дані будуть вимірюватись та передаватися на мікроконтролер та оброблятись за певним алгоритмом. З метою підвищення точності показників, буде здійснюватися визначення середнього значення за заданий інтервал часу. Обробивши та накопичивши ці дані, мікропрограма упакує їх разом з показниками інших сенсорів та координатами поточного місцеположення, отриманими з GPS модуля. Наступним кроком дані будуть відправлені до хмарного сервісу, використовуючи Internet та GSM GPRS модуль.

Для побудови системи управління переміщенням даних використано апаратну платформу. Керування буде здійснюватися або безпосередньо за допомогою радіопульта керування, яким може бути звичайний смартфон, або платформа може керуватись заздалегідь запрограмованим алгоритмом, рухаючись заданим маршрутом дослідження.

Розроблена система містить такі компоненти: мікрокомп’ютер Raspberry Pi, GSM модуль, GPS модуль, лічильник Гейгера, сенсор температури та

атмосферного тиску, програмований мікроконтролер ESP8266, чотири двигуни постійного струму, драйвер двигунів.

В підсистемі збору та передачі даних розташовані всі необхідні сенсори: термометр, барометр, GPS модуль для визначення точних координат місцеположення платформи та дозиметр для визначення рівня радіаційного фону. Підсистема управління переміщенням складається з набору двигунів, силової установки, необхідної для керування двигунами (драйвера двигунів) та мікроконтролера, який буде відповідати за переміщення платформи у просторі.

Окремим блоком виступає хмарний сервіс, який буде надавати зручний інтерфейс користувача для перегляду накопичених даних на карті у вигляді міток на заданому маршруті.

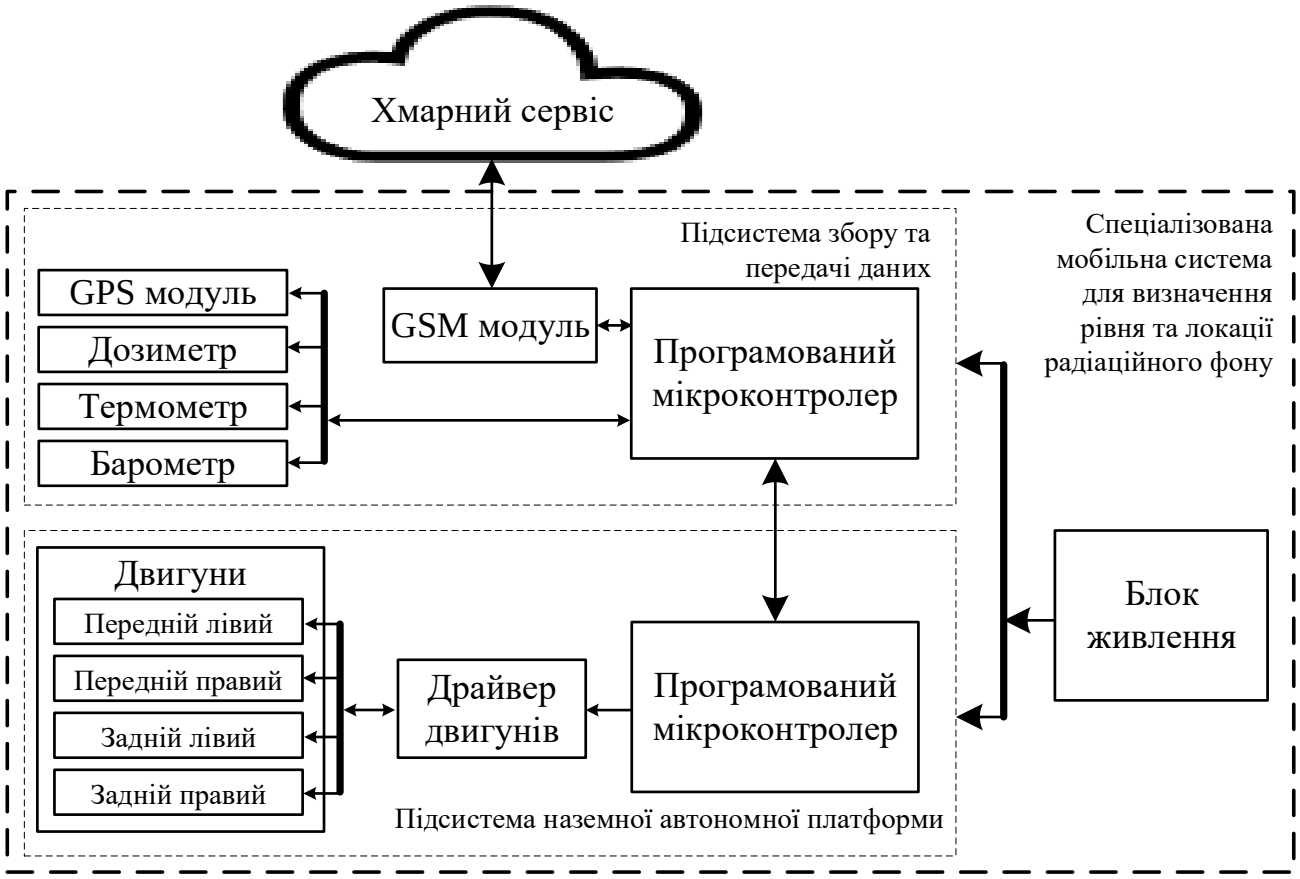


Рис. 5.5. Структурна схема проектованої системи

Для проектування підсистема наземної автономної платформи було використано Lolin NodeMCU V3, яка генерує сигнали управління та силову установку безпосереднього керування двигунами. В її основі лежить мікросхема драйвера двигунів L293D та два мотори постійного струму.

Враховуючи те, що спроектована мобільна система для визначення рівня та локації радіаційного фону складається з двох функціонально повних підсистем, то і підхід до реалізації програмного коду кожної з підсистем буде відрізнятися.

Для розробки програмного забезпечення модуля керування підсистеми, яка відповідає за збір, обробку та передачу накопичених даних до хмарного сервісу, було використано високорівневу мову програмування Python з надзвичайно великою екосистемою бібліотек з відкритим кодом.

Ця мова програмування відрізняється зрозумілістю та легкістю до сприйняття, що дозволяє доволі швидко опанувати основні можливості та навчитися перевикористовувати вже готові рішення. Для зручності написання програмного коду було використано середовище розробки PyCharm Community Edition, яке безкоштовне до завантаження з офіційного сайту.

Для розробки програмного забезпечення модуля керування підсистеми наземної автономної платформи, яка відповідає за управління переміщенням мобільного агента по заданій території та реалізовує систему безпосереднього управління, було використано спеціалізоване середовище розробки програмного забезпечення Arduino IDE для сімейства програмно-апаратних платформ Arduino.

Алгоритм роботи мобільної системи представлений на рис. 5.6. Після ініціалізації мобільний агент переходить до стану очікування запиту на вимірювання. Далі, коли отримано запит на вимірювання, разом з координатами необхідної точки, мобільна платформа рухається в напрямку необхідної точки виміру показників, орієнтуючись за допомогою вбудованого GPS модуля. Коли платформа завершила своє переміщення, починається процес збору даних зі всіх сенсорів, вимір радіаційного фону за допомогою вбудованого дозиметра та отримання точних GPS координат поточного місця розташування. На наступному етапі зібрані дані відправляють до хмарного сервісу через інтернет за допомогою GSM модуля з підтримкою мережної технології безпровідного зв’язку GPRS. Далі система знову переходить в режим очікування нового запиту на вимірювання, зменшуючи використання енергії до мінімуму. Система переходить в режим

очікування, з заданим часом затримки, та збереженням електроенергії. Такий процес буде циклічним за наявності напруги живлення.

а) б)

Рис. 5.6. Граф-схема алгоритму роботи мобільної вимірювальної платформи (а) та хмарного сервісу (б)

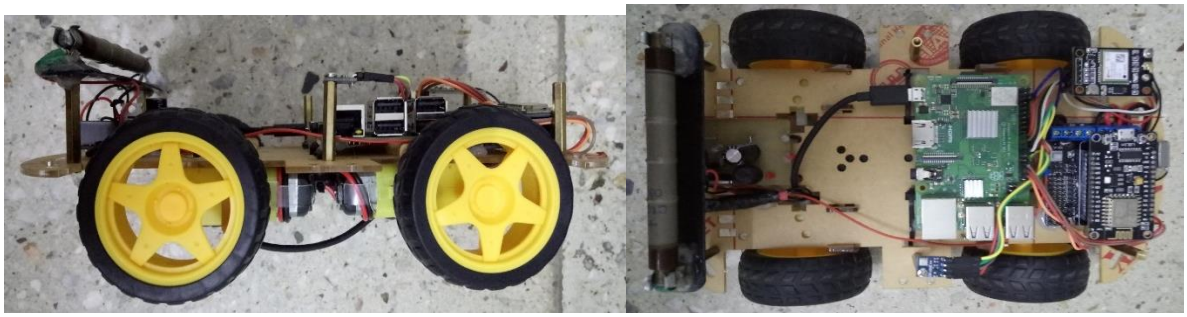
Робота хмарного сервісу. Після старту аплікації на віддаленому хмарному сервісі відбувається ініціалізація. Наступним кроком формується запит на вимірювання, до якого можуть входити GPS координати необхідної точки, де треба провести виміри показників температури, атмосферного тиску та рівня радіаційного фону. Сформоване повідомлення-запит відправляється через інтернет до автономної мобільної платформи для опрацювання. Далі хмарний сервіс переходить в режим очікування отримання даних від мобільного агента. Якщо такі дані не були отримані протягом заданого інтервалу часу, то відправляється повторний запит. Коли дані були отримані, відбувається розпаковка та збереження отриманої інформації до бази даних. Наступним кроком дані обробляються спеціальними алгоритмами для фільтрування неправдивих даних з метою зменшення статистичної похибки фільтруючи інформацію на наявність шуму. Далі формується точка на карті забруднення, в якій зазначено виміряні та опрацьовані відомості про виміряну температуру навколишнього середовища, атмосферний тиск та рівень радіаційного фону. Якщо оператор приймає рішення завершити вимірювання, то система переходить в режим енергозбереження. В іншому випадку процес повторюється циклічно, до того моменту, доки можливо сформувати запит на вимірювання, та подається електроенергія.

Для підключення GPS модуля та отримання координат поточного місця розташування спроектованої мобільної платформи було додатково завантажено та встановлено Python бібліотеку pynmea2, яка дозволяє розшифрувати NMEA 0183 протокол – спеціальний протокол компонування геопозиційних даних. Для підключеня GSM модуля та отримання доступу до мережі інтернет за допомогою GPRS технології Python бібліотека SIM800Modem була використана. Підключення сенсорів температури та атмосферного тиску здійснювалося за допомогою Python бібліотеки bmp280, яка надає зручний інтерфейс для отримання. Для визначення рівня радіаційного фону необхідно підрахувати кількість імпульсів за певний період часу. Тому лічильник Гейгера було підключено до одного з GPIO портів.

Для керування переміщенням платформи використовується бібліотека з відкритим кодом remoteXY, яку було додано до програмного коду модуля

NodeMCU Lolin та кросплатформного додатку RemoteXY, який встановлений на смартфон.

Зовнішній вигляд мобільної платформи зображено на рис. 5.7.



(а) (б)

Рис. 5.7. Зібрана мобільна платформа, вид збоку (а) , вид зверху (б).

Для перевірки працездатності спроектованого пристрою достатньо подати живлення і система одразу після завантаження та ініціалізації переходить до стану очікування запиту на вимірювання. Далі, коли отримано запит на вимірювання, мобільна платформа починає свій рух в напрямку необхідної точки виміру показників, орієнтуючись за допомогою вбудованого GPS модуля. Коли платформа перемістилася, починається процес збору даних зі всіх сенсорів, вимір радіаційного фону за допомогою вбудованого дозиметра, та отримання точних GPS координат поточного місця розташування.

Для керування автономною платформою може бути використаний пульт керування – смартфон, за допомогою сервісу RemoteX. Цей сервіс дозволяє швидко сформувати мобільний додаток, який може підключитися безпосередньо до платформи за допомогою безпровідного мережевого інтерфейсу Wi-Fi або за допомогою Internet. На рис. 5.8. зображено необхідний мінімум об’єктів керування автономною платформою.

В цьому дослідженні для перевірки роботи рухомої платформи, було сформовано необхідний мобільний додаток, який підключався до платформи за допомогою безпровідного мережевого інтерфейсу Wi-Fi. Керування автономною платформою було здійснено за допомогою джойстика або G-сенсора, вбудованого

в смартфон. Під час тесту переміщення платформи було перевірено коректність реакції платформи на керуючі команди.

Рис. 5.9. Пульт безпосереднього управління

Дослідження розробленої платформи для оцінки рівня та локації радіаційного фону проводилося на певній території. Загалом було здійснено дванадцять вимірів. Кожен вимір проводився протягом 30 секунд, що дозволило зменшити рівень похибки до мінімуму. Після завершення серії вимірів, система сформувала таблицю зібраних даних в реляційній базі даних (таблиця 5.1).

Таблиця 5.1

База даних виміряних показників

п/п	GPS координати	Рівень радіації, мкЗв/год	Температура, °C	Атм. тиск, мм рт. ст.
1	49.818090, 24.012752	12	6,2	760
2	49.818254, 24.012956	11	6,1	762
3	49.818509, 24.013300	12	6,2	762
4	49.818398, 24.013463	12	6,2	758
5	49.818242, 24.013450	11	6,3	761
6	49.818048, 24.013434	11	6,2	763
7	49.817888, 24.013420	11	6,1	757
8	49.817715, 24.013410	11	6,0	760
9	49.817527, 24.013399	12	6,0	761
10	49.817531, 24.013058	11	5,9	759
11	49.817543, 24.012715	11	6,0	764
12	49.817819, 24.012731	12	5,9	761

5.3. Апаратно-програмний комплекс вимірювальної системи типу орнітоптер

При проектуванні системи було застосовано модифікований метод істиності моніторингу даних при розподілі ресурсів на основі туманих обчислень для системи з мультисенорсною конфігурацією.

Представлено рішення щодо створення мініатюрного безпілотного аеро-наземного орнітоптера військового призначення враховуючи досвід провідних зарубіжних та вітчизняних компаній, а також науково-дослідних лабораторій. Апаратний комплекс побудований як малогабаритний орнітоптер з використанням найсучасніших електронних компонентів, таких як чіп Ardupilot Mega 2.6 і GSM-модуль для забезпечення автономності та можливості орієнтування в просторі. Збільшуючи вантажопідйомність пристрою, можна встановлювати додаткові модулі, зокрема фото- чи відеокамеру та модуль OSD, який дозволить накладати налаштування телеметрії на зображення, що передається з камери. Контрольоване стабільне зависання мінімізує негативний вплив вібрацій, що виникають під час руху крил моделі, і збереже час автономної роботи.

Зазвичай реальні розробки БПЛА включають мікроконтролер, набір сенсорів для визначення необхідних параметрів і камеру спостереження для передачі відеоінформації на приймач в режимі реального часу. Найбільш широко використовуються гвинтові транспортні засоби та транспортні засоби з нерухомим крилом. Однак ці пристрої мають ряд недоліків і коштують дорого. Поряд з ними поширені інші види пристроїв з нестандартними підходами до конструкції і принципу роботи. Такими пристроями є БПЛА на кшталт орнітоптерів, які використовують помахи крилами для створення підйомної та тягової сили. Малі орнітоптери, які використовують крила-крила для створення аеродинамічних сил, мають низку переваг перед нерухомими та пропелерними крилами. Вони безпечні в експлуатації, оскільки не мають обертових частин і паливних баків. Передбачається, що імітуючи спритність і спритність живих літаючих істот, орнітоптери можуть бути багатоцільовими БПЛА.

У спроектованому БПЛА всі елементи системи підключені до чіпа Ardupilot Mega 2.6. Обрана платформа позиціонується як польотний контролер, який включає як звичайний мікроконтролер, так і повноцінний автопілот. Ardupilot базується на ARM 2.x (DIY Drones), проект містить відкритий код. Плата керування складається як з обчислювальних засобів, так і з можливостей прийняття рішень. Перевагою борту є наявність функції автопілота, що дозволяє машині самостійно рухатися по заданих траєкторіях без зовнішнього втручання. Для підключення автопілота APM можна використовувати різні конфігурації. У даній роботі використовується підключення літакового типу, яке виділяє канали для управління силовою установкою (Turnigy 2615 EDF Outrunner) і сервоприводами (Hitec HS-65MG), які відповідають за поверхні управління (ліфти, слони). Використовувані конфігурації з використанням моделі руля висоти, елеронів і цапф і лише рулів висоти.

У розробленій моделі орнітоптера основною керуючою поверхнею є хвіст, який може контролювати рух по тангажу вгору і вниз і змішане управління - комбінація рухів хвоста з боку в бік, уздовж осі крену, і вгору і вниз для дислокації (реалізований гібридний слон). Для керування напрямком моделі використовуються дві моделі серводвигунів, одна з яких (для керування слоном) згинається разом із хвостом під час руху вгору та вниз. Сервопривід по тангажу закріплений на фюзеляжі, а для борту по крену - на рамі, яка повертається разом з хвостовим оперенням.

Електродвигун через двоступеневий редуктор зменшує рух крил. Підключення конфігурації організовано таким чином, що входи плат автопілота підключаються до чотирьох портів приймача радіосигналу. Один з них використовується для управління обертами двигуна, другий - для вибору режиму польоту, а ще два - для управління сервоприводами. Структурна схема з’єднань наведені на рис. 5.10.

Виходи автопілота підключаються до сервоприводів і до двигуна (через схему ESC - регулятор швидкості). Живлення подається через контролер ревізії одночасно на всі елементи системи. Також до відповідного порту підключається

GPS-модуль. Сам модуль автопілота встановлений на фюзеляжі. Для розпізнавання напрямків руху автопілота і правильного позиціонування він встановлений на спеціальній підставці, щоб зменшити негативний вплив вібрацій на гіроскопи та акселерометри.

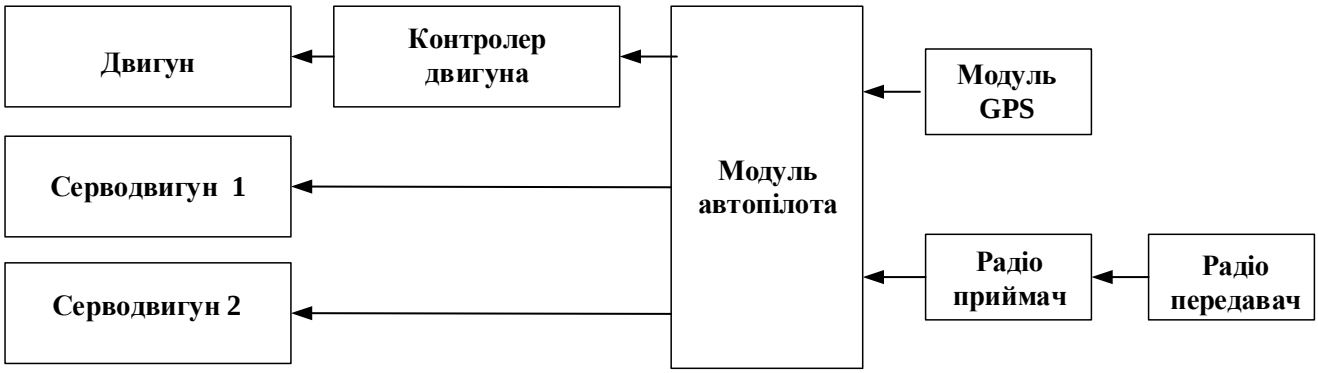


Рис. 5.10. Структурна схема з’єднання основних елементів проектованого БПЛА типу орнітоптер

При такій конфігурації один з важелів пульта управління буде відповідати за управління обертами двигуна і, відповідно, швидкістю моделі, а другий - за управління сервоприводом - тобто напрямком руху автомобіля. пристрій. За перемикання режимів польоту буде відповідати один з доступних перемикачів. Робота розробленого орнітоптера може бути організована в 12 режимах роботи, доступних для цього програмного забезпечення, як для ручного, так і для автономного керування пристроєм.

Конфігурація та програмування борту реалізовано за допомогою Mission Planner, який також дозволяє змінювати дисплей сенсора, будувати план польоту, симулювати політ для навичок ручного управління, перегляд так званих «логів» (окрема мікросхема пам'яті в платі). для «реєстрації» польотних даних, так звана «чорна скринька») тощо.

Перемикання режимів здійснювалося перемикачами на пульті (передавачах). Певне положення перемикача переводиться в певний режим. Також можливе перемикання режимів через наземну станцію управління - комп'ютер, для цього

обов'язкова установка радіомодемного модуля, наявність якого дозволить не тільки перемикати режими, а й отримувати телеметричні дані зображені на рис. 5.11.

Рис. 5.11. Отримання телеметричних даних програмою Mission Planner, фіксованих проектованим орнітоптером

Таким чином, розроблений апаратно-програмний комплекс – малогабаритний БПЛА типу орнітоптер може бути використаний для моніторингу території як бойових дій, так і районів цивільного значення. Збільшуючи потужність пристрою, можна встановити додаткові модулі, такі як фото- або відеокамера і модуль OSD, що дозволить застосувати параметри телеметрії для зображення, які передається з камери.

Контрольоване стабільне зависання мінімізує негативний вплив вібрацій, що виникають під час роботи крил моделі, і збереже час роботи акумулятора. Зміна та оптимізація режимів автономного польоту може покращити керованість і розширити функціональність орнітоптера [211].

5.4. Система для визначення напрямку до джерела звуку

При проектуванні системи було застосовано метод динамічного пошуку помилок в інформаційно-вимрювальній системі.

В якості ядра обрано 32-розрядний мікроконтролер STM32, який виконує всі дії, необхідні для реалізації алгоритмів обробки звуку в проектованому пристрої. Електричні коливання на виходах трьох мікрофонів використовувалися як вхідні сигнали для аналогових електричних коливань. Отримані сигнали підсилюються мікрофонними підсилювачами, які мають програмно-контрольоване співвідношення, яке відповідає амплітуді АЦП номінальному рівню. Підсилювач звуку мікрофона був запрограмований на контрольований рівень посилення за допомогою програмного забезпечення, що дозволило мікрофону реагувати на рівень сигналу навколишнього шуму та підтримувати номінальний вхідний сигнал на вході АЦП. Ця стратегія дозволила знизити як рівень шуму (при низькому рівні вхідного сигналу), так і рівень спотворення сигналу (при сильному сигналі).

Для створення МП було використано кілька схожих концептуальних проектів для існуючих прототипів. В якості центрального мікропроцесора обрано найбільш значущий 32-розрядний мікропроцесор (МК) STM32, який виконує всі етапи реалізації алгоритмів обробки звуку в проектованому пристрої. Електричні коливання на виходах трьох мікрофонів, зображених на рис. 5.12 мають аналоговий характер.

Отримані сигнали підсилюються мікрофонними підсилювачами, які можна налаштувати за допомогою програмного забезпечення, щоб мати співвідношення до номінального рівня АЦП. Використання такого мікрофонного підсилювача з підсиленим програмним посиленням дозволить реагувати на гучність звуку медіа-модуля та підтримувати номінальний рівень вхідного сигналу на вході АЦП. Такий підхід дозволяє зменшити рівень шуму (при низькому вхідному рівні) і підвищити точність сигналу (при високій потужності сигналу). Пристрій має три дистанційних мікрофона, які в кожному каналі складаються з двох секцій. Перша частина імплементована біля мікрофона і перетворює вихідний сигнал у диференціальний сигнал з низьким внутрішнім опором. Другий компонент мікрофонного підсилювача є внутрішнім для пристрою, він перетворює диференціальні вхідні сигнали в стандартний формат, усуває синфазні

перешкоди, які можуть бути викликані лінією зв'язку. Підсилювачі другої частини мікрофонного підсилювача керуються програмою, що дозволяє програмно змінювати номінальні рівні, необхідні для входів АЦП. Отримані результати відображаються на панелі дисплея, а потім передаються для подальшого аналізу.

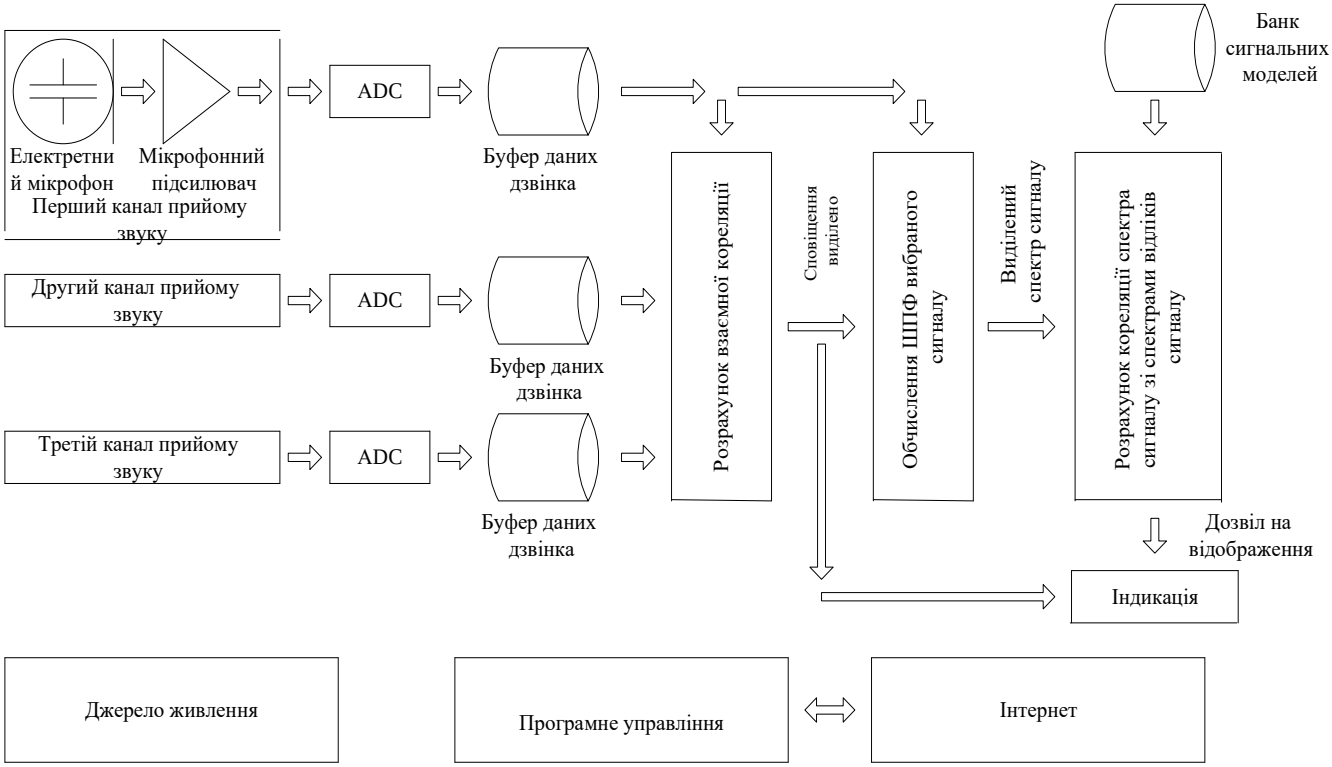


Рис. 5.12. Функціональна схема пристрою для визначення напрямку джерела звуку

Щоб отримати вхід АЦП номінального рівня, кожен потік генеруватиме команди для системи контролю посилення амплітуди (AGP). Це дозволить використовувати показання АЦП і генерувати керуючі сигнали для систем АЦП, які з часом змінюють передачі мікрофонних підсилювачів. На виходи АЦП надходять цифрові потоки даних, які потім передаються в кільцевий буфер для накопичення (це необхідно для реалізації алгоритму крос-кореляції). Кільцеві буфери містять дані, які зберігаються в них таким чином, що кожна нова частина даних переміщує старі дані з буфера.

На першому кроці алгоритм кореляції має здатність знаходити відповідні дані для кожного з каналів прийому із затримкою. Якщо спостерігаються фрагменти даних цього типу, обчислюєтьсявзаємна затримка звукового вводу в

кожному з кільцевих буферів. Після аналізу інформації алгоритм може чітко визначити напрямок, звідки походить звук.

На другому кроці вибрані сигнали перетворюються за допомогою алгоритму ШПФ у спектральне представлення, яке можна інтерпретувати як масив даних, який можна використовувати для розпізнавання типу звуку.

На третьому етапі визначається тип джерела звуку. Для цього спектр спектральних даних прийнятого сигналу зіставляється з основними типами сигналів. Порівняння ґрунтується на алгоритмі зворотної кореляції. Типом сигналу вважали вибірку, яка мала найбільшу кількість відповідних характеристик. Окрім визначення типів сигналів, можна використовувати програмне забезпечення, яке має здатність запам’ятовувати параметри невідомих сигналів у базі даних. Виявлені сигнали відображаються на панелі дисплея (яка складається з триколірних світлодіодів), тут відображається спрямованість джерела звуку, а також характеристики отриманого звуку, це виділяє постріли від снайпера червоним кольором.

Програма мікроконтролера має окрему процедуру, яка керує його функціональністю та полегшує передачу даних через усі блоки та відображення отриманої інформації. Крім того, дані передаються на зовнішній сервер для додаткового аналізу. Внутрішня температура регулюється вбудованим сенсором. Вихід сенсора підключається до АЦП через мультиплексор. Структура мікроконтролера включає інтегровані комірки, такі як Trace Macrocell, що має значний вплив на функціональність налагодження, це дозволяє спостерігати за потоком інструкцій і даних через середину ядра ЦП у режимі реального часу [212].

5.5. Система управління розумним будинком

При проектуванні системи було застосовано модифікований метод істиності моніторингу даних при розподілі ресурсів на основі туманих обчислень для системи з мультисенорсною конфігурацією.

Підсистема безпеки та система управління розумним будинком реалізовані з використанням в якості сенсорів температури та вологості конструкцій на основі кераміки. Виявлено потенційно вразливі місця та розроблено шляхи їх мінімізації.

Запропоновано схему шифрування інформації на основі набору інфраструктури відкритих ключів. Він практично невразливий з точки зору надійності сервера і його безперебійної роботи. Проведено тестування підсистеми безпеки.

Система управління освітленням здійснюється автоматично. Вона визначає місцезнаходження людини і вмикає освітлення тільки там, де це необхідно. Також є можливість регулювати яскравість. Встановлення та зняття квартири з охорони здійснюється за допомогою кодової панелі, яка встановлюється в тамбурі. При відкритті вхідних дверей у людини є 30-35 с, щоб правильно ввести код безпеки. Якщо код не введено, розумний дім увімкне сирену та надішле повідомлення на вказані номери телефонів. Також у кожній кімнаті квартири та кухні встановлені сенсори руху, які допоможуть виявити проникнення ззовні через вікно.

Система клімат-контролю працює на основі своїх алгоритмів, які дозволяють підтримувати параметри температури в приміщенні при різних кліматичних зовнішніх умовах з мінімальними енерговитратами.

Для налаштування параметрів системи використовуються різні сенсори, які вимірюють поточний мікроклімат в будівлі, а також елементи управління вимикачами і панелями. Слід зазначити, що в розробленій системі використовувалися розроблені наноструктурні сенсори температури та вологості. Система здатна контролювати якість повітря (температуру, вологість) за часом доби та пори року, режим провітрювання за допомогою системи автоматичного відкриття вікон, змінювати режим роботи радіаторів і теплої підлоги, автоматично підтримувати температури і вологості в спеціальних приміщеннях, а також випадкової зупинки системи опалення. Типова структура і схема передачі даних у системі «розумний дім» наведена на рис. 5.13. Вона містить базу сертифікати (частина фізичного сервера, яка зберігає всі цифрові підписи, до якої має повний доступ логічний сервер і частковий доступ користувача з мобільного). Мікроконтролер (пристрій, який безпосередньо відповідає за керування розумним будинком), сервер, мобільний пристрій. Інформація зберігається в хмарі.

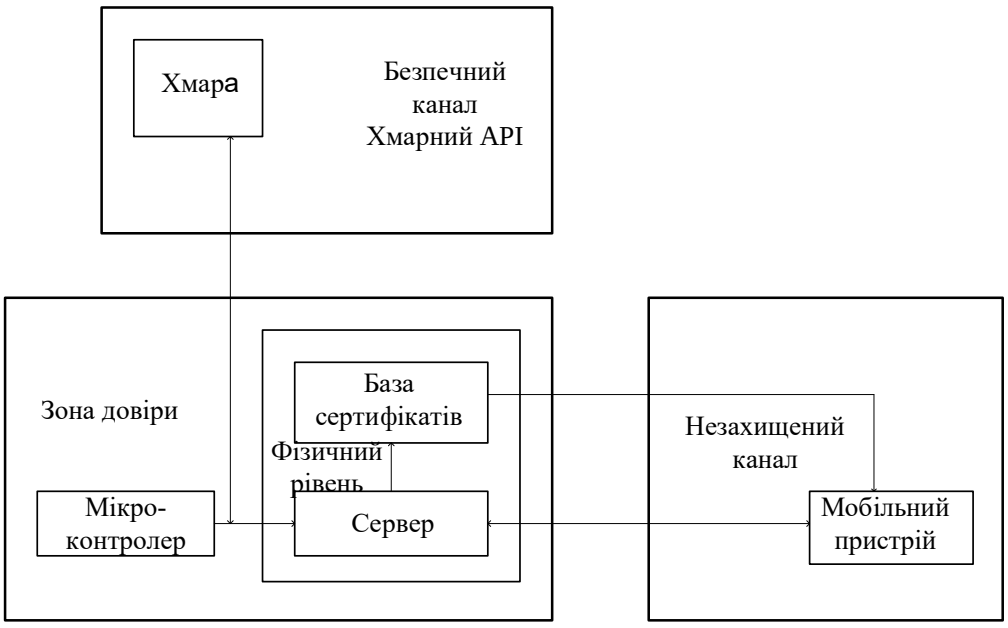


Рис. 5.13. Схема передачі даних в системі розумний будинок

В рамках локальної мережі (мереж сервера) інформація захищена. Незахищений канал користувача становить небезпеку. Один із традиційних сценаріїв - man-in-the-middle, тобто можливість інших індивідів підключатися в канал між сервером і користувачем і підлаштовуватися під когось із цих ключових осіб, беручи на себе роль невидимого посередника. Щоб закрити джерело інформації, слід використовувати схеми шифрування інформації. Як правило, для захисту систем розумного будинку можуть використовуватися криптоалгоритми DES і AES. У процесі шифрування використовується інфраструктура відкритих ключів. Шифрування з відкритим ключем (асиметричне шифрування) використовує пару ключів для шифрування та декодування супроводу. Пара ключів складається з одного відкритого та одного закритого ключів, математично пов'язаних між собою. Відкритий ключ можна поширювати між користувачами та використовувати для шифрування повідомлення. Відкриті ключі використовуються для декодування повідомлення.

Довірена сторона, названа центром сертифікації (СС), надає належний відкритий ключ усім зацікавленим сторонам. У цьому випадку СС видає цифровий сертифікат, який містить відкритий ключ. СС самостійно кодує цей сертифікат відкритим ключем, який відповідає відкритому ключу в сертифікаті. Після

перевірки особистості відправника та одержувача звітів, СС розраховує підтримку хешу, який створить його сертифікат, підписує хеш закритим ключем, належним відкритим ключем в опублікованому сертифікаті, створює новий сертифікат, пов'язуючи підтримку сертифіката з підписаним хешем і відкритий новий сертифікат.

Одержувач отримує сертифікат, розшифровує підписану місиву відкритим ключем центру сертифікації, обчислює нове хеш-зміст сертифіката і порівнює два хеші. Якщо хеші збігаються, перевіряється підпис. Взаємне використання відкритих ключів дозволяє шифрувати та декодувати звіт

Загалом, процес підписання сертифіката дозволяє перевірити, чи відкритий ключ справжній, чи пошкоджений під час транспортування. Перед видачею сертифіката, забезпечення хешування СС, підписує (кодує) хеш власним секретним ключем і включає в себе в шифр хеш у даному сертифікаті. Одержувач перевіряє підтримку сертифіката шляхом декодування хешу з відкритим ключем СС, виконання окремого хеш-обслуговування сертифікатів і порівняння двох хешів. Якщо вони збігаються, сертифікат і відкритий ключ не були змінені.

У результаті система, яка використовує інфраструктуру відкритих ключів, може полегшити безпечну передачу сеансового ключа між сервером і клієнтом. Сертифікатом вибирається критично важлива інформація, яка відома лише користувачу та базі сертифікатів (компонентному серверу). База даних сертифікатів пов'язує ключ із відкритим ключем користувача. Після цього система передачі даних матиме таку структуру (авторизація та прямий зв'язок між вузлами): авторизація, «сеанс зв'язку [213].

5.6. Система SCADA для сайту BTS з використанням Raspberry Pi і Arduino IoT Cloud

При проектуванні системи було застосовано модифікований метод істиності моніторингу даних при розподілі ресурсів на основі туманих обчислень для системи з мультисенорсною конфігурацією, також метод динамічного пошуку помилок.

Система диспетчерського контролю та збору даних (SCADA) - це система, основною метою якої є керування та моніторинг пристроїв у польових умовах, здебільшого розташованих у дуже віддалених та важкодоступних місцях. Розгортання системи SCADA на будь-якому виробництві також забезпечує повну автоматизацію виробничого процесу. Різні умови та параметри системи можна точно вимірювати, контролювати та оптимально керувати в режимі реального часу. Ефективність виробництва значно покращується завдяки характеру збору даних їх обробки та керування виробничими процесами в реальному часі за необхідності.

SCADA передбачає об’єднання апаратних компонентів, таких як сенсори та виконавчі механізми, а також програмне забезпечення, наприклад інтерфейс людини і машини (HMI), для виконання своїх чотирьох основних функцій: збору даних, передачі даних у мережі, обробки даних, а також моніторингу і контролю [214].

Спрощена версія досліджуваної гібридної системи постійного струму для моніторингу зображена на рис. 5.14. Параметри, які представляють інтерес для моніторингу, виділені, і результати контролю відображаються на сайті. Підключені компоненти мають локальні контролери, які можуть дистанційно отримувати керуючий сигнал і здійснювати керуючу дію через виконавчий механізм. Система може здійснює диспетчерське керування енергоменеджменту.

Система передбачає схему диспетчерського контролю та збору даних із відкритим вихідним кодом із використанням оновленої хмарної платформи Arduino IoT для моніторингу та керування прототипом нашої гібридної системи. Сенсори струму, напруги, температури та вологості підключені до мікроконтролера Raspberry Pi для зчитування напруги, струму, температури та вологості та запуску відповідного світлодіода, коли напруга падає нижче попередньо встановленого значення, а температура перевищує задане значення. Ці дані можна відстежувати та контролювати з хмарних інформаційних панелях Arduino та за допомогою мобільних додатків.

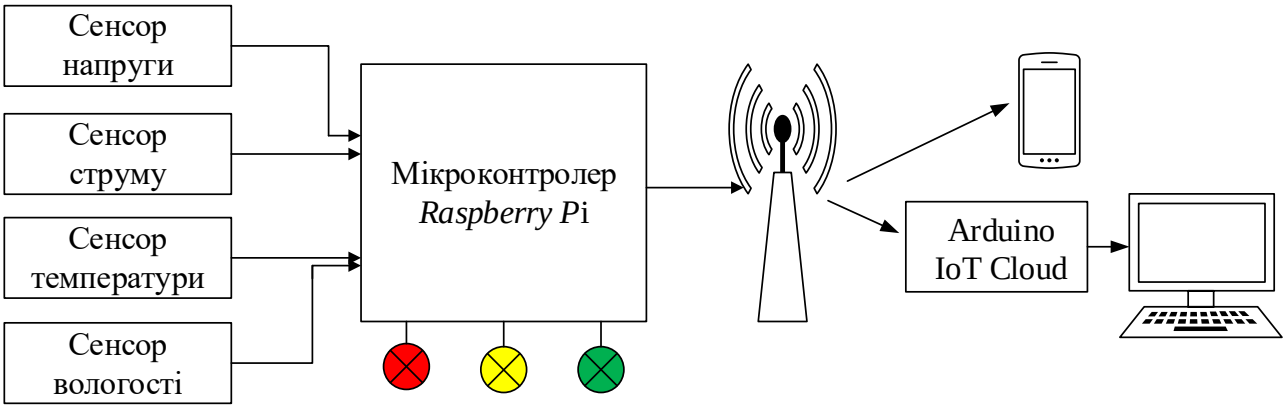


Рис. 5.14. Схема пропонованої системи IoT SCADA

Дана концепція розроблена та реалізована схемою на макетній платі з трьома сенсорами; сенсор напруги, струму, температури та вологості підключено до мікроконтролера (Raspberry Pi) для зчитування значень сенсорів. Жовтий світлодіод використовується як навантаження, для якого вимірюється струм, відповідно зелений і червоний світлодіоди реалізує логіку зниження напруги та логіку керування перегріванням. Сумісність мікроконтролера з Wi-Fi використовується для створення з'єднання з Arduino IoT Cloud. Інформаційні панелі створюються в хмарі для моніторингу та контролю різноманітних змінних. Мобільний додаток також можна використовувати для контролю та моніторингу змінних з будь-якої точки світу. Підключеними сенсорами є FID, а мікроконтролером є RTU. Arduino IoT Cloud представляє MTU, тоді як маршрутизатор Wi-Fi вдома створює канал зв'язку. Три сенсори: сенсор температури та вологості DHT11, електронний сенсор напруги MH і сенсор струму на ефекті Холла ACS 712.

Можливість Wi-Fi мікроконтролера використовується для підключення до Arduino IoT Cloud. Для написання ескізів використовується інтегроване середовище розробки Arduino (IDE) у хмарі. Виміряні дані від сенсорів збираються платою, відображаються на хмарному послідовному моніторі Arduino та надсилаються на інформаційну панель для моніторингу.

Маршрутизатор Actiontec R3000 FibreOP використовується для створення каналу зв'язку між Raspberry Pi і Arduino IoT Cloud MTU). Облікові дані маршрутизатора (SSID і пароль) забезпечують необхідний захист від

несанкціонованого доступу до системи. Хмарна платформа Arduino IoT інтерфейс користувача дозволяє контролювати дані з будь-якого місця та здійснювати контроль за потреби. Створення хмарного облікового запису Arduino IoT і хмарного плану: як і на будь-якій іншій платформі, першим кроком є реєстрація за допомогою функціональної електронної адреси та вибір плану.

Пропонований прототип системи для демонстрації моніторингу та керування малою гібридною енергетичною системою розроблений на макетній платі (рис. 5.15). Система складається з адаптера змінного/постійного струму 9 В або (батареї), перетворювача постійного/постійного струму, сенсора струму, сенсора напруги, сенсорів температури та вологості, світлодіодів (червоного, зеленого та жовтого), мікроконтролера Raspberry Pi та деяких висувних резисторів. Перетворювач DC/DC напругу 9В знижує приблизно до 5,5 В. Жовтий світлодіод відображає навантаження. Сенсор струму підключено до аналогового контакту 34 Raspberry Pi послідовно з жовтим світлодіодом для вимірювання струму, що протікає через світлодіод. Сенсор напруги підключається до аналогового контакту 32 Raspberry Pi на виході перетворювача постійного/постійного струму для вимірювання його вихідної напруги (вхідної напруги в систему). Сенсор температури та вологості підключений до аналогового контакту 33 мікроконтролера Raspberry Pi для зчитування температури та вологості навколишнього середовища. Червоний і зелений світлодіоди з відповідними операторами "If statements" у коді використовуються для реалізації логіки керування. Вони підключені до аналогових контактів 18 і 19 мікроконтролера Raspberry Pi відповідно. Коли температура перевищує 23°C, загоряється червоний світлодіод. Якщо виміряна напруга менше 4,5В, загоряється зелений світлодіод. Ця логіка управління є синонімом того, що можна отримати в польових умовах, де кондиціонер повітря або витяжний вентилятор регулює температуру в укритті або кабіні місця BTS, а зелений світлодіод відображає пускове реле дизель-генератора, яке вмикається, коли напруга падає нижче певного порогу (~ 46 В). Raspberry Pi можна підключити або через USB-порт, або за допомогою бездротового зв’язку (OTA) для бездротового завантаження ескізів із хмари на дошку.

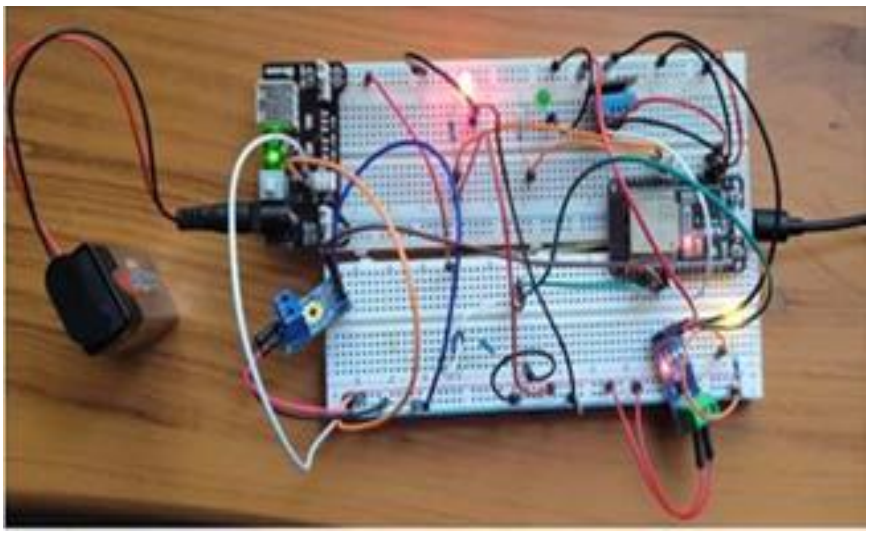


Рис. 5.15. Налаштування експериментальної схеми для прототипу системи IoT SCADA

Струм, напруга, температура та вологість реєструвалися кожні 2 секунди протягом певного періоду за допомогою діаграм. Миттєва температура і вологість також контролювалися за допомогою манометра. Сенсори дуже чутливі, щоб побачити миттєві значення, що вимірюються, порівняно з відстеженням на діаграмі. Вимірюється струм, що протікає через світлодіод навантаження (жовтий), одиниця вимірювання в міліамперах. Виміряна напруга є вихідною напругою перетворювача DC/DC у вольтах. Температура та вологість відповідають умовам навколишнього середовища. Віддалений мобільний додаток Arduino IoT також встановлено на мобільному телефоні, де віддалений моніторинг і керування також можна здійснювати з будь-якої точки за умови підключення до Інтернету та доступу до даних для входу в обліковий запис Arduino IoT (рис. 5.16).

Виміряні дані зберігаються залежно від обраного плану Arduino. Щоб продемонструвати можливість керування системою, температура навколишнього середовища підвищується до 24°C. Коли реєструється вища температура, що перевищує попередньо встановлене значення 23°C, загоряється червоний світлодіод. Світлодіод гасне, коли температура опускається нижче попередньо встановленого значення. Контроль напруги можна здійснювати так само, як і

температури. Коли напруга падає до 4,5 В, зелений світлодіод вмикається і горить, доки напруга не перевищить попередньо визначене значення.

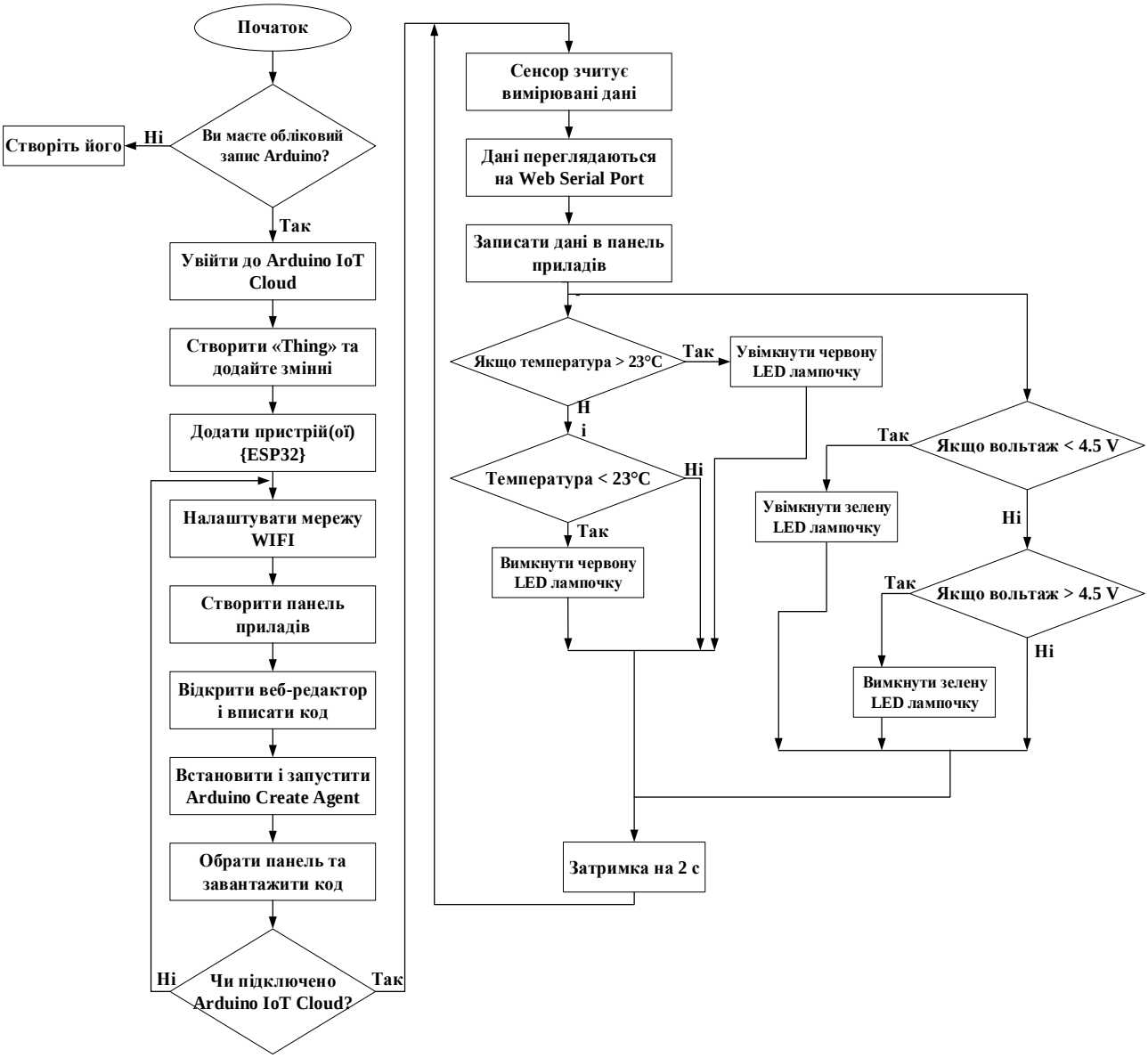


Рис. 5.16. Блок-схема системного рішення IoT SCADA

Система базується на конфігурації Інтернету речей (IoT). Прототип діє як завод/процес, тоді як сенсори є пристроями польових приладів (FID), мікроконтролер *Raspberry P*i є віддаленим терміналом (RTU), Arduino IoT Cloud діє як головний термінал (MTU) для створення необхідної людини-машини. Взаємодія (HMI), а каналом зв'язку є Wi-Fi. Усі компоненти, які використовуються для виконання цього проекту, є недорогими, легкодоступними та з відкритим кодом. У польових умовах для забезпечення необхідного підключення до Інтернету можна використовувати мікроконтролер із опцією для SIM-карт.

Інформаційна панель забезпечує інтерфейс для моніторингу виміряних даних як на ПК, так і на мобільних пристроях. Наглядовий контроль також можна здійснювати як на інформаційній панелі ПК, так і за допомогою мобільного пристрою з будь-якого місця, де є підключення до Інтернету та дані для входу. Це забезпечує певний захист від неавторизованого персоналу.

Висновки до розділу 5

Спроектовано систему визначення положення тіла людини у віртуальному світі. Вона складається з двох незалежних підсистем: збору, обробки та передачі даних про положення тіла людини та візуалізації даних у віртуальному світі. Перша підсистема побудована на базі мікроконтролер ESP8266. Визначення положення частин тіла людини базується на показниках гіроскопа та акселерометра, що знаходяться на модулі GY-521. Було зібрано і відтестовано систему для визначення положення тіла людини, яку можна використовувати із сторонніми шоломами віртуальної реальності. Дослідження системи також показали, що більше ефективно вона буде працювати при використанні кватерніонів на противагу до кутів Ейлера. Перевагою системи, окрім простоти використання, є той факт, що дані в системі можуть оновлюватися з частотою близько 200 Гц, що більш ніж достатньо для комфортного використання. Всі зібрані дані про положення тіла майже миттєво передаються на шолом віртуальної реальності і там візуалізуються.

Спроектовано та розроблено спеціалізовану мобільну систему для визначення рівня та локації радіаційного фону. Працездатність системи перевірена на макете дослідницького мобільного агента на базі колісної платформи. Вимірювальними пристроями слугують виготовлений дозиметр на базі мікроконтролера та газорозрядної трубки (лічильника Гейгера), термометр та барометр. Для визначення точних геологічних координат місця було використано GPS модуль. Для побудови системи безпровідного управління автономною системою та передачі зібраних даних до хмарного сервісу було задіяно модуль безпровідної мережевої технології GSM GPRS. Простота використаних компонентів забезпечує надійність, легкість реалізації та використання. Розроблено алгоритм роботи системи та керуючої програми. Було проаналізовано способи вимірів показників радіаційного фону на відкритій місцевості.

Розроблено апаратно-програмний комплекс – малогабаритний безпілотний літальний апарат типу орнітоптер, який може бути використаний для моніторингу території як бойових дій, так і районів цивільного значення. На даному пристрою можна встановити додаткові модулі, такі як фото- або відеокамера і модуль OSD, що дозволить застосувати параметри телеметрії до зображення, які передається з камери.

Створено мікропроцесорний пристрій для визначення напрямку джерела звуку за допомогою сучасної бази. Розглянуто кілька варіантів алгоритмів, які обчислюють напрямки до джерела звуку. Зокрема, на початковому етапі алгоритм кореляції має здатність шукати відповідні дані з введеною затримкою для кожного з каналів прийому звуку. На другому етапі відібрані сигнали перетворюються за допомогою алгоритму ШПФ у спектральне представлення, яке можна інтерпретувати як масив даних, які можна використовувати для розпізнавання типу звуку. На третьому етапі визначається тип джерела звуку. Для оцінки функціональності пристрою в програмному забезпеченні Multisim було змодельовано мікрофонний тракт підсилювача, що демонструє відповідність обраного базового елемента для розробленого пристрою. Розроблений мікропроцесорний пристрій може бути використаний у військовій справі шляхом поєднання алгоритмів і підходів інших авторів до орієнтування.

Запропоновано підсистему безпеки та систему управління розумним будинком (контроль освітлення та води, система контролю проникнення, система кондиціювання повітря з використанням наноструктурованих сенсорів температури та вологості. Виявлено потенційно вразливі місця та створено шляхи їх мінімізації. Схема шифрування інформації на основі запропоновано набір інфраструктури відкритих ключів, яка є практично невразливою з точки зору надійності сервера та його безперебійної роботи. Проведено тестування підсистеми безпеки.

Представлена система SCADA з відкритим вихідним кодом на основі IoT для моніторингу BTS сайту з використанням Raspberry Pi і Arduino IoT. Вимірювальні параметри – температура, вологість, струм, напруга. Дані обробляються та аналізуються в Arduino IoT Cloud через канал зв'язку мережі Wi-Fi. На основі

віджетів інформаційна панель створюється в Arduino IoT Cloud для моніторингу і контролю досліджуваної системи. Також розгорнуто мобільний додаток для віддаленого моніторингу та керування. Передбачено використання світлодіодів для контролю високої температури та низької напруги. Прототип використовується для демонстрації та ілюстрації того, що можна отримати на базовій приймально-передавальній станції (BTS), де напруга повинна бути в межах допустимого значення і температура в межах прийнятного значення.

Запропоновані у роботі методи та засоби інтелектуалізації апробовані на даних інформаційно-вимірювальних системах.

Опубліковані роботи до даного розділу : 205, 207, 208, 210, 211, 212, 213.

ЗАГАЛЬНІ ВИСНОВКИ

У дисертації вирішено актуальну науково-прикладну задачу інтелектуалізації мультисенсорних інформаційно-вимірювальних систем шляхом удосконалення та розроблення методів та засобів ефективної передачі та валідації інформаційних даних між вимірювальними пристроями і системами та архітектурними рівнями IoT. В результаті виконання роботи отримані такі найвагоміші результати:

1. Запропоновано метод нечіткої обробки в поєднанні з динамічними мультимодальними даними сенсорів на рівні прикладної системи в промисловому Інтернеті. Цей метод відстежує виконання програм, визначає поля, на які впливають умовні оператори за допомогою динамічного аналізу виправлень, і фіксує залежності умовних операторів для кращого регулювання створення граматик тестових випадків, що покращує здатність виконувати код на глибшому рівні. Порівняння результатів експерименту показує, що метод підвищує валідність тестів і швидкість покриття коду, а також ймовірність виявлення аномалій у реалізації протоколу.

2. Досліджено метод очищення даних управління талантами в бездротових сенсорних мережах на основі технології інтелекту. Проаналізовано конкретні форми застосування бездротових сенсорних мереж. Представлено характеристики структури бездротових сенсорних мереж та пропонують технологію очищення даних на основі моделі кластеризації. Запропоновано алгоритм видалення запису реплікації на основі кластерів та перевірено точність методів очищення даних. Отримані результати свідчать про ефективність використання досліджуваного методу.

3. Оптимізовано метод довіри на основі туману для компенсування недоліків та вирішення існуючих проблем у споживачів мережних ресурсів для застосування у системах з мультисенсорною конфігурацією. Показано, що довіра щодо поведінки між вузлами встановлюється на рівні бездротових сенсорних мереж, а довіра даних вузлів і об’єктів – у шар туману. Показано, що завдяки детальному аналізу даних у шарі туману можна відстежувати стан довіри всієї мережі, виявляти

атаки на дані та відновлювати вузли неправильної оцінки. Крім того, шар туману може бути побудованим як надійна третя сторона. Результати експерименту показують, що запропонований механізм довіри має ряд переваг, зокрема, зменшення споживання енергії, забезпечення довірчого стану граничних вузлів і мережі, виявлення деяких прихованих атак на дані та відновлення вузлів з неправильною оцінкою.

4. Розроблено методи та засоби практичного застосування бездротових сенсорних систем та мереж, зокрема. побудовано апаратно-програмну систему для визначення положення тіла людини у віртуальному світі, мобільну платформу для визначення рівня радіаційного фону в режимі реального часу, систему захисту розумного будинку та експертну інформаційно-вимірювальну систему на базі технології SCADA для контролю роботи промислових об’єктів в реальному часі. Запропоновані у роботі методи та засоби інтелектуалізації використані у спроектованих інформаційно-вимірювальних системах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Zanella A., Bui N., Castellani A., Vangelista L., & Zorzi M. (2014). Internet of Things for Smart Cities. IEEE Internet of Things Journal, 1, 22-32.

2. Choi M., Gu J., Blaauw D., & Sylvester D.(2015). Wide Input Range 1.7μW 1.2kS/s Resistive Sensor Interface Circuit with 1 cycle/sample Logarithmic Sub-Ranging. IEEE symp. VLSI Circuits, 330-331.

3. Ashwini D., Chidananda M.V., & Kurian M.Z. (2018). Survey on multi sensor based air and water quality monitoring using. Department of Electronics and Communication Engineering SSIT. Indian J.Sci.Res. 17(2), 147-153.

4. Gupta, G. S., & Quan, V. M. (2018). Multi-sensor integrated system for wireless monitoring of greenhouse environment. IEEE. SAS, 1-6.

5. Al-Sharman, M. K., Emran, B. J., Jaradat, M. A., Najjaran, H., Al-Husari, R., & Zweiri, Y. (2018). Precision landing using an adaptive fuzzy multi-sensor data fusion architecture. Applied soft computing, 69, 149-164.

6. Gupta, G. S., & Quan, V. M. (2018). Multi-sensor integrated system for wireless monitoring of greenhouse environment. 2018 IEEE sensors applications symposium (SAS), 1-6.

7. Shruti Sridharan.(2014). Water Quality Monitoring System Using Wireless Sensor Network. International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), l (3), 4.

8. Karthik Kumar R., Chandra Mohan M., Vengateshapandiyan S., Mathan Kumar M., & Eswaran R. (2014). Solar based advanced water quality monitoring system using wireless sensor network. International Journal of Science, Engineering and Technology Research (IJSETR), 3, (3), 2278 – 7798.

9. Zennaro M., Floross A., & Dogaet Gokhan. (2015). Proposed the design of a water quality monitoring system and, building upon the Sunspot technology. Journal of engineering research, 5, 6.

10. Daudi S. Simbeye & Shi Feng Yang (2014.). Water Quality Monitoring and Control for Aquaculture Based on Wireless Sensor Networks. Journal of networks, 9, 4.

11. Kirankumar S. G., & Ramesh P.T. (2013). Wireless Sensor Network System to Monitor The Fish Farm. Int. Journal of Engineering Research and Applications, 3, (5), 194-197.

12. Khetre A.C., & Hate S.G. (2018). Wireless sensor network for water environment monitoring system. International Journal of Engineering & Technology IJET, 7, 8.

13. Brinda D. (2013-2014) Jain proposed the "Real-Time Water Quality Monitoring System using Internet of Things". Jain Department of Electrical Engineering, School of Engineering. Shiv Nadar University, G. Noida.

14. Saravanan, M., Das, A., & Iyer, V. (2017). Smart water grid management using LPWAN IoT technology. In 2017 Global Internet of Things Summit (GIoTS), 1-6.

15. Yoddumnern, A., Chaisricharoen, R., & Yooyativong, T. (2018). A smart WiFi multi-sensor node for fire detection mechanism based on social network. iJOE, 14(10).

16. Encinas, C., Ruiz, E., Cortez, J., & Espinoza, A. (2017). IoT system for the monitoring of water quality in aquaculture. In IEEE Conference on aquaculture, 6, 507-513).

17. Konyha, J. (2016). Grid-based wide area water quality measurement system for surface water. In 2016 17th international carpathian control conference (ICCC), 341-344.

18. Kamaludin, K. H., & Ismail, W. (2017). Water quality monitoring with internet of things (IoT). In 2017 IEEE Conference on Systems, Process and Control (ICSPC), 18-23.

19. Rasin Z., & Abdullah M. (2016). Water Quality Monitoring System Using Zigbee BasedWireless Sensor Network. International Journal of Engineering & Technology IJET -IJENS, 09, 10.

20. Prasad R., Baig M., Mishra R., Rajalakshmi P., Desai U., & Merchant S. (2011). Real Time Wireless Air PollutionMonitoring System. Ictact Journal On Communication Technology: On Next Generation Wireless Networks And Applications, 2, 2.

21. Devarakonda, S., Sevusu, P., Liu, H., Liu, R.,Iftode, L., & Nath, B. (2013). Real-time Air QualityMonitoring Through Mobile Sensing in Metropolitan Areas, in Proceedings of the 2ndACM SIGKDD International Workshop on Urban Computing, 15.

22. Torfs T., Sterken T., Brebels S., Santana J., Vanden H., Spiering V., Bertsch N., Trapani D., & Zonta, D. (2012). Low power wireless sensor network for building monitoring. IEEE Sens. J., 13(3), 909–915.

23. Wu F., Rüdiger C., & Yuce M.R. (2017). Real-time performance of a self-powered environmental IoT sensor network system. Sensors 17(2), 282.

24. Kim J.Y., Chu C.H., & Shin S.M.(2014) ISSAQ: an integrated sensing systems for real-time indoor air quality monitoring. IEEE Sens. J.,14(12), 4230–4244.

25. Silvani X., Morandini F., Innocenti E., & Peres S. (2015). Evaluation of a wireless sensor network with low cost and low energy consumption for fire detection and monitoring. Fire Technol. 51(4), 971–993.

26. Jelicic V., Magno M., Brunelli D., Paci G., & Benini L. (2013). Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring. IEEE Sens. J., 13(1), 328–338.

27. Sandeep K. P. (2019). Design of a Multi-Sensor based Smart Home System using Artificial Intelligence International Journal for Innovative Research in Science & Technology, 5, 10.

28. Llinas J., & Hall D. L. (1998). An Introduction to Multi-sensor Data Fusion,“ in Proc. 1998 IEEE International Symposium on Circuits and Systems (ISCAS ‘98), 6, 537-540.

29. Chong, C. Y., Chang, K. C., & Mori, S. (2012, May). Fundamentals of distributed estimation and tracking. Signal Processing, Sensor Fusion, and Target Recognition XXI, 8392, 437-450.

30. Ahi, B., & Haeri, M. (2022). Practical distributed maneuvering target tracking using delayed information of heterogeneous unregistered sensors. Signal Processing, 193, 108419.

31. Roecker, J. A., & Theisen, D. K. (2014). Multiple sensor tracking architecture comparison. IEEE Aerospace and Electronic Systems Magazine, 29(9), 28-33.

32. Adam, M. S., Anisi, M. H., & Ali, I. (2020). Object tracking sensor networks in smart cities: Taxonomy, architecture, applications, research challenges and future directions. Future Generation Computer Systems, 107, 909-923.

33. OMG Systems Modeling Language (OMG SysML) Version 1.3. OMG, 2012.

34. Marler R. T., & Arora J. S. (2004). Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, 26 (6), 369-395.

35. Alam, F., Mehmood, R., Katib, I., Albogami, N. N., & Albeshri, A. (2017). Data fusion and IoT for smart ubiquitous environments: A survey. Ieee Access, 5, 9533-9554.

36. Berkner K., & Wells R. (1998). Wavelet transforms and denoising algorithms. Proceedings of the Conference Record of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1–4 (2), 1639–1643.

37. Yan X., Xiong W., Hu L., Wang F., Zhao K. (2015). Missing value imputation based on gaussian mixture model for the internet of things. Math. Probl. Eng. 2015, 548-605.

38. Gao Z., Cheng W., Qiu X., & Meng L. (2015). A Missing Sensor Data Estimation Algorithm Based on Temporal and Spatial Correlation. Int. J. Distrib. Sens. Netw. 2015.

39. Mary I., & Arockiam L. (2017). Imputing the missing data in IoT based on the spatial and temporal correlation. In Proceedings of the IEEE International Conference on Current Trends in Advanced Computing, ICCTAC 2017, Bangalore, India, 2018, 1–4.

40. Li Y., & Parker L.E. (2014-2015). Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks. Inf., 64–79.

41. Li P., Stuart, & Allison E.A.(2015). Multiple imputation: A flexible tool for handling missing data. JAMA—J. Am. Med. Assoc., 314, 1966–1967.

42. Vijayakumar N.N., & Plale B. (2008). Missing event prediction in sensor data streams using kalman filters. Knowl. Discov. Sens. 149-170.

43. Halatchev M., & Gruenwald L. (2005). Estimating Missing Values in Related Sensor Data Streams. The University of Oklahoma, Norman, OK, USA, 83–94.

44. Al-khatib A.A., Mohammed B., & Abdelmajid K. (2020). A Survey on Outlier Detection in Internet of Things Big ata. In Big Data-Enabled Internet of Things; IET, London, UK, 265–272.

45. Shahraki A., & Haugen O. (2019). An outlier detection method to improve gathered datasets for network behavior analysis in IoT. J. Commun. 14, 455–462.

46.Hasan M., Islam M.M.,. Zarif M.I, & Hashem M.A.(2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things, 7, 59-100.

47. Gaddam A., Wilkin T., Angelova M., & Gaddam J. (2020). Detecting Sensor Faults, Anomalies and Outliers in the Internet of Things: A Survey on the Challenges and Solutions. Electronics, 9, 511.

48. Nithyakalyani S., & Gopinath B. (2020). Analysis of Node Clustering Algorithms on Data Aggregation in Wireless Sensor Network; NISCAIR-CSIR: New Delhi, India. Sensors, 20 (6076), 21 – 23.

49. Zhong H., Shao L., Cui J., & Xu Y. (2018). An e_cient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks. J. Parallel Distrib. Comput., 111, 1–12.

50. Liu Y., Gong X., & Xing C. (2014). A novel trust-based secure data aggregation for Internet of Things. In Proceedings of the 9th International Conference on Computer Science and Education, ICCCSE 2014, Vancouver, BC, Canada, 22–24; 435–439.

51. Schimbinschi F., Nguyen X.V., Bailey J., Leckie C., Vu H., & Kotagiri R. (2015). Tracforecasting in complex urban networks: Leveraging big data and machine learning. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1, 1019–1024.

52. Khattak H.A., Hussain R., Ameer Z. (2018). Internet of vehicles: Integrated services over vehicular Ad Hoc Networks. In Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST 2018, Springer: Berlin/Heidelberg, Germany, 224, 61–73.

53. Dalgleish, M., & Hoose, N. (2009). Highway tra_c monitoring and data quality. Tra_c Eng. Control., 50, 29–30.

54. Alinia B., H.Hajiesmaili M., Khonsari A., & Crespi N. (2017). Maximum-quality tree construction for deadline-constrained aggregation in WSNs. IEEE Sens. J., 17, 3930–3943.

55. Sicari S., Grieco, Boggia L.A., Coen-Porisini G., & DyDAP A. (2012). A dynamic data aggregation scheme for privacy aware wireless sensor networks. J. Syst. Softw., 85, 152–166.

56. Wu W., Cao J., Wu H., & Li J.(2013). Robust and dynamic data aggregation in wireless sensor networks: A cross-layer approach. Comput. Netw., 57, 3929–3940.

57. Xu. J., Yang G., Chen Z.Y., Chen L., Yang Z., & Yichang. (2011). Performance analysis of data aggregation algorithms in wireless sensor networks. In Proceedings of the 2011 International Conference on Electrical and Control Engineering, , ICECE 2011, 4619–4622.

58. Satapathy, S. S., & Sarma, N. (2006). TREEPSI: tree based energy efficient protocol for sensor information. In 2006 IFIP international conference on wireless and optical communications networks, 4.

59. Messina, D., Ortolani, M., & Re, G. L. (2007). A network protocol to enhance robustness in tree-based WSNs using data aggregation. In 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems, 1-4.

60. Tang F., You I., Guo S., M.Guo, Ma Y. Intell J. Manuf. (2012). A chain-cluster based routing algorithm for wireless sensor networks, 23, 1305–1313.

61. Guo W., Xiong N., Vasilakos A., Chen G., & Cheng H. (2011). Multi-source temporal data aggregation in wireless sensor networks. Wirel. Pers. Commun, 56, 359–370.

62. Rajkamal R., Ranjan P., & Comput J. (2012). Energy e_cient aggregation for continuous monitoring applications of wireless sensor network. Sci., 8, 55–60.

63. Dehkordi S.A., Farajzadeh K., Rezazadeh J., Farahbakhsh R., Sandrasegaran K., & Dehkordi M.(2020) A survey on data aggregation techniques in IoT sensor networks. Wirel. Netw., 26, 1243–1263.

64. Ding W., Jing X., Yan Z., & Yang L. (2019). A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. Inf. Fusion, 51, 129–144.

65. Qi J., Yang P., Newcombe L., Peng X., Yang Y., & Zhao Z.(2020). An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure. Inf. Fusion, 55, 269–280.

66. De Paola A., Ferraro P., Gaglio S., Re G.L., & Das S. (2017). An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments. IEEE Trans. Mob. Comput., 16, 1502–1515.

67. Wang M., Perera C., Jayaraman P.P., Zhang M., Strazdins P., Shyamsundar R.K., Ranjan R., & Distrib J (2016). City data fusion: Sensor data fusion in the internet of things. Int. Syst. Technol., 7, 15–36.

68. Diez-Olivan A., Del Ser J., Galar D., & Sierra B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. Inf. Fusion, 50, 92–111.

69. Alkhamisi A., Nazmudeen M.S.H., & Buhari S.M. (2016). A cross-layer framework for sensor data aggregation for IoT applications in smart cities. In Proceedings of the IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life.

70. Misbahuddin S., Zubairi J., Saggaf A., Basuni J., Sulaiman A., & Al-Sofi A. (2015). IoT based dynamic road tracmanagement for smart cities. In Proceedings of the 2015 12th International Conference on High-Capacity Optical Networks and Enabling/Emerging Technologies, HONET-ICT 2015. Sensors 2020, 20 (6076), 22 – 23.

71. Aggarwal C.C. (2013) An introduction to sensor data analytics. Managing and Mining Sensor Data; Springer: Boston, MA, USA, 1–8.

72. Kanawaday A., & Sane A. (2018). Machine learning for predictive maintenance of industrial machines using IoT sensor data. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, 2017, 87–90.

73. Yu T., Wang X., & Shami A. (2017). Recursive Principal Component Analysis-Based Data Outlier Detection and Sensor Data Aggregation in IoT Systems. IEEE Internet Things J., 4, 2207–2216.

74. Balakrishna S., Thirumaran M., & Solanki V. (2020). IoT sensor data integration in healthcare using semantics and machine learning approaches. Intell. Syst. Ref. Libr., 165, 275–300.

75. Biosystems, Appl. Note AN20010827, 1-1.

76. X.-B. Jin & Y. Gao, Eds. (2020). Multisensor Information Fusion. Basel, Switzerland: MDPI, 1034-1056.

77. Castanedo F., & World J. (2013). A Review of Data Fusion Techniques, Sci. 1–19.

78. Mitchell H. B., & Springer. (2007) Multi-Sensor Data Fusion: An Introduction. New York, New York, USA.

79. Harris C. J., Bailey A., & Dodd T. J. (2015). Multi-sensor data fusion in defence and aerospace. The Aeronautical Journal, 102, 229-244.

80. Mahmoud M. S., & Khalid H. M. (2015). Distributed Kalman filtering: a bibliographic review. IET Control Theory & Applications, 7(4), 483-501.

81. Roecker J. A., & Theisen D. K. (2014). Multiple sensor tracking architecture comparison. IEEE Aerospace and Electronic Systems Magazine, 29, 9, 28-33.

82. Nigischer C., Bougain S., Riegler R., Stanek P., & Grafinger M. (2021). Multi-domain simulation utilizing SysML: state of the art and future perspectives. Procedia CIRP. 2021 (100), 319-324.

83. Marler R. T., & Arora J. S. (2004). Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, 26 (6), 369-395.

84. Yang S., & Zhang Y. (2010). Wireless Measurement and Control System for Environmental Parameters in Greenhouse. Proceedings of the Measuring Technology and Mechatronics Automation (ICMTMA), 2, 1099-1102.

85. Anuj K.(2010). et al. Prototype Greenhouse Environment Monitoring System. Proceedings of the International Multi Conference of Engineering and Computer Scientist., 2, 17-19.

86. Yoddumnern, A., Chaisricharoen, R., & Yooyativong, T. (2018). A smart WiFi multi-sensor node for fire detection mechanism based on social network. iJOE, 14(10).

87. Liggins M. E., Hall D. L., & Llinas J. (2009). Handbook of Data Fusion: Theory and Practice. Boca Raton, Florida, USA: CRC Press.

88. Liu Z., Xiao G., Liu H., & Wei H. (2022). Multi-sensor measurement and data fusion. IEEE Instrumentation & Measurement Magazine, 25, (1), 28-36.

89. Castanedo F. (2013). A Review of Data Fusion Techniques. Sci. World J., 20, 1-19.

90. Ye J., Dobson S., & McKeever S. (2012). Situation identification techniques in pervasive computing: A review. Pervasive Mob. Comput., 8, 36–66.

91. Xie S., & Chen Z.(2017). Anomaly detection and redundancy elimination of big sensor data in internet of thing. ISSU1703.03225.

92. Hromic H., D. Phuoc Le, Serrano M., Antoni´c, A,. Žarko I.P., Hayes C., & Decker S.(2015). Real time analysis of sensor data for the Internet of Things by means of clustering and event processing. In Proceedings of the IEEE International Conference on Communications, 685–691.

93. Qanbari S., Behinaein N., Rahimzadeh R., Dustdar S., & Gatica S. (2015). Linked Sensed Data Enrichment and Analytics Middleware for IoT Gateways. In Proceedings of the 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, 38–43.

94. Sekiyama M., Kim B.K., Irie S., & Tanikawa T. (2015). Sensor data processing based on the data log system using the portable IoT device and RT-Middleware. In Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence, 46–48.

95. Ding Z.;, Xu J., Yang Q., & SeaCloud D. (2013). A database cluster framework for managing and querying massive heterogeneous sensor sampling data. Journal Supercomput, 66, 260–1284.

96. Zhu T., Dhelim S., Zhou Z., Yang S., & Ning H.(2017). An Architecture for Aggregating Information from Distributed Data Nodes for Industrial Internet of Things. Comput. Electr. Eng., 58, 337–349.

97. Sharma S., & Chen K. A. (2018) ShethToward practical privacy-preserving analytics for IoT and cloud-based healthcare systems. IEEE Internet Comput., 22, 42–51.

98. Zhang C., Liu Y., Wu F., Fan W., Tang J., & Liu H. (2019). Multi-Dimensional Joint Prediction Model for IoT Sensor Data Search. IEEE Access, 7, 90863–90873.

99. Shyamalagowri. M., & Rajeswari R.(2016). Unscented Kalman filter based nonlinear state estimation case study-Nonlinear process control reactor (Continuous stirred tank reactor). In Proceedings of the 10th International Conference on Intelligent Systems and Control, ISCO 2016, Coimbatore, India, 7–8 January 2016.

100. Kumarage H., Khalil I., Alabdulatif A., Tari Z., & Yi X. (2016). Secure Data Analytics for Cloud-Integrated Internet of Things Applications. IEEE Cloud Comput., 3, 46–56.

101. Patni H., Henson C., & Sheth A. (2010). Linked sensor data. In Proceedings of the 2010 International Symposium on Collaborative Technologies and Systems, Chicago, IL, USA, 362–370.

102. Qin Y., Sheng Q.Z., Falkner N.J., Dustdar S., Wang H., & Vasilakos A.V. (2016). When Things Matter: A Survey on Data-Centric Internet of Things. J. Netw. Comput. Appl., 64, 137–153.

103. He J., Wei J., Chen K., Tang Z., Zhou Y., & Zhang Y. (2018). Multitier Fog ComputingWith Large-Scale IoT Data Analytics for Smart Cities. IEEE Internet Things Journal, 5, 677–686.

104. Yoon, G., Choi, D., Lee, J., Choi, H., & Sens J. (2020). Management of IoT Sensor Data Using a Fog Computing Node. Sensors, 20 (6076), 23-33.

105. Raafat H.M., Hossain M.S., Essa E., Elmougy S., Tolba A.S., Muhammad G., & Ghoneim A. (2017). Fog Intelligence for Real-Time IoT Sensor Data Analytics. IEEE, 5, 24062–24069.

106. Singh S.P., Nayyar A., Kumar R., & Sharma A.(2019). Fog computing: From architecture to edge computing and big data processing. J. Supercomput., 75, 2070–2105.

107. Kaur A., Singh P., & Nayyar A. (2020). Fog Computing: Building a Road to IoT with Fog Analytics. In Fog Data Analytics for IoT Applications. Springer: Singapore, 59–78.

108. Qureshi B. (2019) Profile-based Power-aware Workflow Scheduling Framework for Energy-E_cient Data Centers. Future Gener. Comput. Syst., 94, 453–467.

109. Cai, H. Xu, B. Jiang, L. & Vasilakos, A.V.(2017). IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges. IEEE Internet Things J., 4, 75–87.

110. Djedouboum A.C., Abba Ari A.A., Gueroui A.M., Mohamadou A., & Aliouat Z. (2018). Big Data Collection in Large-Scale Wireless Sensor Networks. Sensors, 18, 44-74.

111. Qureshi B. (2018). An a_ordable Hybrid Cloud based Cluster for Secure Health Informatics Research. Int. J. Cloud Appl. Comput., 8, 27–46.

112. Bytes A., Adepu S., & Zhou J. (2019). Towards Semantic Sensitive Feature Profiling of IoT Devices. IEEE Internet Things J., 6, 8056–8064.

113. Abu-Elkheir M., Hayajneh M., & Ali N.A.. Data Management for the Int.

114. 1.Dhananjay, G. V. (2001). Programming and Customizing A.V.R. Microcontroller, 22, Mraw-Hill (USA).

115. Brennan T. M., Ernst J. M., Day C. M., Bullock D. M., Krogmeier J.V., & Martchouk M. (2009). Influence of Vertical Sensor Placement on Data Collection Efficiency from Bluetooth MAC Address Collection, ASCE Journal of Transportation Engineering, 136, 1104-1109.

116. Sung W. T., & Tsai, M. H. (2011). Multi-Sensor Wireless Signal Aggregation for Environmental Monitoring System via Multi-Bit Data Fusion. Applied Mathematics and Information Sciences, 5, 589–603.

117. Forlano L. (2009). WiFi Geographies: When Code Meets Place. The Information Society, 25, 344–352.

118. Гераїмчук М.Д., Івахів О.В., Паламар М.І., & Шевчук Б.М. (2010). Основи побудови перспективних безпроводових сенсорних мереж. ЕКМО, 124.

119. Matsuda, T., Noguchi, T., & Takine, T. (2011). Survey of network coding and its applications. IEICE transactions on communications, 94(3), 698-717.

120. Talooki, V. N., et al. (2015). Security concerns and countermeasures in network coding based communication systems: A survey. Computer Networks, 83 (4), 422-445.

121. Ustun, T. S., & Khan, R. H. (2015). Multiterminal hybrid protection of microgrids over wireless communications network. IEEE Transactions on Smart Grid, 6(5), 2493-2500.

122. Ahlswede R., Cai N., Li S.-Y.R., & Yeung R.W. (2000). Network information flow. IEEE Trans. Inform. Theory, 46 (6), 1204 – 1216.

123. Jaggi S., Langberg M., Katti S., Ho T., Katabi D., & Medard M. (2008). Resilient Network Coding in the Presence of Byzantine Adversaries. IEEE Transactions on Information Theory, 54(7), 2596–2603.

124. Яцків В.В. (2013). Метод мережного кодування в системі залишкових класів. Комп'ютерні системи та мережі. Національного університету «Львівська політехніка», 773, 157-164.

125. Akyildiz I. F., & Vuran M. C. (2010). Wireless Sensor Networks, New York: John Wiley & Sons, 571 p.

126. Talooki, V. N., et al. (2015). Security concerns and countermeasures in network coding based communication systems: A survey. Computer Networks, 83 (4), 422–445.

127. Kang J., Zhou B., Ding Z., & Lin S. (2008). LDPC coding schemes for error control in a multicast network. IEEE International Symposium on Information Theory, 822- 826.

128. Fragouli C., Le Boudec J. Y., & Widmer J. (2008). Network coding: an instant primer. ACM SIGCOMM Computer Communication Review, 36 (1), 63-68.

129. Karvonen, H., Shelby, Z., & Pomalaza-Raez C. (2004). Coding for energy efficient wireless embedded networks. In: Wireless Ad-Hoc Networks, 2004 International Workshop on. IEEE, 300-304.

130. Kang J., Zhou B., Ding Z., & Lin S. (2008).LDPC coding schemes for error control in a multicast network. IEEE International Symposium on Information Theory, 822 – 826.

131. Яцків В. В & Яцків Н. Г. (2013). Метод кодування зображень в системі залишкових класів. Труды *МНПК* «Современные информационные и электронные технологии» (СИЭТ-2013), 44-46.

132. Яцків В.В. (2010). Метод кодування та передавання мультимедійних даних в безпровідних сенсорних мережах. В.В.Яцків, А.О. Саченко, Су Цзюнь. Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: тези доповідей V Міжнародної науково- практичної конференції. – Запоріжжя: ЗНТУ, 133-135.

133. Campobello, G., Leonardi, A., & Palazzo, S. (2010). Energy Saving and Reliability in Wireless Sensor Networks Using a CRT-based Packet Splitting Algorithm. University of Messina, Italy.–2010.

134. Misra S., Reisslein M., & Xue G. (2008). A survey of multimedia streaming in wireless sensor networks. IEEE Communications Surveys and Tutorials, 10 (1–4), 18–39.

135. Яцків В.В. (2010). Метод підвищення надійності передачі даних в безпровідних сенсорних мережах на основі системи залишкових класів. Радіоелектроніка та інформатика, 2, 32–35.

136. Akyildiz F., Pompili D., & Melodia T. (2005). Underwater Acoustic Sensor Networks: Research Challenges. Ad Hoc Net., 3 (3), 257–79.

137. Ahmad, M. W., Mourshed, M., Mundow, D., Sisinni, M., & Rezgui, Y. (2016). Building energy metering and environmental monitoring. A state-of-the-art review and directions for future research. Energy and Buildings, 120, 85–102.

138. Al Dakheel, J., Del Pero, C., Aste, N., & Leonforte, F.(2020). Smart Buildings Features and Key Performance Indicators: A Review. Sustainable Cities and Society, 102328. doi:10.1016/j.scs.2020.102328.

139. Alexakis G., Panagiotakis S., Fragkakis A., Markakis E., & Vassilakis, K. (2019). Control of Smart Home Operations Using Natural Language Processing, Voice Recognition and IoT Technologies in a Multi-Tier Architecture. Designs 2019, 3 (32).

140. Apanaviciene, R., Vanagas, A., & Fokaides, P.A. (2020). Smart Building Integration into a Smart City (SBISC): Development of a New Evaluation Framework. Energies 2020, 13, 2190.

141. Awada, M., Becerik-Gerber, B., Hoque, S., O'Neill, Z., Pedrielli, G., Wen, J., & Wu, T. (2021). Ten questions concerning occupant health in buildings during normal operations and extreme events including the COVID-19 pandemic. Building and Environment 188, 107480.

142. Retrieved from URLhttps://nodon.fr/nodon/capteur-temperature-humidite-enocean (Accessed on 16 June 2021).

143. Retrieved from URL https://www.eltako.com/fileadmin/downloads/fr/Fiches_techniques/Fichetechnique_Eltako-radio_FCO2TF65.pdf (Accessed on 16 June 2021).

144. Retrieved from URL https://www.eltako.com/fileadmin/downloads/fr/Fiches_techniques/Fiche_technique_Eltako-radio_FWZ14-65A.pdf (Accessed on 16 June 2021).

https://nodon.fr/nodon/detecteur-douverture-portes-et-fenetres-enocean (Accessed on 16 June 2021).

145. https://market.thingpark.com/media//datasheet//d/o/do13-421b-e.pdf (Accessed on 16 June 2021).

146. Retrieved from URL http://nano-sense.com/wp-content/uploads/2018/08/E4000-Fiche-produit-detaillee.pdf Accessed on 16 June 2021).

147. https://www.eltako.com/fileadmin/downloads/fr/_bedienung/FRW-ws_30000053-2_frz.pdf (Accessed on 16 June 2021).

148. Diachok R., & Klym H. (2022). Current state of development of intelligent information and measuring systems for environmental monitoring with multisensor configuration. Visnyk of Kherson National Technical University, 2(81), 55-69.

149. Bajc, T., Banjac, M., Todorovic, M., & Stevanovic, Z. (2019). Experimental and statistical survey on local thermal comfort impact on working productivity loss in university classrooms. Thermal Science, 23, 379–392.

150. Balikhina, T., Al Maqousi, A., AlBanna, A., & Shhadeh, F. (2017, December). System architecture for smart home meter. In 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), 1-5.

151. Бабак В.П., Марченко Б.Г., & Фриз М.Є. (2004).Теорія ймовірностей, випадкові процеси та математична статистика, 287.

152. Марченко Н. Б. Нечипорук В.В.,Нечипорук О. П., & Пепа Ю. В. (2004). Методи оцінювання точності інформаційно-вимірювальних. Наукова думка, 200.

153. Марченко Н.Б., & Мислович М.В. (2004). Особливості використання нестаціонарних лінійних випадкових процесів для моделювання процесів в електроенергетиці. Технічна електродинаміка. Тематичний випуск: “Силова електроніка та енергоефективність”. Ч. 3, 97-100. Наука, 1982, 296.

154. Марченко Н.Б. (2006). Анализ точностных характеристик при моделировании линейных субгауссовых случайных процессов и их использование в информационно-измерительных системах. Электронное моделирование, 26 (6), 63-71.

155. Марченко Н.Б. (2004). Використання моделей субгауссівських процесів при моделюванні інформаційних сигналів. Технічна електродинаміка. Тематичний випуск: “Проблеми сучасної електротехніки”. Ч. 5, 117-120.

156. Марченко Н.Б. (2006). Особенности моделирования субгауссовых случайных величин и процессов. Сборник тезисов докладов по материалам 10-й

Юбилейной международной научной конференции “Теория и техника передачи, приема и обработки информации”. Ч. 2. Харьков-Туапсе: Харьковский национальный университет радиоэлектроники, 159-160.

157. Марченко Н.Б. (2004). Деякі особливості використання субгауссових випадкових процесів в інформаційно-вимірювальних системах. Вісник Тернопільського державного технічного університету, 9 (4), 139-146.

158. Ціделко В.Д., & Яремчук Н.А. (2002). Невизначеність вимірювання. Обробка даних і подання результату вимірювання: Монографія. Видавництво «Політехніка», 176.

159. Марченко Н.Б. (2003). Про другий наслідок з нерівності Чебишева та його використання при оцінці точності вимірювань. Тези доповідей V Міжнародної науково-технічної конференції “ABIA-2003”, 1. Київ: Національний авіаційний університет, 11.101-11.104.

160. Марченко Б.Г., Марченко Н.Б., & Фриз М.Є. (2004). Спеціальні глави математики: Навчальний посібник. Тернопіль: ТДТУ ім. Івана Пулюя, 159.

161. Q. Li, Y. Tian, Q. Wu, Q. Cao, H. Shen, & H. Long. (2020). A Cloud-Fog-Edge closed-loop feedback security risk prediction method. IEEE 8(1), 29004–29020.

162. Ericd, Knapp et al. (2014). Industrial Network Security. Smart Power Grids, SCADA and other IIS Key Infrastructures, CA: NDIP, 18-26.

163.Qianmu Li, Shunmei Meng, Shuo Wang, Jing Zhang & Jun Hou. (2019). CAD command-level anomaly detection for vehicle- road collaborative charging network. IEEE, 7, 34910–34924.

164. Raval S. (2015). Black Energy a threat to Industrial Control Systems network security. International Journal of Advance Research in Engineering. Sci Technol.Vjl. 2, 31–34 .

165. IIS-CERT. Information products [EB/OL], (2018). Retrieved from URL https://IIS-cert.us-cert.gov/

166.China National Vulnerability Database. (2018). Vulnerability of Industrial Internet Industry [EB/OL], Retrieved from URL http://IIS.cnvd.org.cn/

167. Wan S., Li M., Liu G, & Wang C. (2019). Recent advances in consensus protocols for blockchain: a survey. Wireless Networks, 1-15.

168. Garibaldi J.M. (2019). The need for fuzzy AI, IEEE/CAA J. Autom. Sinica, 6(3), 610–622.

169. Liu, B., Shi, L., Cai, Z., & Li, M. (2012, November). Software vulnerability discovery techniques: A survey. 2012 fourth international conference on multimedia information networking and security, 152-156.

170. Cui Baojiang, Zhang Xiangqian, Zhang Tianxin, & Zhang Qin. (2018). Embedded system vulnerability mining technology based on in-memory fuzzing test. The 13th International Conference on Broadband, Wireless Computing, Communication and Applications. Taichung, Taiwan, 439-449.

171.Aitel D. (2014). An introduction to SPIKE, the fuzzer creation kit [EB/OL]. Retrieved from URL http://www.blackhat.com/presentations/bh-usa-02/bh-us-02-aitel-spike.ppt

172. Devarajan G. & Unraveling J. (2015). SCADA protocols: using Sulleyfuzzer [EB/OL], Retrieved from URL http://www.defcon.org/html/defcon-15/dc15speakers.html

173. Peach. (2015). Retrieved from URL http://www.peachFuzzer.com

174. Zalewski F. (2017). American fuzzy lop [EB/OL]. Retrieved from URL http://lcamtuf.coredump.cx/afl/, 2017-12-25

175. Gan S., Zhang C., & Qin X, et al. (2018). CollAFL: path sensitive fuzzing, in 2018 IEEE Symposium on Security and Privacy (SP) (IEEE Computer Society, San Fransisco, CA, USA), 660–677.

176. Byres, E. J., Hoffman, D., & Kube, N. (2006). On shaky ground–a study of security vulnerabilities in control protocols. Proc. 5th American Nuclear Society Int. Mtg. on Nuclear Plant Instrumentation, Controls, and HMI Technology, 1-7.

177. Michael Toecker. (2013). Response fuzzing [EB/OL]. Retrieved from URL http://www.digitalbond.com/blog/response-Fuzzing/

178. Qianmu Li, Shunmei Meng, Sainan Zhang, Jun Hou, & Lianyong Qi. (2019). Complex attack linkage decision-making in edge computing networks. IEEE, 7, 12058 – 12072.

179. Newsome J., & Song J. (2014). Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. NDSS, 5, 3–4.

180. Chen L., Liu S., & Xiao D. (2014). A Cisco IOS heuristic fuzz test method. Comput Eng., 40, 68–73.

181. Li Q., Wang Y., Ziyuan P., Wang S., & Zhang (2019). A time series association state analysis method in Smart Internet of electric vehicle charging network attack. Transport Res Record, 2673, 217–228 .

182. Cha S.K., Woo M., & Brumley D. (2015). Program-adaptive mutational fuzzing, in 2015. IEEE Symposium on Security and Privacy, San Jose, CA.

183. Brumley D., Jager I., Avgerinos T., et al. (2011). BAP: a binary analysis platform, in International Conference on Computer Aided Verification, Berlin, Heidelberg, 463-469.

184. Liu, H., Kou, H., Yan, C., & Qi, L. (2019). Link prediction in paper citation network to construct paper correlation graph. EURASIP Journal on Wireless Communications and Networking, 2019(1), 1-12.

185. Wan, S., Li, X., Xue, Y., Lin, W., & Xu, X. (2020). Efficient computation offloading for Internet of Vehicles in edge computing-assisted 5G networks. The Journal of Supercomputing, 76, 2518-2547.

186. Qi, L., Zhang, X., Li, S., Wan, S., Wen, Y., & Gong, W. (2020). Spatial-temporal data-driven service recommendation with privacy-preservation. Information Sciences, 515, 91-102.

187. Klym H., & Diachok R. (2022). Dynamic search for errors in industrial internet protocols for application in multisensor control systems. Computer systems and information technologies, 3, 65-74.

188. Upton D.W., Saeed B.I., Mather P.J., et al. (2018). Wireless sensor network for radiometric detection and assessment of partial discharge in high-voltage equipment. Radio Sci., 53(3), 357–364.

189. Mekikis P.V., Kartsakli E., Antonopoulos A., et al. (2018). Connectivity analysis in clustered wireless sensor networks powered by solar energy. IEEE Trans. Wirel. Commun, 17(4), 2389–2401.

190. Aygör D., Rehman S.U., & ÇelebİF.V. Impact of buffer management solutions on MAC Layer Performance in Wireless Sensor Networks. IEICE Transac. Commun. E101.B(9), 2058–2068 (2018).

191. Alomari A., Comeau F., Phillips W., et al. (2018). New path planning model for mobile anchor-assisted localization in wireless sensor networks. Wirel. Netw, 8, 1–19.

192. Kumar L., Sharma V., & Singh A. (2018). Cluster-based single-sink wireless sensor networks and passive optical network converged network incorporating sideband modulation schemes. Opt. Eng. 57(2), 1.

193. Lee W.K, Schubert M.J.W., Ooi B.Y., et al. (2018). Multi-source energy harvesting and storage for floating wireless sensor network nodes with long range communication capability . IEEE Trans. Ind. Appl. 54(3), 2606–2615.

194. Zhang W., Yang J., Fang Y., et al. (2017). Analytical fuzzy approach to biological data analysis. Saudi J. Biol. Sci., 24(3), 563–573.

195. Diachok R., & Klym H. (2022). Data cleaning method in wireless sensor-based on intelligence technology // Measuring Equipment and Metrology, 83(2), 2, 5-10.

196. Дячок Р. В., & Клим Г. І. Метод очищення мережевих даних на базі технології інтелекту. Всеукраїнська науково-практична конференція молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості», 13 жовтня 2022, Івано-Франківськ, Україна, 209-210.

197. Hu P., Dhelim S., Ning H., & Qiu. (2017). Survey on fog computing: Architecture, key technologies, applications and open issues. Journal of Network and Computer Applications, 98.

198. Ridhawi I. A., Aloqaily M., & Boukerche A. (2019). Comparing fog solutions for energy efficiency in wireless networks: Challenges and opportunities. IEEE Wireless Communications, 26 (6), 80–86.

199. Teslya N., & Ryabchikov I. (2017). Blockchain-based platform architec- ture for industrial iot, in 2017 21st Conference of Open Innovations Association (FRUCT), 321–329.

200. Tseng, L., Wong, L., Otoum, S., Aloqaily, M., & Othman, J. B. (2020). Blockchain for managing heterogeneous internet of things: A perspective architecture. IEEE network, 34(1), 16-23.

201. Basir, R., Qaisar, S., Ali, M., Aldwairi, M., Ashraf, M. I., Mahmood, A., & Gidlund, M. (2019). Fog computing enabling industrial internet of things: State-of-the-art and research challenges. Sensors, 19(21), 4807.

202. O'donovan, P., Gallagher, C., Bruton, K., & O'Sullivan, D. T. (2018). A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications. Manufacturing letters, 15, 139-142.

203. Mehdipour F., Javadi. B., & Mahanti A. (2016). FOG-Engine. Towards Big Data Analytics in the Fog. In Proceedings of the Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing. Auckland, New Zealand, 640–646.

204. Ko, K., Son, Y., Kim, S., Lee, Y., & Dis C.O. (2017). A distributed and concurrent offloading framework for mobile edge cloud computing. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 763–766.

205. Diachok R., & Klym H. (2022). Modified fog-based trust method of data monitoring for multi-sensor configuration systems. Measuring Equipment and Metrology, 83(4), 47-55.

206. Трач І.Б., Клим Г.І., Дячок Р.В., Карбовник І.Д.(2022). Проєктування мікропроцесорних пристроїв для визначення напрямку до джерела звуку. Військово-технічний збірник, 27, 35-45.

207. Diachok R., Trach I., Klym H., Karbovnyk I., & Dunets R. (2018). Hardware

and software complex of intellectualized ornithopter-type UAV for military applications. Electronics and information technologies, 10, 31-40.

208. Diachok R., Klym H. (2022). Monitoring trust status during Fog level data analysis of the sensor network // Proceedings of the 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), 1-6.

209. Diachok R., Klym H., Vasylchyshyn I., & Karbovnyk I. (2022). Definition system of human body position in virtual reality. Proceedings of the16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 358-361.

210. Diachok R., Dunets R., & Klym H. (2018). System of detection and scanning bar codes from raspberry Pi web camera. Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 184-187.

211. Diachok R., Klym H., & Vasylchyshyn I. (2021). Real-time mobile-based platform for determining level and location of radiation background. Proceedings of the 22nd International Conference on Computational Problems of Electrical Engineering (CPEE), 1-4.

212. Трач І.Б., Клим Г.І., Дячок Р.В., Карбовник І.Д. (2022). Проєктування мікропроцесорних пристроїв для визначення напрямку до джерела звуку. Військово-технічний збірник, 27, 35-45.

213. Klym H., Dunets R., Horbatyi I., & Diachok R. Security subsystem and smart home management system. Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 2018, 194-197.

214. Yadav G. & Paul (2021). Architecture and Security of SCADA Systems: A Review. International Journal of Critical Infrastructure Protection, 430-433.

ДОДАТОК А. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

1. Diachok R., Klym H. Data cleaning method in wireless sensor-based on intelligence technology // Measuring Equipment and Metrology, 2022, 83(2), No 2, p. 5-10. https://science.lpnu.ua/uk/istcmtm/vsi-vypusky/vypusk-83-no2-2022/data-cleaning-method-wireless-sensor-based-intelligence

2. Klym H., Diachok R. Dynamic search for errors in industrial internet protocols for application in multisensory control systems // Computer systems and information technologies, 2022, No 3, p. 65-74. https://csitjournal.khmnu.edu.ua/index.php/csit/article/view/172/105

3. Diachok R., Klym H. Modified fog-based trust method of data monitoring for multi-sensor configuration systems // Measuring Equipment and Metrology, 2022, 83(4), 47-55. https://science.lpnu.ua/uk/istcmtm/vsi-vypusky/volume-83-no4-2022/modified-fog-based-trust-method-data-monitoring-multi-sensor

4. Diachok R., Klym H. Current state of development of intelligent information and measuring systems for environmental monitoring with multisensor configuration // Visnyk of Kherson National Technical University, 2022, No 2(81) 55-69. http://kntu.net.ua/index.php/eng/content/view/full/85326

5. Трач І.Б., Клим Г.І., Дячок Р.В., Карбовник І.Д. Проєктування мікропроцесорних пристроїв для визначення напрямку до джерела звуку // Військово-технічний збірник. – Випуск, 2022, No 27, с. 35-45. http://vtz.asv.gov.ua/article/view/268041

6. Diachok R., Trach I., Klym H., Karbovnyk I., Dunets R. Hardware and software complex of intellectualized ornithopter-type UAV for military applications // Electronics and information technologies, 2018, Issue 10, p. 31-40. http://elit.lnu.edu.ua/pdf/10_3.pdf

7. Diachok R., Klym H. Monitoring trust status during Fog level data analysis of the sensor network // Proceedings of the 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2022, p. 1-6. https://ieeexplore.ieee.org/abstract/document/10018674

8. Diachok R., Klym H., Vasylchyshyn I., Karbovnyk I. Definition system of human body position in virtual reality // Proceedings of the16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2022, p. 358-361. https://ieeexplore.ieee.org/abstract/document/9766850

9. Klym H., Dunets R., Horbatyi I., Diachok R. Security subsystem and smart home management system // Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 2018, p. 194-197. https://ieeexplore.ieee.org/abstract/document/8409126

10. Diachok R., Dunets R., Klym H. System of detection and scanning bar codes from raspberry Pi web camera // Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT'2018), 2018, p. 184-187. https://ieeexplore.ieee.org/abstract/document/8409124

11. Diachok R., Klym H., Vasylchyshyn I. Real-time mobile-based platform for determining level and location of radiation background. Proceedings of the 22nd International Conference on Computational Problems of Electrical Engineering (CPEE), 2021, p. 1-4. https://ieeexplore.ieee.org/abstract/document/9585271

12. Дячок Р. В., Клим Г. І. Метод очищення мережевих даних на базі технології інтелекту // Всеукраїнська науково-практична конференція молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості», 13 жовтня 2022, Івано-Франківськ, Україна, С. 209-210.

ДОДАТОК Б. Програмний код реалізації методу очищення мережевих даних

```
import numpy as np
class Canopy: def __init__(self, T1, T2, distance_metric='euclidean',
        threshold=None, medoids=None): self.T1 = T1
    self.T2 = T2 self.distance_metric = distance_metric
    self.threshold = threshold self.medoids = medoids
      def fit(self, X):
    self.X = X self.N = X.shape[0]
        # Створюємо порожній список для кластерів
    self.clusters = []
    # Якщо не задано медоїди, то вибираємо їх випадково if self.medoids is None:
      self.medoids = np.random.choice(self.N, size=int(self.N*0.1), replace=False)
    # Створюємо перший кластер з першого медоїда first_medoid = self.medoids[0]
    self.clusters.append([first_medoid])
    # Ітерація по медоїдам for i in range(1, len(self.medoids)):
      medoid = self.medoids[i]
      # Перевіряємо відстань між поточним медоїдом та всіма іншими медоїдами
distances = self._calculate_distance(medoid, self.medoids[:i])
          # Якщо всі відстані більше T2, то створюємо новий кластер
      if np.all(distances > self.T2): self.clusters.append([medoid])
      else: # Знаходимо кластери, до яких належать інші медоїди
        memberships = [] for j in range(i):
          if distances[j] <= self.T1: memberships.append(j)
              # Якщо немає жодного кластера, до якого належать інші медоїди,
        # то створюємо новий кластер if len(memberships) == 0:
          self.clusters.append([medoid]) else:
          # Додаємо поточний медоїд до всіх кластерів, до яких належать інші
медоїди for membership in memberships:
            self.clusters[membership].append(m medoid)
        # Знаходимо кластери, до яких належать точки, що не є медоїдами
non_medoid_indices = np.setdiff1d(np.arange(self.N), self.medoids)
```

```
            non_medoid_memberships = [] for j in range(len(self.clusters)):
                cluster_indices = np.array(self.clusters[j]) distances =
self._calculate_distance(medoid, non_medoid_indices[cluster_indices])
                if np.any(distances <= self.T1): non_medoid_memberships.append(j)
                        # Якщо немає жодного кластера, до якого належать точки, що не є
медоїдами,
            # то додаємо новий кластер if len(non_medoid_memberships) == 0:
                self.clusters.append([medoid]) else:
                # Додаємо поточний медоїд до всіх кластерів, до яких належать точки, що
не є медоїдами for membership in non_medoid_memberships:
                    self.clusters[membership].append(medoid)
    # Якщо порігове значення не задано, то вибираємо його як середнє значення
відстаней в кластерах if self.threshold is None:
        distances = [] for i in range(len(self.clusters)):
            indices = np.array(self.clusters[i]) if len(indices) > 1:
                d = self._calculate_distance_matrix(indices, indices) d = d[np.triu_indices_from(d,
k=1)]
                distances.extend(d) self.threshold = np.mean(distances)
        # Фінальна кластеризація
    self.labels_ = -1 * np.ones(self.N, dtype=int) for i in range(len(self.clusters)):
        indices = np.array(self.clusters[i]) if len(indices) > 1:
            d = self._calculate_distance_matrix(indices, indices) within_cluster_distances =
d[np.triu_indices_from(d, k=1)]
            medoid_index = np.argmin(np.mean(d, axis=1)) medoid = indices[medoid_index]
            self.labels_[indices] = i else:
            medoid = indices[0] self.labels_[medoid] = i
                return self.labels_
def predict(self, X):
    # Якщо модель не навчена, то повертаємо помилку if not hasattr(self, 'labels_'):
        raise ValueError('The model has not been trained yet.')
    # Розраховуємо відстані між кожною точкою X та кожним медоїдом distances =
```

```
self._calculate_distance_matrix(X, self.X[self.medoids])
        # Знаходимо ближчий медоїд до кожної точки X
    medoid_indices = np.argmin(distances, axis=1)
    medoids = self.X[self.medoids][medoid_indices]
# Знаходимо відстані між кожною точкою X та кожним медоїдом distances =
self._calculate_distance(X, medoids)
        # Призначаємо кожну точку до відповідного кластера
    labels = -1 * np.ones(X.shape[0], dtype=int) for i in range(len(self.clusters)):
        indices = np.array(self.clusters[i]) d = distances[:, i]
        mask = d <= self.T1 labels[mask] = i
        # Якщо є точки, які не були призначені до жодного кластера, то призначаємо їх
до найближчого кластера
    if np.any(labels == -1): nearest_cluster_indices = np.argmin(distances, axis=1)
        labels[labels == -1] = nearest_cluster_indices[labels == -1]
    return labels
def _calculate_distance(self, a, b): # Розраховує відстань між масивом a та масивом b
    return np.linalg.norm(a[:, np.newaxis] - b[np.newaxis, :], axis=2)
def _calculate_distance_matrix(self, a, b): # Розраховує матрицю відстаней між масивом
a та масивом b
    return np.linalg.norm(a[:, np.newaxis] - b[np.newaxis, :], axis=2)
```

ДОДАТОК В. Програмний код реалізації методу динамічного пошуку помилок

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>

#define SERVER_IP "192.168.1.100" // IP-адреса сервера
#define SERVER_PORT 502 // Порт сервера
#define BUF_SIZE 1024 // Розмір буфера

int main(int argc, char **argv) {
    int sockfd;
    struct sockaddr_in servaddr;
    char buf[BUF_SIZE];
    int n;

    // Створюємо сокет
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("socket error");
        exit(1);
    }

    // Заповнюємо адресу сервера
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(SERVER_PORT);
    if (inet_pton(AF_INET, SERVER_IP, &servaddr.sin_addr) <= 0) {
```

```
        perror("inet_pton error");
        exit(1);
    }

    // Підключаємося до сервера
    if (connect(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr)) < 0) {
        perror("connect error");
        exit(1);
    }

/ Відправляє запит до сервера Modbus TCP і повертає відповідь
int modbus_tcp_request(int sockfd, const uint8_t* req_data, size_t req_len, uint8_t*
resp_data, size_t resp_len) {
    // Створюємо буфер для відправки запиту
    uint8_t send_buffer[MAX_TCP_PACKET_SIZE];
    uint8_t* send_ptr = send_buffer;

    // Створюємо заголовок для TCP пакета
    struct tcp_header tcp_hdr;
    tcp_hdr.src_port = htons(rand() % 65536);
    tcp_hdr.dest_port = htons(MODBUS_TCP_PORT);
    tcp_hdr.seq_num = htonl(rand() % 4294967296);
    tcp_hdr.ack_num = 0;
    tcp_hdr.data_offset = 5;
    tcp_hdr.flags = TCP_FLAG_SYN;
    tcp_hdr.window_size = htons(2048);
    tcp_hdr.checksum = 0;
    tcp_hdr.urgent_ptr = 0;

    // Записуємо заголовок до буферу
    memcpy(send_ptr, &tcp_hdr, sizeof(tcp_hdr));
```

```
send_ptr += sizeof(tcp_hdr);

// Записуємо запит Modbus TCP до буферу
memcpy(send_ptr, req_data, req_len);
send_ptr += req_len;

// Відправляємо пакет на сервер
if (send(sockfd, send_buffer, send_ptr - send_buffer, 0) < 0) {
    perror("send");
    return -1;
}

// Очікуємо відповідь від сервера
uint8_t recv_buffer[MAX_TCP_PACKET_SIZE];
int recv_len = recv(sockfd, recv_buffer, MAX_TCP_PACKET_SIZE, 0);
if (recv_len < 0) {
    perror("recv");
    return -1;
}

// Перевіряємо, чи відповідь є дійсним запитом Modbus TCP
if (recv_len < sizeof(tcp_hdr) + 2) {
    fprintf(stderr, "Invalid Modbus TCP response length\n");
    return -1;
}
const uint8_t* resp_ptr = recv_buffer + sizeof(tcp_hdr);
if (resp_ptr[1] != req_data[1]) {
    fprintf(stderr, "Invalid Modbus TCP response function code\n");
    return -1;
}
if (resp_ptr[0] != (req_data[0] | 0x80)) {
```

```
        fprintf(stderr, "Invalid Modbus TCP response slave address\n");
        return -1;
    }

  // Безкінечний цикл для генерації запитів до сервера
    while (1) {
        // Генеруємо випадковий Modbus функціональний код
        uint8_t function_code = rand() % 128 + 1;

        // Генеруємо випадковий Modbus адрес регістра
        uint16_t register_address = rand() % 65536;

        // Генеруємо випадковий Modbus кількість регістрів
        uint16_t register_count = rand() % 127 + 1;
    // Закриваємо сокет
    close(sockfd);

    return 0;
}
```

ДОДАТОК Г. Програмний код модифікованого методу істинності моніторингу даних при розподілі ресурсів на основі туманних обчислень

```
% Кількість кластерівnum_clusters = 8;
% Кількість вузлів у кожному кластері
num_nodes = 300;
% Максимальний рівень кластераmax_level = 4;
% Створення порожньої матриці для зберігання вузлів кожного кластера
cluster_nodes = cell(num_clusters, max_level);
% Генерація випадкових кластерних структурfor i = 1:num_clusters
   % Додавання вузлів зовнішнього шару кластера for j = 1:num_nodes
      cluster_nodes{i,1} = [cluster_nodes{i,1}, j]; end
      % Генерація внутрішніх рівнів кластера
   for level = 2:max_level % Кількість вузлів на поточному рівні
      level_nodes = floor(num_nodes/(2^(level-1)));
      % Додавання вузлів на поточний рівень for j = 1:level_nodes
         cluster_nodes{i,level} = [cluster_nodes{i,level}, num_nodes+(j-
1)*2+1:num_nodes+j*2]; end
            % Оновлення загальної кількості вузлів
      num_nodes = num_nodes + level_nodes*2; end
end
% Виведення структури кожного кластерafor i = 1:num_clusters
   fprintf('Cluster %d:\n', i); for level = 1:max_level
      fprintf('Level %d: %s\n', level, mat2str(cluster_nodes{i,level})); end
   fprintf('\n');end
```

ДОДАТОК Д. Акти впровадження

“ЗАТВЕРДЖУЮ”
Проректор з науково-педагогічної роботи
Національного університету
“Львівська політехніка”
_____ О.Р. Давидчак
_____ 2023 р.

А К Т
про впровадження у навчальний процес результатів дисертації
Дячка Романа Васильовича

Цей акт складено про те, що результати дисертації Дячка Романа Васильовича впроваджено у навчальний процес кафедри «Спеціалізовані комп’ютерні системи» Національного університету «Львівська політехніка».

Впровадження результатів дисертації полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт.

Зокрема для викладання дисципліни «Дослідження та проектування спеціалізованих програмних систем» для студентів освітньо-кваліфікаційного рівня «магістр», що навчаються за спеціальністю 123 «Комп’ютерна інженерія (освітня програма «Спеціалізовані комп’ютерні системи»), використано такі результати:

- загальні принципи побудови спеціалізованих систем з мультисенсорною конфігурацією;
- архітектурні рішення для побудови сенсорних систем.

У лекційному курсі «Дослідження та проектування спеціалізованих програмних систем» для студентів освітньо-кваліфікаційного рівня «магістр», що навчаються за спеціальністю 123 «Комп’ютерна інженерія (освітня програма «Спеціалізовані комп’ютерні системи»), використано такі результати:

- метод очищення даних в сенсорних мережах;
- метод моніторингу даних на основі туманних обчислень.

Завідувач кафедри СКС
д.т.н., професор — Роман КОЧАН

Професор кафедри СКС
д.т.н., професор — Роман ДУНЕЦЬ

Професор кафедри СКС
д.т.н., професор — Галина КЛИМ

"ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Національного університету
"Львівська політехніка"
_____ І.В. Демидов
_____ 2023 р.

А К Т
використання наукових результатів
дисертації Дячка Романа Васильовича,
представленої на здобуття наукового ступеня доктора філософії

Комісія у складі: голови комісії – начальника науково-дослідної частини д.т.н., с.н.с. Небесного Р.В. та членів комісії - завідувача кафедри СКС Кочана Р.В., професора кафедри СКС Клим Г.І., професора кафедри СКС Колодія З.О., доцента кафедри СКС Гонсьор О.Й. цим актом підтверджують, що результати дисертації Дячка Р.В., зокрема:

- метод очищення даних в бездротових сенсорних мережах;
- метод моніторингу даних при розподілі ресурсів на основі туманних обчислень;
- метод нечітких тестів у поєднанні з динамічними мультимодальною передачею даних

використано у науково-дослідній роботі, фінансованій Міністерством освіти і науки України, що виконується на кафедрі СКС і включено до звіту: "Оптимізовані нанокомпозити та сенсорні структури для оборонних систем контролю безпеки та виявлення загроз" (№ держ. реєстру 0122U000807)

Одержані автором результати використано:

- при розробленні засобів інтелектуальних мультисенсорних систем;
- при розробленні моделей передачі даних у сенсорних мережах;
- при проектуванні систем контролю безпеки.

Голова комісії:

начальник
науково-дослідної частини
д.т.н., с.н.с. Роман НЕБЕСНИЙ

Члени комісії:

Завідувач кафедри СКС Роман КОЧАН

професор кафедри СКС Галина КЛИМ

професор кафедри СКС Зеновій КОЛОДІЙ

доцент кафедри СКС Оксана ГОНСЬОР

Національна академія наук України
ДЕРЖАВНЕ ПІДПРИЄМСТВО
НАУКОВО-ТЕЛЕКОМУНІКАЦІЙНИЙ ЦЕНТР
"УКРАЇНСЬКА АКАДЕМІЧНА І ДОСЛІДНИЦЬКА МЕРЕЖА"
ІНСТИТУТУ ФІЗИКИ КОНДЕНСОВАНИХ СИСТЕМ НАН УКРАЇНИ

79011 Львів, вул. Свєнціцького, 1
тел./факс: +380 322 768405, ел. пошта: uarnet@uar.net

ЗАТВЕРДЖУЮ
Директор Державного підприємства
Науково-телекомунікаційний центр
«Українська академічна і дослідницька мережа»
ІФКС НАН України

к.ф.-м.н. І.А. Процикевич

АКТ

про впровадження результатів дисертації аспіранта
кафедри «Спеціалізовані комп'ютерні системи»
Національного університету «Львівська політехніка»
Дячка Романа Васильовича

Цим актом підтверджується, що основні положення дисертації Дячка Р.В. використовуються у виробничому процесі Державного підприємства Науково-телекомунікаційний центр «Українська академічна і дослідницька мережа» Інституту фізики конденсованих систем НАН України, зокрема:

- метод динамічного пошуку помилок у промисловій системі інтернету;
- метод очищення даних у бездротовій сенсорній мережі на базі технологій інтелекту;
- моніторинг правдивості даних мережі під час розподілу ресурсів на рівні туману.

Професіонал з інноваційної діяльності **О.З. Гірняк**