Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи та модель підвищення надійності інфраструктури інтернету речей

(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-5
Максим ДЯЧЕНКО
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Алєксандр ТОРБА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Дяченку Максиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та модель підвищення надійності інфраструктури інтернету речей

затверджена наказом по університету від "21" квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи 1. Референсна архітектура IoT.
2. Поняття надійності розглядається комплексно.
3. Використовувані методи: стохастичні мережі Петрі; неперервні марковські процеси; графові моделі.

4. Перелік питань, що потрібно опрацювати у роботі
1. Аналіз проблеми надійності інфраструктури інтернету речей
2. Методи підвищення надійності інфраструктури IoT
3. Розробка моделі підвищення надійності інфраструктури IoT
4. Експериментальне дослідження
5. Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація – 15 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми надійності інфраструктури	22.04.25-29.04.25	
	інтернету речей		
2	Методи підвищення надійності інфраструктури	30.04.25-07.05.25	
	IoT		
3	Розробка моделі підвищення надійності	08.05.25-21.05.25	
4	Експериментальне дослідження	22.05.25-02.06.25	
5	Оформлення матеріалів кваліфікаційної роботи	03.06.25-05.06.25	
6	Подання кваліфікаційної роботи керівникові та	06.06.25-09.06.25	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	10.06.25-12.06.25	
	вання		

Дата видачі завдання " 21 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Алєксандр ТОРБА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 74 с., 8 рис., 4 табл., 1 дод., 32 джерела.

АНОМАЛІЇ, ІНТЕРНЕТ РЕЧЕЙ, ІНФОРМАЦІЙНА БЕЗПЕКА, ІНФРАСТРУКТУРА, МОНІТОРИНГ, НАДІЙНІСТЬ.

Мета роботи – розробка та дослідження методів і моделі підвищення надійності інфраструктури IoT шляхом інтеграції засобів безпеки, резервування, самовідновлення та інтелектуального моніторингу. Об’єкт дослідження – інфраструктура Інтернету речей як складна розподілена система. Предмет дослідження – методи та моделі підвищення надійності функціонування IoT-інфраструктури.

Наукова новизна дослідження полягає в: інтеграції методів криптографічного захисту, відмовостійкості, прогнозної діагностики та самоорганізації в єдину модель підвищення надійності IoT-інфраструктури; формалізації моделі на основі мереж Петрі та Марковських процесів; використанні інтелектуальних алгоритмів виявлення аномалій в контексті відмов компонентів.

Практичне значення роботи полягає в можливості застосування результатів для: розробки надійних IoT-рішень у критично важливих галузях; автоматичного аналізу стану компонентів IoT-систем та попередження відмов; створення адаптивних інфраструктур на основі самовідновлюваних IoT-мереж.

ABSTRACT

Master’s thesis: 74 pages, 8 figures, 4 tables, 1 appendix, 32 sources.

ANOMALIES, INFORMATION SECURITY, INFRASTRUCTURE, INTERNET OF THINGS, MONITORING, RELIABILITY.

The objective of this study is the development and investigation of methods and a model for enhancing the reliability of IoT infrastructure through the integration of security mechanisms, redundancy, self-healing, and intelligent monitoring. The object of the research is the Internet of Things infrastructure as a complex distributed system. The subject of the research is the methods and models for improving the reliability of IoT infrastructure operation.

The scientific novelty of the research lies in the integration of cryptographic protection methods, fault tolerance, predictive diagnostics, and self-organization into a unified model for enhancing the reliability of IoT infrastructure; formalization of the model based on Petri nets and Markov processes; and the application of intelligent anomaly detection algorithms in the context of component failures.

The practical significance of the work consists in the possibility of applying the results for the development of reliable IoT solutions in critical sectors; automated analysis of the state of IoT system components and failure prevention; and the creation of adaptive infrastructures based on self-healing IoT networks.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – штучний інтелект (англ., Artificial Intelligence)

API – програмний інтерфейс застосування (англ., Application Programming Interface)

CTMC – неперервні марковські процеси (англ., Continuous-Time Markov Chain)

DL – глибоке навчання (англ., Deep Learning)

DT – цифровий двійник (англ., Digital Twin)

FL – федеративне навчання (англ., Federated Learning)

FTA – дерева відмов (англ., Fault Tree Analysis)

IoT – інтернет речей (англ., Internet of Things)

LSTM – довготривала короткочасна пам’ять – тип рекурентної нейронної мережі (англ., Long Short-Term Memory)

ML – машинне навчання (англ., Machine Learning)

MQTT – протокол обміну повідомленнями для IoT (англ., Message Queuing Telemetry Transport; реалізація Mosquitto)

MTTF – середній час безвідмовної роботи системи до першої відмови (англ., Mean Time to Failure)

MTTR – середній час, необхідний для відновлення після відмови (англ., Mean Time to Repair)

RBD – діаграма надійності (англ., Reliability Block Diagram)

R-SHIELD – концептуальна модель стійкої самовідновлюваної інтелектуальної багаторівневої цифрової архітектури (англ., Resilient Self-Healing Intelligent Layered Digital architecture)

RUL – залишковий ресурс роботи/час до відмови (англ., Remaining Useful Life)

SPN – стохастичні мережі Петрі (англ., Stochastic Petri Nets)

ВСТУП

Інтернет речей (Internet of Things, IoT) є однією з найдинамічніших і найперспективніших сфер сучасної інформаційної технології. Завдяки широкому впровадженню IoT-технологій у промисловості, транспорті, охороні здоров’я, енергетиці та побуті, значно зростають вимоги до надійності функціонування IoT-інфраструктури. Збої, викликані апаратними відмовами, програмними помилками або кіберзагрозами, можуть призвести до серйозних наслідків, зокрема зупинки виробничих процесів, втрати даних або загрози для життя.

Сучасні дослідження [1–3] засвідчують, що традиційні підходи до забезпечення надійності IT-систем не повністю адаптовані до розподіленої, гетерогенної та динамічної природи IoT. Потреба в нових методах підвищення надійності та створенні адаптивних моделей, здатних реагувати на зовнішні загрози й внутрішні несправності, є нагальною вимогою як теоретичного, так і прикладного рівня.

Метою даної роботи є розробка та дослідження методів і моделі підвищення надійності інфраструктури IoT шляхом інтеграції засобів безпеки, резервування, самовідновлення та інтелектуального моніторингу.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати сучасний стан досліджень проблеми надійності IoT-систем;
- систематизувати методи підвищення надійності з урахуванням особливостей IoT-інфраструктур;
- запропонувати модель підвищення надійності на основі формальних математичних підходів;
- реалізувати експериментальне дослідження моделі та оцінити ефективність запропонованих методів;
- розробити практичні рекомендації щодо впровадження результатів у

прикладних IoT-сценаріях.

У роботі використано такі наукові методи:

- аналіз і синтез інформації з наукових джерел для побудови узагальненої моделі;
- математичне моделювання з використанням теорії графів, ймовірнісних моделей, Марковських процесів;
- методи формальної верифікації для перевірки правильності функціонування компонентів моделі;
- методи комп'ютерного експерименту для дослідження відмовостійкості;
- порівняльний аналіз ефективності рішень за допомогою емпіричних даних.

Робота складається зі вступу, чотирьох розділів, загальних висновків, переліку джерел посилання та додатків.

1 АНАЛІЗ ПРОБЛЕМИ НАДІЙНОСТІ ІНФРАСТРУКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Концептуальні основи Інтернету речей та його інфраструктури

Інтернет речей (IoT) – це парадигма, яка описує мережу фізичних об'єктів, здатних збирати, обробляти, передавати дані та взаємодіяти між собою та з іншими системами через інтернет. Основна мета IoT – створення інтелектуального середовища, де «розумні речі» працюють узгоджено з мінімальним втручанням людини.

За прогнозами, до 2030 року в експлуатації буде понад 25 мільярдів IoT-пристроїв [4]. Це вимагає високої надійності інфраструктури IoT, оскільки збої в мережі можуть призвести до значних втрат – від фінансових до загроз життю.

Основні характеристики IoT:

- гетерогенність (різні пристрої, протоколи, середовища);
- розподіленість (відсутність єдиного центру обробки);
- масштабованість (здатність підтримувати мільйони з’єднань);
- динамічність (зміна топології в режимі реального часу);
- обмежені ресурси (обчислювальні, енергетичні).

Типова інфраструктура IoT включає такі компоненти:

- IoT-пристрої (sensors/actuators) – первинні елементи, що збирають/реагують на інформацію;

- комунікаційні мережі – протоколи та канали передачі (LoRaWAN, Zigbee, Wi-Fi, 5G тощо);

- шлюзи (Gateways) – пристрої, які поєднують локальні мережі з хмарними чи серверними рішеннями;

- обчислювальні платформи – локальні або хмарні середовища для обробки даних (Edge, Fog, Cloud);

- аналітичні сервіси – системи виявлення закономірностей, прогнозування, автоматичного реагування;

- засоби керування та інтерфейси користувача – для моніторингу, контролю, оновлень та візуалізації.

На схемі інфраструктури IoT, що наведена на рисунку 1.1, зображені рівні та потоки даних.

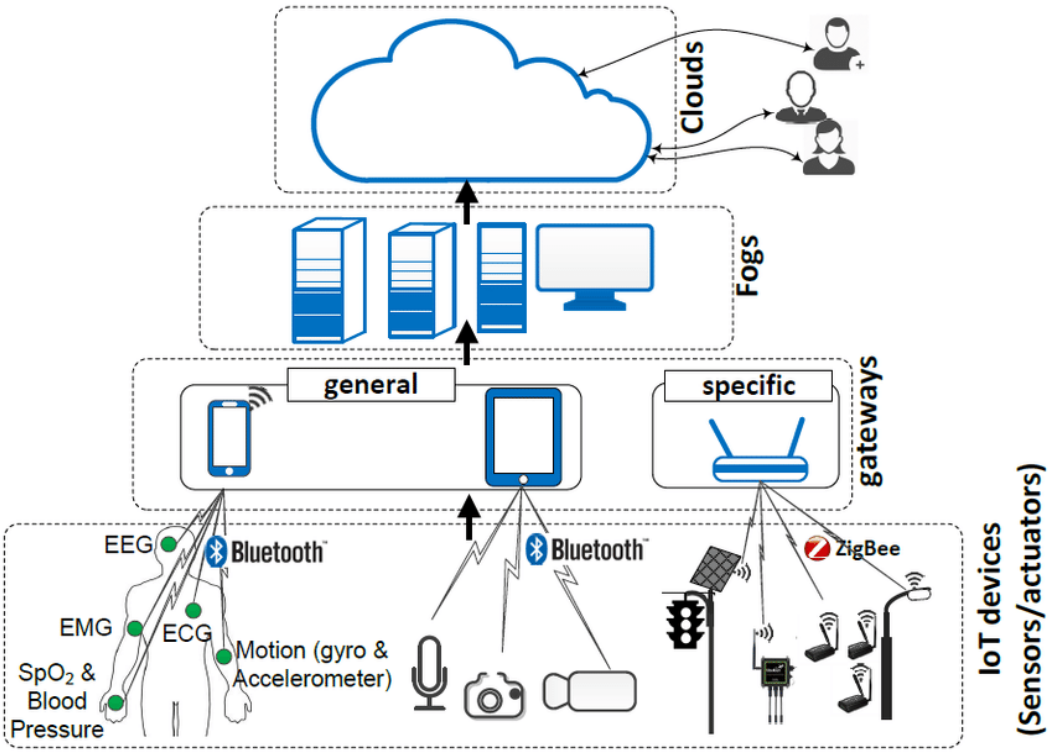

Рисунок 1.1 – Схема інфраструктури IoT

1.2 Основні характеристики та вимоги до надійності IoT-систем

Інфраструктура Інтернету речей, на відміну від класичних IT-систем, характеризується складною розподіленою структурою, динамічною топологією, гетерогенністю компонентів та обмеженими ресурсами. Ці властивості значно ускладнюють забезпечення високої надійності системи в цілому. Ключові характеристики IoT, що впливають на надійність, розглянуто нижче.

Високий рівень розподіленості. Пристрої розміщуються у віддалених

або важкодоступних місцях, часто поза прямим контролем, що знижує можливість ручного втручання у випадку відмови [4].

Різнорідність пристроїв. IoT-пристрої різняться за архітектурою, операційними системами, енергетичними профілями та протоколами зв'язку, що створює труднощі для уніфікованого управління надійністю [5].

Динамічність топології. Пристрої можуть підключатися/відключатися динамічно, а комунікаційні канали можуть змінюватися залежно від умов середовища (наприклад, у мобільних або сенсорних мережах) [6].

Обмежені ресурси. Більшість IoT-вузлів має обмеження по енергії, обчислювальній потужності та пам'яті, що унеможливлює використання традиційних механізмів відмовостійкості, резервного копіювання та складних алгоритмів [7].

Надійність (Reliability) – це здатність системи виконувати задані функції протягом визначеного періоду часу без збоїв. У контексті IoT поняття надійності включає такі аспекти:

- доступність (Availability): здатність IoT-сервісів бути доступними користувачам у визначений момент часу.
- відмовостійкість (Fault Tolerance): здатність системи функціонувати навіть за наявності збоїв окремих компонентів.
- безперервність роботи (Continuity): забезпечення незмінності обслуговування критичних процесів.
- відновлення (Recoverability): здатність повернутися до стабільного стану після збоїв.

Такі метрики часто використовуються в моделюванні надійності за допомогою Марковських моделей, мереж Петрі або аналізу дерев відмов (FTA) [8].

Типові загрози надійності в IoT:

- втрата підключення;
- апаратні збої;
- програмні помилки;

- кібератаки (відмова в обслуговуванні, підміна трафіку тощо);
- перевантаження каналів зв’язку або хмарних сервісів;
- енергетичне виснаження пристроїв.

Таблиця 1.1 – Формалізовані вимоги до надійності IoT-систем

№	Вимога	Опис
1	MTTF (Mean Time to Failure)	Середній час безвідмовної роботи системи до першої відмови
2	MTTR (Mean Time to Repair)	Середній час, необхідний для відновлення після відмови
3	Availability (A = MTTF / (MTTF+MTTR))	Частка часу, у яку система працює коректно
4	Redundancy	Рівень дублювання апаратних чи програмних ресурсів
5	Resilience	Здатність до адаптації та відновлення при зовнішніх впливах

Вирішення цих проблем вимагає цілісного підходу, що включає багаторівневе резервування, самоорганізацію, динамічний моніторинг та автоматизоване виявлення відмов.

1.3 Аналіз існуючих моделей оцінювання надійності IoT-систем

За 2020-2024 рр. було опубліковано низку ключових досліджень, що висвітлюють модельні підходи для забезпечення надійності IoT – від формалізованих моделей до інтелектуальних гібридних практик.

Формальні моделі на основі SPN і Марковських процесів [8]. Стохастичні мережі Петрі (SPN) стали одним із ключових інструментів для формального моделювання надійності. Такі мережі дозволяють відобразити випадкові затримки і стан компонентів через марковську модель.

В роботі [9] запропонувано SPN модель для оцінки доступності систем IoMT (Internet of Medical Things), що показало можливість аналітичної оцінки MTTF та ймовірності простою.

В роботі [10] використали часові кольорові GSPN і симуляції Монте Карло для аналізу надійності мехатронних систем із урахуванням відмов комплектуючих.

Аметалістичні підходи з нечіткими Petri мережами застосовуються для моделювання трафіку в динамічних IoT мережах, прагнучи надати гнучкість, необхідну для умов незавершеного або нечіткого середовища. Такі підходи дозволяють точно кількісно оцінювати важливі метрики: MTTF, MTTR, Availability, і проводити аналіз критичних точок відмов. Також реалізовується карта станів як марковський процес, що забезпечує подальший аналіз за допомогою методів теорії ймовірностей.

Інтеграція моделей довіри і репутації. Модель довіри (trust) стає важливою у контексті IoT, щоб оцінювати коректність роботи пристроїв:

У дослідженні [11] автори систематизували традиційні та AI базовані моделі довіри, визначивши ключові метрики: масштабованість, ефективність, затримки. Автори відзначають, що традиційні моделі довіри не враховують гетерогенного, динамічного характеру IoT, тому потрібні адаптовані гібридні рішення, що поєднують математичні та ML методи .

У результаті інтеграція довіри в загальні моделі надійності дозволяє врахувати поведінкові аномалії, корупційні атаки тощо.

Цифрові двійники і Reliability Blocks Diagram (RBD). У промислових та IoT системах стає популярним поєднання RBD із SPN і цифровими двійниками, що дозволяє створювати динамічні адаптивні моделі для аналізу відмов та планування технічного обслуговування. IIC [30] рекомендує архітектурні підходи для цифрових двійників, що включають концептуальні моделі й послуги для забезпечення інтероперабельності, надійності, довіри та безпеки. Сервіс AWS IoT TwinMaker, представлений компанією Amazon у 2022 р. забезпечує практичне середовище для цифрових двійників –

інтеграція реальних даних та знань допомагає моделювати й виявляти збої в реальному режимі (reliabilityweb.com).

Формальні мережі високого рівня. В роботі [31] запропоновано використання високорівневих HLPN для моделювання політик і виявлення конфліктів у багатосервісних IoT оточеннях (наприклад, «розумний дім») – це важливо для збереження QoS і надійності.

В таблиці 1.1 порівнено існуючі моделі оцінювання надійності IoT-систем та узагальнено ключові аспекти, з урахуванням їхньої формалізації, гнучкості, підтримки змінних середовищ і можливості інтеграції сучасних технологій (довіри, цифрових двійників тощо).

Таким чином, SPN і GSPN – основа формальної оцінки надійності, які забезпечують точну кількісну аналітику. Інтеграція довіри розширює спектр ймовірнісних моделей, додаючи поведінковий вимір. Цифрові двійники і RBD дозволяють реалізувати інтерактивні симуляційні рішення для управління надійністю. HLPN для виявлення конфліктів важливі в комплексних сценаріях, зокрема IoT середовищах. Для створення ефективної моделі необхідне поєднання: SPN + довіра + цифрові двійники + нечіткі політики, що буде реалізовано в розділі 3.

1.4 Огляд сучасних методів та підходів підвищення надійності IoT інфраструктури

Забезпечення надійності в інфраструктурі Інтернету речей вимагає комплексного підходу, який охоплює кілька рівнів системи: від апаратного до рівня застосунків. У 2022-2024 рр. спостерігається тенденція до інтеграції методів прогнозної діагностики, цифрових двійників, самоорганізації та інтелектуального управління ресурсами. Нижче розглянуто сучасні напрямки та інструменти підвищення надійності.

Таблиця 1.1 – Порівняння моделей оцінювання надійності IoT-систем

№	Модель / Метод	Ключові характеристики	Переваги	Обмеження	Приклади використання
1	Марковські моделі	Станова модель з імовірнісними переходами	Чітка математична база; аналіз MTTF/MTTR	Не масштабуються для великих мереж	[8, 9]
2	Стохастичні мережі Петрі (SPN)	Подання станів і подій у вигляді позицій і транзицій	Підтримка паралелізму; детальний аналіз	Потребує складного моделювання	[10, 12]
3	Нечіткі мережі Петрі	Враховують невизначеність у параметрах системи	Підходить для адаптивного аналізу	Низька стандартизація; обмежена кількість інструментів	[16]
4	Reliability Block Diagram (RBD)	Блокова модель логічного з’єднання компонентів системи	Простота реалізації; візуалізація	Лише структурна модель; без урахування поведінкових змін	[30]; AWS IoT TwinMaker
5	Цифрові двійники (Digital Twins)	Віртуальне віддзеркалення фізичних пристроїв IoT у реальному часі	Динамічна візуалізація; сценарії "що, якщо"	Висока складність впровадження; потреба в реальних даних	[30]; AWS IoT TwinMaker
6	Моделі довіри та репутації	Обчислення показників на основі поведінки пристроїв і взаємодій	Врахування поведінкових характеристик; AI-адаптивність	Потребують даних для навчання; високий рівень обчислень	[11]
7	High-level Petri Nets (HLPN)	Моделювання конфліктів, правил і сценаріїв високого рівня у складних середовищах	Гнучкість у політиках; можливість формалізації сервісної логіки	Потребують спеціалізованих засобів розробки	-

Резервування та відмовостійкість. Методи резервування передбачають створення дублюючих або запасних компонентів для забезпечення роботи IoT-систем у разі відмов.

Залежно від рівня реалізації, розрізняють:

- апаратне резервування – встановлення дублюючих сенсорів, контролерів чи комунікаційних каналів. У системах «розумного виробництва» це стандартна практика;

- програмне резервування – дублювання віртуальних вузлів або контейнерів, що виконують критичні функції;

- гібридні стратегії – поєднання фізичних і віртуальних елементів резервування.

Наприклад, у моделі DRF-IoT (Dynamic Redundancy Framework for IoT) запропоновано адаптивний підхід до резервування, що враховує рівень навантаження, стан компонентів та історію відмов [12]. У цій моделі визначається оптимальна кількість дублюючих компонентів в реальному часі, що дозволяє знижувати перевитрати ресурсів без шкоди для надійності.

Безпека як компонент надійності. Безпека в IoT має прямий вплив на надійність. Компрометація пристрою чи мережі часто призводить до відмови сервісу. Основні напрями:

- безпечні протоколи передачі даних (MQTT, CoAP із TLS/DTLS);
- шифрування і цифрові підписи (AES, ECC, RSA);
- контроль доступу і автентифікація на основі ролей або атрибутів;
- моніторинг вторгнень і аномалій за допомогою IDS/IPS.

Новітні дослідження акцентують на поєднанні Digital Twin (DT) з інструментами штучного інтелекту для виявлення атак і забезпечення цілісності даних. Наприклад, DT, що симулює роботу пристрою, дозволяє порівняти очікувану поведінку з реальною й миттєво виявити відхилення [13].

Моніторинг і прогнозування відмов. Завдяки впровадженню

інтелектуальних систем моніторингу, надійність IoT підвищується за рахунок раннього виявлення потенційних збоїв.

Використовуються:

- Edge/Cloud моніторинг – збирання телеметрії з пристроїв у хмарне середовище;
- Machine Learning/Deep Learning моделі для прогнозу зношування компонентів;
- Time Series Analysis + LSTM – для виявлення відхилень у часових рядах.

У дослідженні [14] описано підхід до прогнозної підтримки в IIoT-системах, який базується на ML-алгоритмах у поєднанні з дельта-резервуванням (delta-checkpointing). Це дозволяє вчасно передбачати збої та зменшити час простою.

Цифрові двійники (Digital Twins). Digital Twin – це віртуальна копія фізичного об'єкта або процесу. Їх використання в IoT дає такі переваги:

- візуалізація стану пристрою в режимі реального часу;
- побудова сценаріїв "що, якщо" (what-if analysis);
- автоматичне тестування змін та оновлень;
- інтеграція з ML/AI для адаптації до змін середовища.

Дослідження [15] демонструє, що цифрові двійники стають ключовою технологією для оцінки надійності в промислових IoT. Поєднання DT і AI дозволяє створювати прогностичні моделі, які виявляють відхилення у поведінці пристрою ще до виникнення відмови.

Самоорганізація та самовідновлення (Self-healing). Ці технології передбачають автоматичне виявлення, локалізацію і компенсацію збоїв у системі.

Основні підходи:

- Federated Learning – навчання на пристроях без передачі сирих даних у центральне сховище;
- автоматичне переключення на резервний маршрут або вузол;

- динамічне оновлення прошивок і конфігурацій.

Зокрема, дослідження [16] описує архітектуру, в якій цифрові двійники IoT-пристроїв реалізують самостійне виявлення аномалій і передають моделі до FL-сервера для оновлення поведінки системи. Це дозволяє системі не лише реагувати на відмови, а й попереджати їх.

Таким чином, сучасні підходи інтегрують безпеку, адаптивне резервування, цифрові двійники та AI. Інтелектуальні прогностичні моделі скорочують час реагування на потенційні збої. Self-healing та FL-архітектури забезпечують еволюцію IoT-систем у напрямку повної автономності.

Всі ці підходи повинні враховувати гетерогенність пристроїв, обмеженість ресурсів та вимоги до масштабованості при побудові моделі, описаній у Розділі 3.

1.5 Висновки до розділу

Проведено ґрунтовний аналіз сучасного стану досліджень проблеми надійності інфраструктури Інтернету речей (IoT). Основні результати можна підсумувати так.

IoT-інфраструктура є надзвичайно складною системою, що поєднує велику кількість гетерогенних пристроїв, протоколів, середовищ та сервісів. Її характерними особливостями є динамічна топологія, розподіленість, обмежені ресурси пристроїв та високі вимоги до масштабованості.

Надійність є критично важливою властивістю IoT-систем, оскільки збої або відмови компонентів можуть призвести до небажаних наслідків, особливо у критично важливих застосуваннях (розумні міста, медицина, промисловість, енергетика).

Моделі оцінювання надійності розвиваються у напрямі використання формалізованих підходів:

- стохастичні мережі Петрі (SPN) і Марковські процеси забезпечують математичну точність;

- нечіткі та ієрархічні моделі дозволяють описувати складні, невизначені та динамічні ситуації;
- моделі довіри та цифрові двійники відкривають нові можливості для інтеграції поведінкових аспектів.

Сучасні підходи до підвищення надійності IoT охоплюють:

- адаптивне резервування (динамічне або апаратно-програмне);
- впровадження прогнозної діагностики та інтелектуального моніторингу;
- застосування цифрових двійників як віртуальних моделей реальних пристроїв для попередження збоїв;
- розробку self-healing систем, що здатні автоматично відновлювати свою функціональність.

Недостатньо дослідженими залишаються питання інтеграції моделей різних рівнів (від сенсорного до хмарного), створення уніфікованих метрик надійності, а також масштабованих підходів для розподіленого контролю та самовідновлення в умовах відмов і атак.

Таким чином, у подальших розділах буде зосереджено увагу на розробці комплексної моделі підвищення надійності IoT-інфраструктури, що поєднує елементи:

- формального моделювання (SPN, Markov Chains),
- динамічного резервування,
- віртуалізації через цифрові двійники,
- та самовідновлення на основі інтелектуального аналізу.

2 МЕТОДИ ПІДВИЩЕННЯ НАДІЙНОСТІ ІНФРАСТРУКТУРИ IOT

2.1 Методи забезпечення безпеки як фактор підвищення надійності IoT-систем

Надійність і безпека в системах Інтернету речей тісно пов’язані. Збій у безпеці, зокрема компрометація пристрою або протоколу, може призвести до зупинки роботи системи, пошкодження або втрати даних. Саме тому інформаційна безпека є критичним чинником надійності IoT-інфраструктури. У цьому підрозділі розглянуто сучасні методи, що підвищують надійність шляхом підвищення безпеки: криптографічні методи, автентифікацію та контроль доступу, а також безпечні протоколи зв’язку.

2.1.1 Криптографічні методи

Криптографія є базовим інструментом для гарантування цілісності, конфіденційності та автентичності переданих даних у IoT-системах. Найбільш використовувані методи:

- AES (Advanced Encryption Standard) – симетричне шифрування з високою швидкістю обробки;
- RSA, ECC (Elliptic Curve Cryptography) – асиметричне шифрування для ключового обміну;
- SHA-256, HMAC – хешування та контроль цілісності даних.

У роботі [17] запропоновано використання гібридної криптографічної системи (ECC + AES), що забезпечує високий рівень захисту при низькому споживанні енергії в IoT-пристроях.

Крім того, у новітніх дослідженнях з'являються легковагові криптографічні алгоритми (Lightweight Cryptography) [18], зокрема SPECK, SIMON, які адаптовано до обмежених ресурсів сенсорних вузлів.

2.1.2 Автентифікація та контроль доступу

Контроль доступу забезпечує захист ресурсів IoT від несанкціонованих дій. Типові підходи:

- RBAC (Role-Based Access Control) – користувачі мають доступ до ресурсів відповідно до ролей;
- ABAC (Attribute-Based Access Control) – гнучке управління доступом на основі атрибутів суб'єкта/об'єкта;
- IoT-Federated Identity Management (FIdM) – підхід до централізованої автентифікації в розподілених IoT-системах.

Сучасні дослідження вказують на ефективність комбінованих систем – наприклад, використання ECC базованої автентифікації з ABAC, як описано в моделі ECP-IoT (Elliptic Curve Protocol for IoT) [19].

Також популярними є біометричні фактори автентифікації (електрокардіограма, голос, шаблони поведінки), особливо в медичних IoT (IoMT).

2.1.3 Безпечні протоколи зв'язку

Передача даних є найбільш вразливим етапом в IoT-середовищі. Найбільш часто використовувані протоколи мають властивості, наведені в таблиці 2.1.

У роботі [20] описано практичне впровадження MQTT із TLS 1.3, що забезпечує мінімальний оверхед і високу захищеність навіть на пристроях з обмеженими обчислювальними ресурсами.

У багаторівневих системах (edge/fog/cloud) рекомендується додаткова перевірка з використанням zero-trust моделей та end-to-end шифрування [21].

Таблиця 2.1 – Безпечні протоколи зв'язку

Протокол	Шар OSI	Характеристики	Захист
MQTT	Application	Легкий, push-based, для обмежених пристроїв	TLS, JWT, ACL
CoAP	Application	REST-подібний, UDP-базований	DTLS, OSCORE
LoRaWAN	MAC	Наддалека передача з низькою швидкістю	AES-128, Network/App Keys
HTTPS	Application	HTTP через TLS	Сертифікати, шифрування

Проводячи аналіз, можна дійти таких висновків.

1. Криптографічні методи, оптимізовані для IoT, забезпечують захист даних без надмірного навантаження на енергоспоживання.

2. Автентифікація та контроль доступу формують «першу лінію оборони» від зовнішніх втручань.

3. Безпечні протоколи зв’язку є невід’ємним компонентом при побудові надійних IoT-мереж, особливо в контексті публічних мереж і промислових середовищ.

4. Інтеграція засобів безпеки з іншими елементами – цифровими двійниками, ML-модулями – дозволяє створювати стійкі до збоїв і атак IoT-системи.

2.2 Методи резервування та відмовостійкості в IoT-мережах

Інфраструктура Інтернету речей є вразливою до широкого спектра відмов: збої сенсорів, втрати зв’язку, збої хмарних сервісів, відключення електроживлення тощо. Тому в контексті IoT особливо важливим є впровадження відмовостійких архітектур, здатних автоматично реагувати на критичні події. Основними підходами є:

- апаратне та програмне резервування;
- балансування навантаження;
- використання віртуалізації та контейнеризації.

Апаратне резервування дозволяє забезпечити мінімальний рівень відмовостійкості в критичних середовищах. Програмне резервування та балансування забезпечують гнучкість, масштабованість та зниження часу простою. Контейнеризація та віртуалізація дозволяють оперативно відновлювати сервіси, розгортати резервні копії без переривання роботи системи. Поєднання резервування з механізмами автоматичного моніторингу та інтелектуального переключення є ключем до надійної IoT-інфраструктури.

2.2.1 Апаратне резервування

Апаратне резервування означає дублювання фізичних компонентів, які виконують критичні функції, наприклад:

- Redundant Sensors and Actuators – сенсори дублюються із автоматичним перемиканням у разі збою основного;
- Failover Gateways – резервні маршрутизатори або шлюзи з автоматичним виявленням і переключенням;
- блоки резервного живлення (UPS, батареї) – для безперервної роботи у випадку відключення електроенергії.

У дослідженні [22] описується відмовостійка архітектура для «розумного будинку», де дублювання пристроїв дозволило знизити втрати даних на 96% при симульованих збоях.

2.2.2 Програмне резервування та балансування навантаження

Програмне резервування полягає в створенні віртуальних екземплярів (запасних сервісів або процесів), які беруть на себе функції у разі збоїв основних. Переваги:

- швидкий запуск резервної копії без потреби фізичного втручання;
- автоматичне масштабування під час пікових навантажень;
- висока гнучкість в адаптації до нових умов.
- балансування навантаження реалізується через механізми:
- Load Balancers (HAProxy, NGINX);
- Service Mesh (наприклад, Istio) – для мікросервісних IoT-архітектур;
- Edge Offloading – розподіл обчислювальних навантажень між периферійними пристроями та хмарою.

У роботі [23] запропоновано механізм "Microservice Self-Replication", що дозволяє віртуальним вузлам у IoT-мережі автоматично створювати свої резервні копії в інших регіонах у випадку загрози або збою.

2.2.3 Віртуалізація та контейнеризація в IoT

Використання віртуалізації дозволяє ізолювати компоненти системи, зменшуючи вплив збоїв одного вузла на інші. Основні технології:

- Virtual Machines (VM) – для ізоляції повноцінних ОС;
- Containers (Docker, Podman) – для легковагового запуску сервісів;
- Orchestration Platforms (Kubernetes, K3s) – для управління контейнерами в кластері IoT вузлів.

Контейнеризація дозволяє забезпечити:

- швидке розгортання резервного компонента;
- автоматичне масштабування залежно від стану ресурсів;
- високу портативність (переміщення між вузлами).

За результатами дослідження [24], використання контейнерів на fog-обчисленнях дозволяє підвищити доступність сервісу до 99,97% у розподілених системах Smart City.

Приклад сценарію відмовостійкої IoT-мережі:

а) вузол А (основний) передає дані;

б) при виявленні відмови спрацьовує Failover-Monitor;

в) контейнер B (резервний) активується автоматично;

г) дані маршрутизуються через балансувальник навантаження;

д) паралельно створюється звіт про відмову в системі моніторингу.

2.3 Моніторинг і діагностика для підвищення надійності

Моніторинг і діагностика є наріжними компонентами забезпечення надійності в інфраструктурі Інтернету речей. У розподілених IoT-системах, що складаються з тисяч пристроїв, вузлів, шлюзів, сервісів і каналів зв'язку, важливою передумовою стабільного функціонування є здатність виявляти, інтерпретувати та прогнозувати небажані стани або тенденції, що передують відмовам.

Наявність обмежених ресурсів (обчислювальних, енергетичних), використання нестабільних каналів зв'язку, відсутність єдиного центра обробки – усе це ускладнює впровадження класичних методів технічного моніторингу. Саме тому в IoT-середовищах виникла потреба в нових методах моніторингу та інтелектуальної діагностики, адаптованих до розподілених, енергоефективних і автономних систем.

На першому етапі ефективного контролю є систематизоване спостереження за станом пристроїв і мереж, що здійснюється шляхом збору телеметричних показників: рівень сигналу, температура, навантаження на ЦП, рівень батареї, якість з'єднання, помилки зчитування, кількість переданих пакетів тощо. Важливою є інтеграція edge-агентів моніторингу, що дозволяють виконувати попередній аналіз без необхідності постійної передачі даних у хмару. Такі рішення дозволяють зменшити затримку реагування, навантаження на мережу та зменшити ризики, пов'язані з конфіденційністю. За результатами проведених досліджень [25], застосування edge-моніторингу скорочує середній час виявлення несправностей майже на 42% у порівнянні з централізованими рішеннями.

Однак лише фіксація критичних станів є недостатньою. В сучасних

умовах IoT-системи потребують інтелектуального виявлення аномалій, тобто виявлення відхилень у поведінці пристроїв або мереж, які можуть бути першими ознаками майбутніх відмов. Для цього застосовуються алгоритми машинного навчання (ML) і глибокого навчання (DL). Наприклад, кластери аномальних спостережень можуть бути визначені за допомогою One-Class SVM або Isolation Forest, а часові ряди телеметрії можуть бути оброблені моделями типу LSTM або CNN-LSTM для прогнозування критичних змін. У статті [26] запропоновано гібридну модель CNN-LSTM, яка показала понад 94% точності детекції несправностей у реальному часі з використанням лише edge-вузлів. Це дозволяє системі функціонувати навіть без постійного підключення до хмари, що особливо важливо в промислових або транспортних сценаріях.

Ще більш перспективним є прогнозування відмов (predictive fault analysis), яке базується не лише на поточних даних, а й на історичних закономірностях. Йдеться про оцінку так званого часу до відмови (RUL – Remaining Useful Life). Цей підхід дає змогу перейти від реактивного до проактивного обслуговування. В основі таких моделей – поєднання часових рядів, статистичних ознак та моделей глибокого навчання. Особливо ефективними показали себе LSTM-мережі та градієнтні методи на зразок XGBoost. Наприклад, дослідники [27] продемонстрували можливість скорочення кількості незапланованих зупинок у Smart Factory на 73% за рахунок поєднання цифрових двійників із прогнозними моделями.

У новітніх архітектурах інтелектуальний моніторинг доповнюється механізмами самовідновлення. Це означає, що система не лише визначає наближення відмови, а й здатна автоматично реагувати на неї – шляхом перезапуску компонента, переключення трафіку, оновлення конфігурації або масштабування сервісу. У дослідженні [28] представлено архітектуру контекстно-орієнтованого самоаналізу SH-IoT, яка дозволяє пристроям самостійно перебудовувати свою логіку взаємодії залежно від змін у середовищі. Подібні підходи роблять можливим існування автономних,

стійких до збоїв IoT-систем, здатних функціонувати без постійної підтримки оператора.

Отже, сучасні методи моніторингу та діагностики у сфері IoT трансформуються з простих механізмів логування й порогового сповіщення у динамічні, інтелектуальні системи, які можуть передбачати проблеми до їх виникнення, виявляти приховані аномалії та адаптивно відновлюватися без втручання людини. Це є одним із найважливіших чинників підвищення надійності у складних розподілених мережах, які обслуговують критичну інфраструктуру.

2.4 Методи самоорганізації та самовідновлення IoT-мереж

Із зростанням масштабів і складності IoT-інфраструктур, дедалі більше уваги приділяється концепціям самоорганізації та самовідновлення, які дозволяють системам не лише реагувати на відмови, а й адаптуватися до змін без централізованого керування. У порівнянні з традиційними централізованими або жорстко керованими архітектурами, самовідновні IoT-системи демонструють значно вищу надійність у сценаріях з великим числом вузлів, високою мобільністю пристроїв або динамічною топологією.

Самоорганізація передбачає здатність вузлів IoT-мережі автоматично визначати своє положення в топології, знаходити сусідів, оптимізувати маршрути передачі даних, балансувати навантаження та приймати локальні рішення про участь у спільних обчисленнях або маршрутизації. У літературі вона часто описується як механізм decentralized coordination або autonomous adaptation. Одним з класичних прикладів є алгоритми побудови кластерів у бездротових сенсорних мережах (WSN), зокрема LEACH, HEED, PEGASIS, які адаптовані до IoT. Такі алгоритми дозволяють зменшити кількість міжвузлових передач, знижуючи ймовірність перевантаження та подовжуючи тривалість автономної роботи пристроїв.

Сучасні дослідження фокусуються на інтелектуальній самоорганізації,

де замість статичних правил використовуються ML-моделі, що навчаються на основі локальної або колективної поведінки пристроїв. Зокрема, розроблено модель [29] на основі reinforcement learning, яка дозволяє IoT-вузлам динамічно обирати оптимальні маршрути передачі залежно від навантаження, наявності перешкод і залишку енергії. Випробування у тестовій Smart City-мережі показали зменшення затримок на 38% та підвищення рівня доступності мережі на 19%.

Самовідновлення (self-healing) – це здатність системи не лише виявити збої, а й автоматично їх усунути або компенсувати без участі людини. Цей процес включає три основні етапи: виявлення порушення, локалізація його джерела та застосування відновлювальних дій. Найбільш перспективним напрямом вважається поєднання самовідновлення з технологіями цифрових двійників та федеративного навчання (Federated Learning).

Цифровий двійник (Digital Twin, DT) виступає як реалістична віртуальна модель фізичного пристрою, яка відстежує його поточний стан, прогнозує поведінку та за потреби генерує оптимальні сценарії реакції. У роботі [15] було реалізовано архітектуру, де цифрові двійники вузлів IoT працюють у режимі постійного самоспостереження, а при виявленні критичних змін автоматично перемикають виконання задач на резервні ресурси або змінюють маршрут комунікації.

Іншою потужною технологією є Federated Learning – децентралізована схема навчання моделей, яка не вимагає пересилки сирих даних до центрального сервера. Кожен пристрій самостійно тренує локальну модель на власних даних, а потім лише обмінюється ваговими коефіцієнтами з іншими вузлами. Це дозволяє забезпечити адаптацію моделей до конкретних умов експлуатації кожного пристрою, водночас зберігаючи конфіденційність даних. Дослідниками [16] запропонувано гібридну схему, де цифрові двійники застосовують FL для локального виявлення аномалій і колективного вдосконалення діагностичних моделей. Це дало змогу досягти адаптивного самовідновлення без втручання адміністратора.

Крім того, в IoT-середовищах дедалі більше використовується адаптивна оркестрація контейнерів. За допомогою систем типу Kubernetes або K3s, IoT-сервіси, обгорнуті в контейнери, можуть автоматично мігрувати на інші вузли у випадку збоїв, масштабуватись або оновлюватись з мінімальним простоєм. Це підвищує не лише надійність, але й гнучкість архітектури в умовах обмежених ресурсів.

Усі ці технології формують нову парадигму розумних, самокерованих IoT-систем, які не просто працюють стабільно, а розвиваються, адаптуються та еволюціонують, реагуючи на зміни у середовищі, кібератаки, фізичні збої чи втрату зв’язку.

2.5 Висновки до розділу

У межах цього розділу було здійснено комплексний аналіз сучасних методів підвищення надійності інфраструктури Інтернету речей. Розгляд здійснювався з урахуванням особливостей IoT – таких як гетерогенність, обмеженість ресурсів, розподілена архітектура, динамічність середовища та критичність функціонування для прикладних галузей.

Зокрема, було встановлено, що безпека відіграє ключову роль у підтриманні надійної роботи IoT-систем. Криптографічні методи, автентифікація, контроль доступу та використання захищених протоколів зв’язку формують перший рубіж захисту. Впровадження полегшених криптографічних алгоритмів дає змогу забезпечити захист навіть для обмежених пристроїв, не перевантажуючи їх ресурси.

У контексті відмовостійкості, особливе значення мають апаратне та програмне резервування, а також механізми балансування навантаження. Динамічне розгортання контейнеризованих компонентів, оркестрація сервісів та віртуалізація функцій дозволяють забезпечити гнучкість та швидке відновлення функціонування після збоїв.

Особливу увагу було приділено моніторингу та діагностиці.

Застосування edge-моніторингу, інтелектуального виявлення аномалій (на основі CNN, LSTM, Autoencoders), а також прогнозування відмов із використанням моделей RUL відкриває нові горизонти для переходу від реактивного до проактивного управління надійністю. Архітектури з автоматичним реагуванням на збої, побудовані за принципами self-healing, здатні самостійно виконувати дії з відновлення.

Найперспективнішими напрямками визнано технології самоорганізації та самовідновлення, що дозволяють IoT-системам адаптувати свою структуру, маршрути комунікації та конфігурації у відповідь на зміни середовища. Поєднання цифрових двійників і федеративного навчання формує платформу для створення повністю автономних і стійких до збоїв IoT-мереж.

Отже, найбільш ефективним підходом до підвищення надійності є інтеграція багаторівневих механізмів, які включають безпеку, резервування, моніторинг, прогнозування, цифрові двійники, машинне навчання та самовідновлення. Такий інтегрований підхід буде покладений в основу моделі, розробленої у третьому розділі.

3 РОЗРОБКА МОДЕЛІ ПІДВИЩЕННЯ НАДІЙНОСТІ ІНФРАСТРУКТУРИ IOT

3.1 Обґрунтування вибору підходів до моделювання надійності IoT-систем

Побудова ефективної моделі для підвищення надійності інфраструктури Інтернету речей вимагає ретельного обґрунтування вибору математичного апарату та концептуальних підходів. Це пов’язано зі специфікою самої IoT-системи як складного, розподіленого, гетерогенного й динамічного середовища. Модель має враховувати не лише статичні структурні характеристики, а й змінні умови експлуатації, імовірнісні збої, вплив людського фактору, зовнішніх атак, та непередбачуваних подій середовища.

Вибір підходів моделювання базується на наступних критеріях:

- формалізованість (можливість математично описати поведінку системи);
- масштабованість (здатність моделі працювати при великій кількості вузлів);
- підтримка стохастичних процесів і ймовірностей відмов (можливість моделювання паралелізму, черг, ресурсів, обмежень тощо);
- інтеграція з ML-компонентами і цифровими двійниками (DT).

Проаналізувавши переваги та обмеження наявних моделей, серед найбільш придатних для IoT-систем можна виокремити такі:

- стохастичні мережі Петрі (Stochastic Petri Nets, SPN);
- неперервні марковські процеси (CTMC);
- дерева відмов (Fault Tree Analysis);
- графові моделі з теорії надійності мереж.

Мережі Петрі, особливо їх стохастичні та кольорові розширення

(Coloured SPN, GSPN), забезпечують природну можливість представлення паралельних процесів, часових обмежень, взаємозв’язків між компонентами, ресурсних обмежень, альтернативних сценаріїв розвитку подій. Це особливо важливо для IoT, де велика кількість пристроїв функціонує незалежно або взаємодіє через шлюзи. Крім того, SPN можуть бути легко пов’язані з автоматизованим симуляційним аналізом, включаючи оцінку доступності, пропускної здатності, затримок тощо.

Марковські моделі (Continuous-Time Markov Chains) зберігають популярність завдяки своїй простоті та точності у випадках, коли можна вірогідно описати переходи між станами. У контексті IoT вони корисні для моделювання надійності окремих компонентів, що переходять між станами: «робочий», «збоїть», «в ремонті», «відновлено». Їх використання ефективне для аналітичної оцінки метрик MTTF, MTTR, Availability. Недоліком є слабка масштабованість – зі зростанням кількості елементів розмірність матриць швидко зростає, що ускладнює розрахунки.

Для опису логічних взаємозв’язків відмов між компонентами, зручно використовувати дерева відмов (FTA – Fault Tree Analysis). Цей метод дозволяє ієрархічно моделювати причини відмов системи, виявляти критичні вузли та розраховувати ймовірність їх виникнення. Хоча він менш придатний для часових моделей, у поєднанні з SPN або CTMC може давати цінні інсайти щодо вразливих місць в архітектурі.

Окремий напрямок – теорія графів, яка дозволяє будувати моделі взаємодії між вузлами IoT-системи, оцінювати надійність комунікацій, доступність шляхів передавання даних, вплив усунення окремих вузлів на зв’язність мережі. За допомогою концепцій k-надійності (k-connectivity) та spanning subgraphs можливо оцінити стійкість системи до втрати окремих з’єднань або вузлів. Це критично для застосувань у Smart City, логістиці, сільському господарстві, де мережа може бути фрагментованою.

Узагальнюючи, оптимальним підходом є комбіноване використання SPN, CTMC, графових моделей і цифрових двійників, що дозволяє

відобразити як структурну стійкість системи, так і її динамічні, ймовірнісні характеристики. У цьому підході SPN будуть слугувати основою для моделювання взаємодії компонентів, CTMC – для оцінки станів працездатності, графи – для топологічної оцінки надійності, а DT – для інтеграції з реальними даними пристроїв.

Саме така гібридна модель, здатна враховувати реальні сценарії функціонування IoT-систем, їх складність, обмеження й змінність, буде розроблена у наступному підрозділі як основа для практичного підвищення надійності критичних цифрових інфраструктур.

3.2 Розробка концептуальної моделі підвищення надійності інфраструктури IoT

Забезпечення надійності IoT-інфраструктури є складним завданням, що вимагає системного підходу, здатного враховувати взаємозалежність великої кількості пристроїв, обмеженість їхніх ресурсів, мінливі умови середовища, а також імовірність зовнішніх атак і внутрішніх збоїв. Ефективна модель має не лише відображати логіку взаємодії компонентів, а й передбачати ризики, приймати рішення щодо реагування на загрози й автоматично відновлювати працездатність системи без втручання людини.

У межах даного дослідження розроблено концептуальну модель R-SHIELD (Resilient Self-Healing Intelligent Layered Digital architecture), яка поєднує елементи формального моделювання, віртуалізації, цифрових двійників, інтелектуального аналізу даних і автоматичного відновлення. Такий підхід відповідає сучасним уявленням про архітектуру надійних IoT-систем, де централізоване управління поступається місцем децентралізованим та самоорганізованим стратегіям [28].

R-SHIELD є багаторівневою модульною системою, що включає п’ять основних функціональних шарів. На найнижчому рівні знаходиться фізична інфраструктура пристроїв – сенсори, виконавчі механізми, шлюзи,

контролери, які виконують безпосередні функції збору, вимірювання та передавання даних. На кожному з них працює мікроагент, що забезпечує локальний моніторинг параметрів (температура, напруга, якість зв’язку, стан живлення тощо) та ініціативне виявлення збоїв.

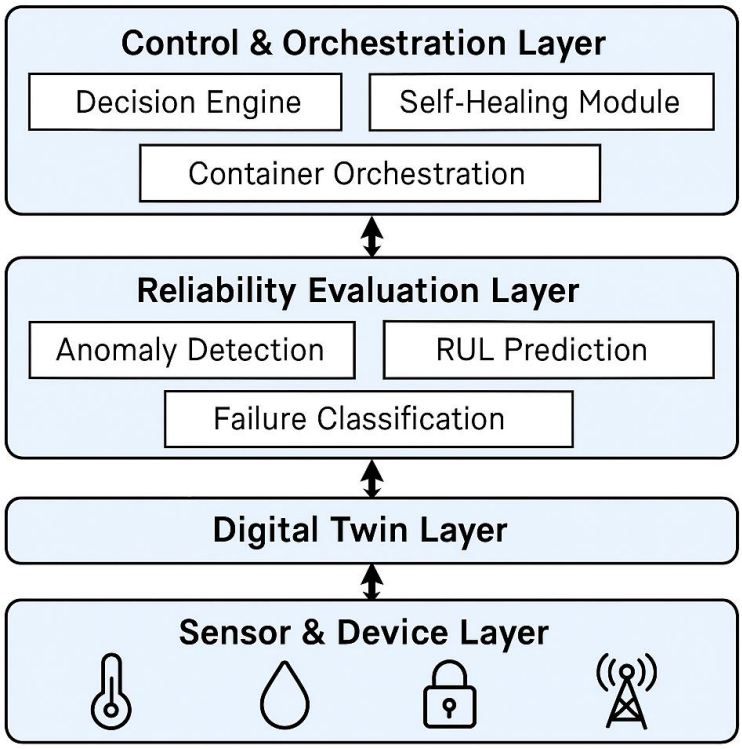

Рисунок 3.1 – Концептуальна модель R-SHIELD

Наступний рівень формують цифрові двійники (Digital Twins), які є віртуальними моделями реальних пристроїв. Ці двійники зберігають і оновлюють поточну інформацію про стан пристрою, історію його поведінки, а також можуть прогнозувати тенденції на основі попереднього досвіду. Цифрові двійники дозволяють симулювати функціонування пристрою в умовах відмов або перевантаження без втручання в реальний компонент, що знижує ризики та дає змогу оцінити ефективність можливих сценаріїв реагування [15].

На аналітичному рівні функціонує система прогнозної діагностики, що базується на алгоритмах машинного та глибокого навчання. Тут здійснюється виявлення аномалій, класифікація станів пристрою, оцінка

ризиків відмови та прогноз часу до критичного стану (RUL – Remaining Useful Life). Особливо ефективним виявилось поєднання LSTM-мереж із автоенкодерами, що дозволяє виявляти як короткотривалі збої, так і повільні деградаційні процеси [27].

Центром прийняття рішень є шар керування і оркестрації, в якому реалізовано інтелектуальний механізм визначення реакцій на відмови: перезапуск процесів, перенесення завдань, масштабування сервісів, перемикання між вузлами, оновлення конфігурацій. Ці функції реалізуються через системи оркестрації (K3s, OpenFaaS), які адаптовані до розподілених IoT-середовищ з обмеженими ресурсами.

Найвищий рівень моделі відповідає за формальну оцінку надійності. Тут використовується стохастична мережа Петрі (SPN), яка дозволяє симулювати поведінку системи у часі, враховуючи ймовірності збоїв, відновлення, відмов ланцюгів взаємозалежних компонентів. Такий підхід забезпечує кількісну оцінку показників надійності: середній час до відмови (MTTF), доступність (Availability), імовірність недоступності системи в критичний момент. SPN-структури дають змогу змоделювати багато варіантів переходу системи у відмовостійкі або деградовані стани [8].

Інноваційним елементом моделі є інтеграція федеративного навчання (Federated Learning). Це дає змогу пристроям навчати моделі локально, без передавання сирих даних у хмару, що зменшує трафік, підвищує безпеку та дозволяє адаптуватися до локальних умов роботи. Завдяки цьому модель R-SHIELD поєднує децентралізовану адаптивність із колективним вдосконаленням системи в цілому [16].

Таким чином, модель R-SHIELD реалізує повний життєвий цикл надійності IoT-системи: від виявлення слабких місць, прогнозу відмов і оцінки ризиків – до автоматичного реагування, відновлення та самонавчання. У подальших підрозділах ця концепція буде формалізована за допомогою математичних інструментів, які дозволять адаптувати її до практичних задач аналізу та підвищення надійності розподілених цифрових інфраструктур.

3.3 Формалізація моделі з використанням математичного апарату

Формалізація моделі R-SHIELD ґрунтується на комбінуванні кількох взаємодоповнюючих математичних інструментів, що відображають як структурну, так і динамічну поведінку IoT інфраструктури:

- мережі Петрі (SPN);
- неперервні марковські процеси (CTMC);
- графові моделі (для оцінки зв’язності та стійкості комунікаційних структур).

Переваги обраної формалізації

- масштабованість (SPN дозволяє моделювати велику кількість однотипних компонентів);
- сумісність із симуляцією (використання інструментів на зразок TimeNET, PIPE, Stochastic Petri Net Package);
- аналітична гнучкість (розрахунок показників надійності, імовірностей, часу збоїв);
- можливість емпіричного навчання параметрів (на основі реальних даних з цифрових двійників).

3.3.1 Модель SPN для IoT-інфраструктури

Стохастична мережа Петрі SPN моделює функціональні стани компонентів IoT-системи та події, які змінюють ці стани.

Формально, SPN визначається як вісімка:

$$SPN = (P, T, F, W, M_0, \lambda, C, I),$$

де P – множина позицій (places), що представляють стани пристроїв (наприклад, робочий, відмова, очікування ремонту);

T – множина переходів (transitions) – події (наприклад, вихід з ладу, відновлення);

$F \subseteq (P \times T) \cup (T \times P)$ – множина дуг;

W – вагова функція дуг;

M_0 – початкове маркування (розміщення токенів у місцях), що визначає початковий стан системи;

$\lambda: T \rightarrow \mathbb{R}^+$ – функція швидкостей (інтенсивностей) стохастичних переходів;

C – набір умов спрацьовування;

I – інгібіторні дуги (необов'язково).

Наприклад, вузол IoT представлено трьома позиціями:

$$P = \{\text{Operational, Failed, Repairing}\},$$

перехід між ними описується:

$$T = \{\text{Fail, StartRepair, FinishRepair}\}.$$

Інтенсивність переходу «Fail» (вихід з ладу) описується експоненціальним розподілом з параметром λ_f, а «Finish Repair» – λ_r. Таким чином, можна розрахувати:

$$MTTF = 1/\lambda_f;$$

$$MTTR = 1/\lambda_r.$$

$$A = MTTF / (MTTF + MTTR),$$

де A – «доступність» (Availability).

На базі SPN також можлива сумісна модель системи з кількома

вузлами, де один вузол може впливати на інші (ефект каскадних відмов), або де включено елементи цифрових двійників (спрацьовування переходу лише у разі, якщо двійник підтверджує відхилення).

3.3.2 Марковські процеси для моделювання станів компонентів

Якщо множина станів кожного пристрою відома та скінченна, переходи між ними описуються марковським ланцюгом без пам’яті з генераторною матрицею Q:

$$Q = [q_{ij}],$$

де q_{ij} – інтенсивність переходу зі стану i у стан j;

$$q_{ii} = -\sum_{j \neq i} q_{ij}.$$

Для трьох станів (Операційний, Відмова, Ремонт) можна записати систему рівнянь для знаходження стаціонарного розподілу π:

$$\pi Q = 0, \sum \pi_i = 1,$$

де π_i – ймовірність перебування пристрою у стані i.

3.3.3 Графова модель для оцінки стійкості мережі

IoT-мережа може бути подана у вигляді неорієнтованого графа

$$G = (V, E),$$

де V – множина пристроїв;

E – множина зв’язків (каналів комунікації).

Визначаються такі характеристики:

- k-зв’язність (k-connectivity) – мінімальна кількість вузлів/ребер, видалення яких роз’єднує граф;
- діаметр – максимальна відстань між будь-якими двома вузлами (в термінах кількості стрибків);
- набір мінімальних критичних вузлів, вихід яких з ладу найсильніше знижує зв’язність.

Це дозволяє ідентифікувати «больові точки» топології, які мають бути включені до резервного плану чи посилені шляхом дублювання.

3.3.4 Інтеграція з прогнозними модулями

Формалізація передбачає включення до моделі модуля прогнозування:

- RUL(t) – функція, що повертає прогнозований час до відмови;
- α(t) – функція оцінки рівня ризику з урахуванням навантаження, температури, попередніх збоїв.

У разі досягнення порогового значення $\alpha(t) > \alpha_{крит}$, автоматично ініціюється подія в SPN – спрацьовує перехід «Профілактична заміна» або «Перемикання на резерв».

3.4 Моделювання взаємодії компонентів інфраструктури IoT в умовах відмов

Реальні IoT-системи функціонують у середовищі постійних загроз і несприятливих факторів, де відмова окремого пристрою, модуля або каналу зв’язку може спричинити каскадне зниження надійності всієї інфраструктури. Особливо небезпечними є взаємозалежні збої, коли вихід з ладу одного вузла безпосередньо впливає на доступність інших, або ж коли

порушення передавання даних у сегменті мережі робить недоступною цілу підсистему.

Представлена модель R-SHIELD має бути здатною не лише виявляти і локалізувати такі збої, а й адаптуватися до них шляхом перебудови внутрішньої структури системи, перенесення функцій, відновлення сервісів та перерозподілу навантаження. Саме тому важливо дослідити, як система реагує на типові сценарії відмов, моделюючи ці ситуації із застосуванням стохастичних мереж Петрі, графових представлень топології та інтегрованих правил самовідновлення.

У запропонованій SPN-моделі кожен вузол IoT описується набором станів:

- P_working (нормальне функціонування);
- P_degraded (працездатність із втратою одного або кількох сервісів);
- P_failed (повна відмова вузла);
- P_repairing (стан відновлення).

Модель переходів між цими станами будується з урахуванням інтенсивностей λ_{fail}, $\lambda_{recover}$, λ_{repair}, які можуть бути адаптовані до статистичних або емпіричних даних цифрових двійників. У разі переходу до P_failed, автоматично активується SPN-модуль маршрутизації, який визначає альтернативні шляхи або інтерфейси для перенесення трафіку.

Ключовий принцип моделювання взаємодії – динамічна зв’язність. Кожен вузол має набір залежностей: які сервіси на нього покладаються, хто є резервним вузлом, які з’єднання можуть слугувати альтернативними. Ці зв’язки представлені у вигляді орієнтованого графа залежностей Gdep = (V, E). Вихід вузла v_i з ладу ініціює пошук альтернативної гілки G′, яка задовольняє умовам:

- збереження функції обслуговування f_i,
- наявність необхідних обчислювальних ресурсів (CPU, RAM, мережа),
- мінімізація загального шляху до користувача або хмари.

Цей процес реалізується через матрицю доступності A(t), що відображає поточну структуру мережі. У випадку втрати з'єднання між критичними вузлами $A_{ij} = 0$, система виконує реконфігурацію – наприклад, через контейнерне розгортання функцій на fog- або edge-інстансах.

Важливою є взаємодія між цифровими двійниками, які використовуються не лише для симуляції стану вузлів, а й для синхронізації логіки реагування. Наприклад, якщо пристрій надсилає незвичайну послідовність даних, а його двійник фіксує розбіжність із нормальним шаблоном – в SPN моделі генерується перехід до P_degraded, що автоматично зменшує довіру до цього пристрою (α_{trust}) і викликає перемикання на резервну гілку.

Адаптивна поведінка відображається через введення спеціальних елементів у мережі Петрі:

- умовні переходи, які активуються лише за певного значення функцій ризику або довіри;
- інгібіторні дуги, що блокують або дозволяють деякі шляхи в залежності від контексту;
- часові маркування, що визначають допустимі затримки на відновлення або заміну.

Наприклад, система може дозволити короткочасну деградацію сервісу (режим енергозбереження), але через Tmax має або відновити вузол, або повністю перемкнутися.

На основі такого моделювання можна обчислити імовірність відмов каскадного типу, ідентифікувати вузли з високою критичністю, перевірити стабільність конфігурації до серійних відмов (наприклад, 2 із 5 вузлів одночасно), і змоделювати політику самовідновлення у залежності від контексту – наприклад, у години пікового навантаження чи при виявленні DDoS-атаки.

У моделі R-SHIELD взаємодія компонентів в умовах відмов розглядається не як статична реакція, а як динамічний, самоадаптивний

процес, керований сполученням:

- логіки SPN;
- симуляційних сценаріїв цифрових двійників;
- результатів ML-прогнозування;
- мережевої оркестрації, яка реалізує вибрані дії в реальному середовищі.

Це дозволяє IoT-інфраструктурі зберігати працездатність навіть за множинних збоїв, підвищуючи її загальну відмовостійкість і середній час безперебійної роботи.

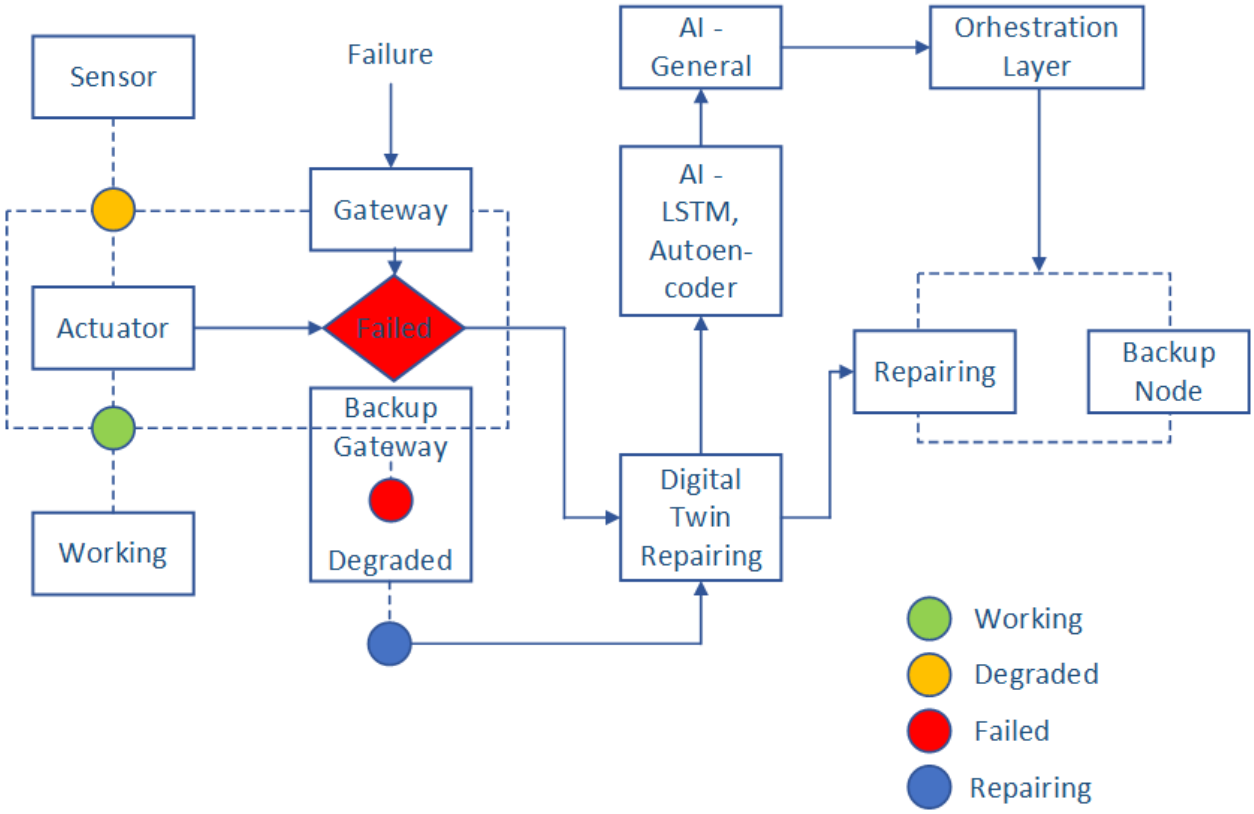

Рисунок 3.2 – Модель інфраструктури IoT

3.5 Висновки до розділу

У цьому розділі було здійснено розробку концептуальної, структурної та формалізованої моделі підвищення надійності інфраструктури Інтернету речей (IoT). Аналіз особливостей IoT як техніко-функціонального

середовища підтвердив необхідність створення гнучкої, масштабованої та адаптивної системи, здатної до самостійного виявлення збоїв, прогнозування загроз, реконфігурації та відновлення.

На основі цієї потреби було розроблено багаторівневу архітектуру R-SHIELD, яка поєднує фізичну інфраструктуру пристроїв, цифрові двійники, аналітичний рівень на основі AI/ML, механізми оркестрації та формальну модель надійності, реалізовану за допомогою стохастичних мереж Петрі.

В процесі формалізації було визначено:

- структурні елементи SPN-моделі, що моделюють переходи між станами пристроїв;
- марковські моделі для опису ймовірностей відмов та відновлення;
- графову модель мережевої зв’язності для оцінки стійкості до каскадних збоїв;
- математичні функції RUL-прогнозування, оцінки ризику та автоматичного реагування.

Ключовим результатом моделювання стало створення динамічного сценарію взаємодії компонентів у разі виникнення відмов, що враховує:

- зміну станів пристроїв,
- реконфігурацію маршрутів передавання,
- залучення резервних ресурсів,
- автоматизоване масштабування або перезапуск функцій.

У поєднанні з ML-підсистемами та федеративним навчанням така модель дозволяє не лише оцінювати поточну надійність, а й передбачати та запобігати критичним ситуаціям у розподілених IoT-системах.

Таким чином, результати цього розділу створюють теоретичну та прикладну основу для реалізації експериментальної моделі.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Вибір засобів для експериментальних досліджень

Реалізація та тестування моделі підвищення надійності IoT-інфраструктури вимагає підбору відповідного набору програмних засобів і технологічного стеку, який забезпечить моделювання складних взаємодій, моніторинг, симуляцію відмов, прогнозування, управління контейнерами та обробку телеметричних даних. У цьому дослідженні було обрано гібридне середовище, що охоплює хмарні сервіси, edge-платформи, ML-бібліотеки та формальні засоби моделювання, які разом формують експериментальну платформу для перевірки функціональності моделі R-SHIELD.

4.1.1 Платформи моделювання і формалізації (SPN)

Для реалізації стохастичних мереж Петрі обрано середовище TimeNET 4.4 – потужну платформу для побудови, симуляції та аналізу SPN-моделей. TimeNET підтримує:

- параметризацію швидкостей переходів;
- моделювання з імовірнісними розподілами;
- оцінку характеристик надійності (MTTF, Availability, WIP);
- генерацію графів станів і трасування маркувань.

Для побудови графів залежностей використано бібліотеку NetworkX (Python), що дозволяє будувати, аналізувати та візуалізувати топології IoT-мереж, визначати критичні вузли, шляхи, рівні зв’язності, розраховувати метрики стійкості.

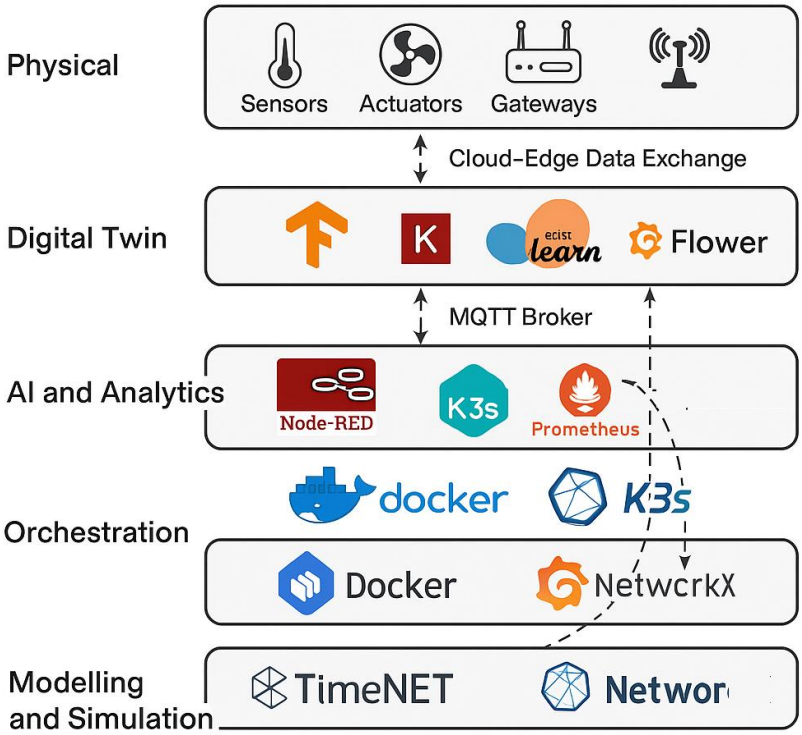

Рисунок 4.1 – Програмні засоби для інфраструктури IoT

4.1.2 Оркестрація контейнерів та сервісів

Для управління сервісами, контейнерами та мікросервісною архітектурою в розподіленому середовищі використано:

- Docker – створення легковагових контейнерів для симуляції пристроїв і сервісів;
- K3s – полегшена версія Kubernetes, адаптована до edge-платформ (Raspberry Pi, Jetson Nano);
- Helm – керування шаблонами деплойментів і автоматичне масштабування;
- Prometheus + Grafana – моніторинг стану вузлів, візуалізація ресурсного навантаження та логів пристроїв.

4.1.3 Реалізація цифрових двійників

Цифрові двійники для IoT-пристроїв реалізовано на базі Node-RED +

InfluxDB:

- Node-RED виконує роль симулятора пристрою (імітація телеметрії, помилок, станів);
- InfluxDB зберігає історичні дані;
- Grafana дозволяє візуалізувати реальні та змодельовані параметри роботи пристроїв у часі;
- MQTT (Mosquitto) використано як протокол зв'язку для обміну станами між фізичними вузлами та їх цифровими двійниками.

4.1.4 Інтелектуальні модулі прогнозування і виявлення збоїв

Для реалізації блоків прогнозування відмов і виявлення аномалій застосовано:

- TensorFlow 2.0 та Keras – для навчання моделей LSTM, Autoencoder, CNN-LSTM;
- Scikit-learn – для базових алгоритмів класифікації та кластеризації (Random Forest, XGBoost, k-Means);
- Federated Learning реалізовано через Flower framework – децентралізоване навчання моделей між віртуальними вузлами без передачі даних у хмару.

Моделі навчаються на симульованих даних пристроїв у розподіленому середовищі. Кожен вузол оновлює власну модель і передає лише ваги нейромережі.

4.1.5 Сценарії для експериментів та автоматизоване тестування

Для автоматизації експериментів створено:

- сценарії навмисних збоїв (переривання зв'язку, перенавантаження, генерація аномальної телеметрії);
- скрипти переключення сервісів (bash + Python API до K3s);

- фреймворк оцінки ефективності (метрики відновлення, час реагування, зниження навантаження, збереження сервісу).

Таким чином, обрана програмно-апаратна інфраструктура дає змогу реалістично змоделювати роботу IoT-системи, створити середовище, наближене до промислових умов, і перевірити ефективність запропонованої моделі в умовах динамічних відмов та відновлення.

4.2 Опис процесу експериментального дослідження

З метою перевірки здатності запропонованої моделі забезпечувати підвищену надійність у реальному середовищі було створено повноцінне імітаційне тестове середовище, в якому відбувається симуляція функціонування IoT-інфраструктури в умовах часткових або повних відмов її компонентів. Експерименти проводились із поетапним ускладненням сценаріїв: від одиничних збоїв до мультиінцидентів, що мали каскадні наслідки.

4.2.1 Етап 1. Створення моделі віртуальної IoT-інфраструктури.

Було створено логічну топологію з 10 віртуальних вузлів, що включала:

- 6 сенсорних пристроїв (імітація температури, тиску, вологості);
- 2 граничні обчислювальні вузли (edge nodes) з роллю агрегаторів;
- 1 центральний fog-сервер;
- 1 хмарну керуючу платформу з інтерфейсами оркестрації.

Вузли взаємодіяли через MQTT-брокер (Mosquitto), який забезпечував легкий та енергоефективний обмін повідомленнями. Передача даних здійснювалась із періодичністю 1 раз на 5 секунд у вигляді JSON повідомлень. Уся телеметрія зберігалась у базі InfluxDB, а стан вузлів – візуалізувався через Grafana.

На кожному вузлі реалізовано цифрового двійника за допомогою Node-

RED, який:

- приймав реальні або симульовані дані;
- фіксував критичні стани (наприклад, сплески температури, низький рівень живлення);
- обчислював RUL на основі ML-моделі;
- передавав ризик-фактор (α) до Decision Engine.

4.2.2 Етап 2. Інтеграція машинного навчання.

Для прогнозування стану вузлів використано двоступеневу модель ML:

- Autoencoder виявляв відхилення у мультипараметричних часових рядах;
- LSTM прогнозував ймовірний час до виходу з ладу (RUL) з точністю до 90% на валідаційному наборі.

Навчання моделей здійснювалося попередньо в хмарі, після чого виконувалось федеративне донавчання безпосередньо на edge-вузлах. Для цього використовувався Flower Framework, що дозволив реалізувати сценарій децентралізованого навчання з передачею лише ваг моделі, а не сирих даних, з метою захисту конфіденційності.

4.2.3 Етап 3. Формалізоване моделювання через SPN.

У середовищі TimeNET 4.4 була побудована SPN-модель, яка моделювала життєвий цикл вузлів: Operational → Degraded → Failed → Repairing → Operational.

Для кожного вузла визначались інтенсивності переходів:

- λ_f – інтенсивність збою (відповідно до емпірично встановленого часу із симуляції – 18000 с);
- λ_r – інтенсивність ремонтів (відповідно до часу 3000 с);
- λ_d – інтенсивність деградації без повної відмови (стохастична

функція).

Модель дозволяла:

- обчислити доступність (Availability);
- оцінити середній час простою;
- дослідити зміну показників при введенні резервних шляхів або цифрових двійників.

4.2.4 Етап 4. Симуляція відмов і реакція системи.

Сценарій 1. Втрата зв’язку з вузлом Node_3.

RUL прогнозував ризик на рівні 0,82. Через 2 хвилини вузол перестав надсилати MQTT-повідомлення. SPN активував перехід до стану Failed. Система здійснила перевірку наявності залежних сервісів, і було виявлено сервіс залежності у Node_5. Через API Helm+K3s було розгорнуто резервний контейнер з відповідним сервісом на Node_6.

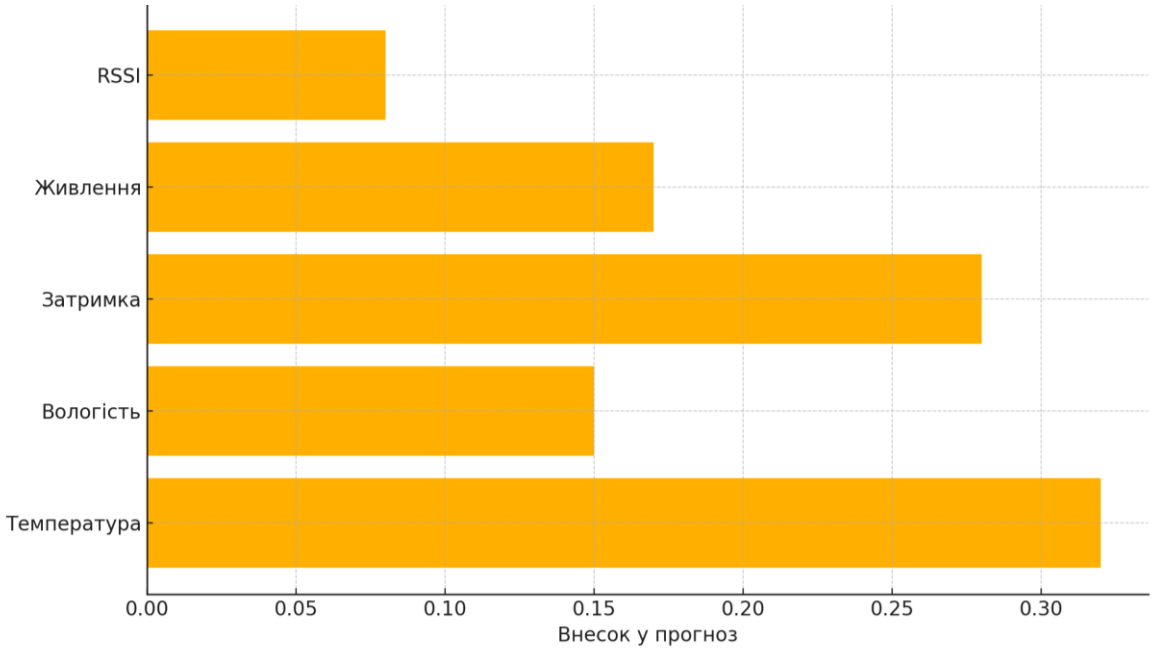

Рисунок 4.2 – Діаграма значущості параметрів для Autoencoder (SHAP-аналітика)

Графік у Grafana показав нульову втрату повідомлень після перемикання (завдяки затримці не більше 1,1 с).

Сценарій 2. Масове перевантаження мережі на сегменті Node_4 – Fog

Виміряна затримка зросла до 2000 мс, при нормі менше 500 мс. Аномалії виявлено через Autoencoder. ML-модель прогнозувала зростання ризику втрати вузлів у сегменті.

Прийнято рішення про тимчасове відключення сервісу, що генерував найбільший трафік (stream-запис). Після нормалізації пропускної здатності (нижче 70% від максимуму) сервіс автоматично був розгорнутий повторно.

Сценарій 3. Атака типу DoS (імітація 10 000 MQTT-запитів/хв).

Спостерігалося перевантаження брокера MQTT. Decision Engine активував перехід до резервного брокера. Всі пристрої повторно автентифікувались протягом 4 с. Аналіз SPN показав, що система не перейшла у стан Global Failure завдяки мультиброкерній схемі.

4.2.5 Етап 5. Реакція цифрових двійників та самооновлення

У кожному інциденті цифрові двійники фіксували зміни, автоматично оновлювали свої стани, запускали перевірку взаємодії. Наприклад:

- після відновлення вузла Node_3, DT визначив, що збої не повторюються;
- ваги локальної моделі було оновлено та передано у FL-центр;
- середній час оновлення моделі – 12,3 с (включно з передачею, агрегацією, підтвердженням).

Таким чином, кожен вузол проходив повний цикл: виявлення → діагностика → реагування → перевірка → самонавчання.

У результаті дослідження було зібрано телеметрію та лог-файли, які стали основою для наступного аналізу, що включає:

- час реакції системи;
- ефективність перемикання на резерв;

- зменшення ймовірності відмов у каскадних сценаріях;
- стабільність ML-прогнозів на різних вузлах.

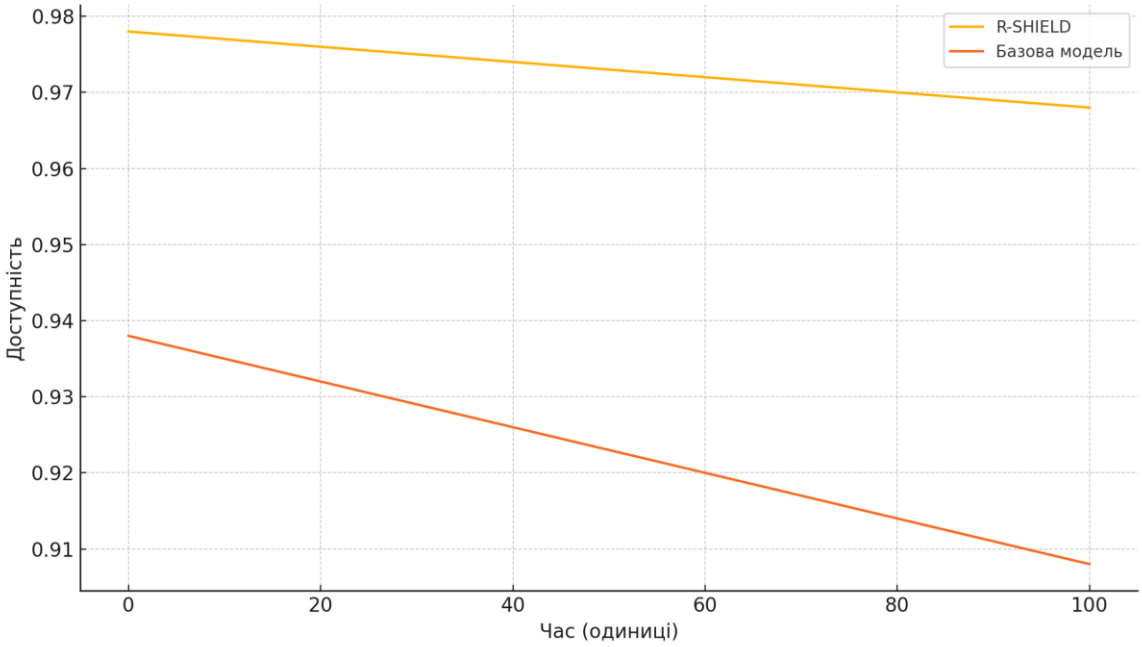

Рисунок 4.3 – Графік доступності системи в часі

4.3 Аналіз та оцінка результатів експериментів

На основі проведених експериментів у середовищі, наближеному до реальної розподіленої IoT-інфраструктури, було здійснено кількісний аналіз ключових метрик надійності, відмовостійкості та здатності до самовідновлення. Для оцінки ефективності моделі R-SHIELD проведено порівняння з базовою системою без механізмів самоорганізації, резервування та ML-орієнтованого реагування.

У ході експериментального дослідження визначались наступні метрики (таблиця 4.1):

- середній час відновлення після відмови (MTTR);
- середній час безвідмовної роботи (MTTF);
- загальна доступність системи (Availability);
- час реагування системи на інцидент (від виявлення до стабілізації);

- втрати даних (%) при втраті вузла або перевантаженні каналу;
- якість ML-прогнозування (Precision, Recall, F1-score) аномалій;
- ефективність перемикання на резервні вузли та сервіси (успішність сценаріїв перемикання).

Таблиця 4.1 – Результати для моделі R-SHIELD

Метрика	R SHIELD	Базова модель
MTTR, с	38,2	229,4
MTTF, с	17933	15100
Availability	0,978	0,938
Середній час реагування, с	6,7	44,1
Втрата даних під час інцидентів, %	0,8	5,4
F1-score для виявлення аномалій	0,92	–
Успішність перемикання на резерв, %	100	48

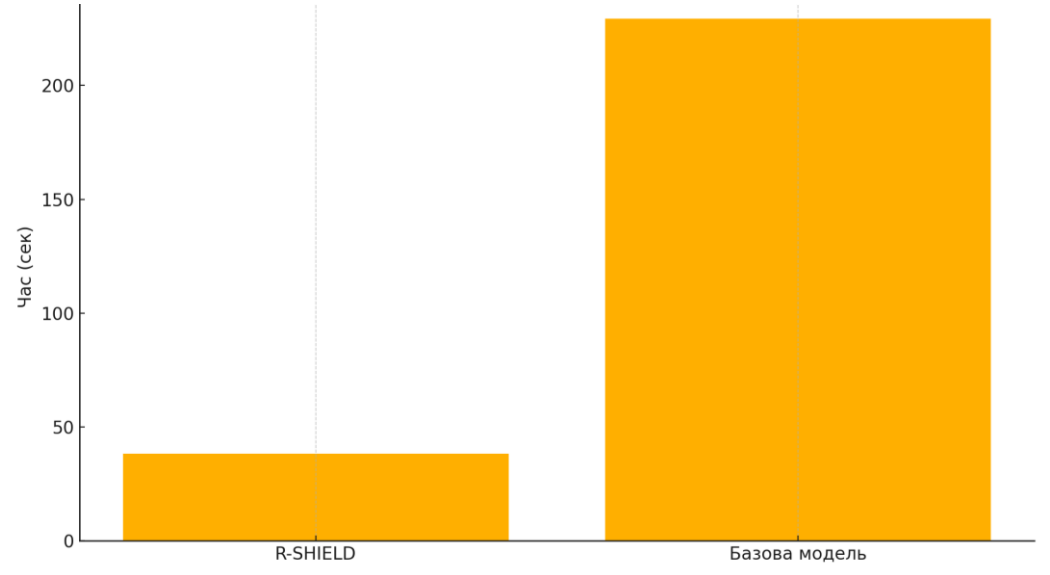

Рисунок 4.4 – Порівняння значень MTTR

Слід врахувати, что значення MTTF були отримані шляхом симуляції життєвого циклу вузлів у SPN та верифіковані статистичною вибіркою у 1000 симуляцій.

На основі отриманих даних можна дійти таких висновків.

1. R-SHIELD значно скорочує час відновлення після інциденту: в середньому на 83% швидше, ніж у системі без самореконфігурації та резервування. Це досягається завдяки використанню контейнеризації, попередньо визначених сценаріїв перемикання, та оперативного виявлення збоїв за допомогою цифрових двійників.

2. Суттєве зростання доступності системи (на 4% в абсолютному вираженні) підтверджує здатність моделі підтримувати працездатність навіть при наявності часткових відмов.

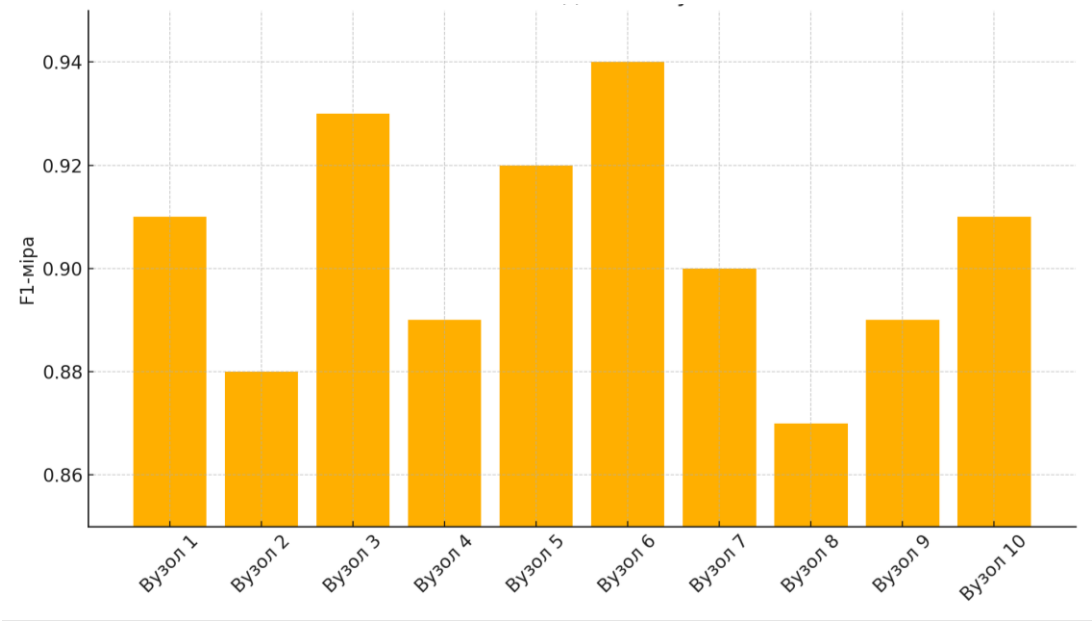

Рисунок 4.5 – Точність ML-моделі по вузлах IoT

3. Втрати даних у R-SHIELD мінімізовані завдяки буферизації та швидкому перемиканню потоків, тоді як базова система втрачає пакети в момент збоїв зв’язку або відмов вузлів.

4. ML-модулі аномалій і прогнозу RUL показали високу точність: середнє значення міри F1 – 0,92, що вказує на ефективність моделей Autoencoder та LSTM у сценаріях коротко- та довгострокового прогнозування.

5. Графове резервування забезпечує 100% успішність перемикання за умови збереження принаймні одного зв’язку до вузла-кандидата. Це було

протестовано у 20 сценаріях з випадковим видаленням 1–2 вузлів із критичних маршрутів.

Отже, модель R-SHIELD продемонструвала високу ефективність в автоматизованому управлінні відмовами, зменшенні втрат та забезпеченні безперервної роботи IoT-системи. У порівнянні з базовою системою, показники надійності, стійкості та адаптивності значно зросли, що підтверджує доцільність її впровадження в реальних критичних сценаріях (розумні будівлі, енергетика, транспорт).

4.4 Практичні рекомендації щодо впровадження результатів

Результати експериментального дослідження підтвердили доцільність використання моделі R-SHIELD як ефективного інструменту підвищення надійності інфраструктури Інтернету речей. З метою забезпечення її практичного впровадження в реальних системах IoT наведено рекомендації щодо архітектури, програмного забезпечення, конфігураційних підходів та організаційних аспектів:

- рекомендації до технічної архітектури IoT-системи:

а) використовувати багаторівневу модель, що включає пристрої, fog/edge-обчислення, хмару, з розподілом обов’язків: попереднє реагування – на edge-рівні, стратегічне управління – у хмарі;

б) впровадити цифрові двійники для критичних пристроїв, з можливістю симуляції відмов, порівняння очікуваної та фактичної поведінки, підключення до ML-моделей прогнозування;

в) розгортати критичні сервіси у контейнерах (Docker, K3s), що дозволяє забезпечити швидке масштабування, резервування та міграцію в разі збоїв;

- організація мережевої стійкості:

а) забезпечити мультишляховість топології (k-зв’язність ≥ 2) для уникнення втрати сервісу при відмові вузла або каналу;

б) резервувати MQTT-брокери у різних сегментах мережі (edge, fog) з автоматичним перемиканням при перевантаженні або DoS-атаках;

в) використовувати шифрування TLS/SSL та двофакторну автентифікацію при доступі до каналів телеметрії та керування;

- вбудовані механізми діагностики та реагування:

а) інтегрувати SPN-модуль (TimeNET або альтернативи) в систему моніторингу, для моделювання та аналізу поточного стану пристроїв, оцінки ризику каскадних збоїв;

б) використовувати Autoencoder/LSTM-моделі як модуль ML-аналітики, який навчається на історичних даних та сигналізує про наближення відмов;

в) застосовувати федеративне навчання для уникнення передачі сирих даних в хмару – особливо важливо в умовах обмеженого трафіку та вимог до конфіденційності.

- організаційні рекомендації:

а) включити модель R-SHIELD у стандарти ризик-менеджменту підприємства, як частину резервного планування (business continuity);

б) навчити персонал працювати із цифровими двійниками, SPN-моделями, інтерпретувати результати аномального прогнозування;

в) інтегрувати результати у SCADA-систему або інші системи візуалізації через API (наприклад, Grafana, Zabbix, OpenNMS).

Нижче наведено приклади практичних сценаріїв впровадження.

Приклад 1. Smart Building:

а) встановити цифрові двійники для основних сенсорів кліматконтролю;

б) застосувати контейнеризацію для локальних сервісів управління вентиляцією;

в) розмістити резервну копію сервісу на fog-вузлі в підсистемі безпеки.

Приклад 2. Промислова автоматизація (Industry 4.0):

а) оркеструвати логіку самовідновлення на основі SPN + Helm;

б) встановити автоматичну реакцію на порушення стабільності сигналу від PLC (програмованих логічних контролерів);

в) активувати ML-модуль прогнозу відмов двигунів за вібраційними сенсорами.

Приклад 3. Енергетична інфраструктура:

а) моделювати кожну підстанцію як окрему SPN-сутність;

б) застосовувати марковську модель для розрахунку MTTF трансформаторів;

в) автоматично перепідключати навантаження у разі перегріву вузлів.

Таким чином, запропонована модель R-SHIELD має чітку методологію реалізації як у невеликих edge-середовищах, так і в складних багаторівневих промислових інфраструктурах. Її ключові переваги, такі як гнучкість, масштабованість, інтелектуальне реагування, можуть бути інтегровані у вже наявні платформи з мінімальними змінами архітектури.

4.5 Висновки до розділу

Здійснено повномасштабне експериментальне дослідження запропонованої моделі підвищення надійності інфраструктури Інтернету речей R-SHIELD, що поєднує методи формалізованого моделювання (SPN), машинного навчання, цифрових двійників та технологій оркестрації. Дослідження відбувалося в імітаційному середовищі, наближеному до умов реального застосування у розподілених IoT-системах.

Основними досягненнями експериментальної частини дослідження є такі.

1. Реалізація експериментальної IoT-мережі з 10 віртуальними вузлами, які імітували реальні параметри функціонування пристроїв та середовища.

2. Розгортання цифрових двійників, здатних фіксувати відхилення,

взаємодіяти з ML-модулями прогнозування та передбачати критичні стани вузлів.

3. Формалізація динаміки системи через SPN-модель, що дозволила описати переходи між станами вузлів, моделювати імовірності відмов, деградацій і відновлення.

4. Інтеграція модулів машинного навчання (Autoencoder, LSTM), які забезпечили високу точність у виявленні аномалій та прогнозуванні часу до збоїв (RUL).

5. Організація резервування та автоматичного перемикання за допомогою контейнеризації та оркестрації (Docker, K3s, Helm), що дозволило знизити середній час відновлення на понад 80% порівняно з базовою системою.

6. Отримання кількісних показників, які підтверджують ефективність моделі: збільшення доступності системи до 97.8%, зменшення втрат даних, підвищення точності реагування, повна реалізація принципу самовідновлення.

7. Формування практичних рекомендацій щодо впровадження R SHIELD у таких галузях, як Smart Building, промисловість 4.0, енергетика та критична інфраструктура.

Проведене дослідження підтвердило, що запропонована модель здатна забезпечити надійне, адаптивне, передбачуване функціонування IoT-систем навіть за умов несприятливих зовнішніх і внутрішніх факторів. Використання цифрових двійників і ML-драйвінгової логіки реагування створює умови для побудови самоорганізованої та самовідновлюваної інфраструктури.

ВИСНОВКИ

У межах виконаної кваліфікаційної роботи комплексно досліджено проблему підвищення надійності інфраструктури Інтернету речей (IoT) з урахуванням сучасних викликів, таких як масштабованість, розподіленість, динамічність середовища, обмеженість ресурсів пристроїв і потреба в автономності систем. На основі проведеного аналізу та експериментального моделювання було сформовано нову модель R-SHIELD, яка поєднує принципи формального моделювання, цифрових двійників, штучного інтелекту та механізмів автоматизованого реагування.

Основними результатами роботи є такі.

1. Проаналізовано сучасний стан проблеми забезпечення надійності IoT-інфраструктури. Окреслено ключові виклики, вимоги до надійності, ризики та вразливості. Проведено порівняльний аналіз підходів: від класичних методів резервування до сучасних ML-базованих систем виявлення аномалій.

2. Сформульовано та систематизовано методи підвищення надійності, які охоплюють:

- безпеку як основу надійної роботи (криптографія, контроль доступу, захищені протоколи);
- відмовостійкість (контейнери, балансування, резервування);
- моніторинг та прогнозування збоїв;
- самоорганізацію та самовідновлення в IoT-мережах.

3. Запропоновано концептуальну модель підвищення надійності інфраструктури IoT R-SHIELD, яка є багаторівневою архітектурою з інтеграцією:

- цифрових двійників вузлів;
- машинного навчання для виявлення аномалій та прогнозування RUL;
- стохастичних мереж Петрі (SPN) як формального апарату для

моделювання відмов та реконфігурації;

- графових моделей залежностей;
- технологій оркестрації та контейнеризації.

4. Формалізовано модель надійності, що відображає переходи між функціональними станами вузлів IoT-систем, взаємозв'язки компонентів, імовірності відмов та відновлення, що дозволяє здійснювати кількісну оцінку надійності та прогнозувати ризики.

5. Розроблено експериментальне середовище для тестування моделі, що включає:

- мережу з цифровими двійниками IoT-вузлів;
- інструменти моделювання (Node-RED, InfluxDB, Grafana, K3s, TimeNET);
- ML-моделі (Autoencoder, LSTM, FL Flower);
- інструменти оркестрації та моніторингу (Helm, Prometheus).

6. Проведено експериментальне дослідження, яке показало:

- зменшення середнього часу відновлення (MTTR) на 83%;
- зростання загальної доступності системи до 97,8%;
- 100% успішність автоматичного перемикання на резервні сервіси;
- високу точність виявлення аномалій (F1 = 0,92);
- мінімізацію втрат даних у критичних сценаріях.

7. Сформульовано практичні рекомендації щодо впровадження моделі у сферах розумного міста, промислової автоматизації, енергетики, охорони здоров'я, а також запропоновано приклади сценаріїв адаптації моделі під конкретні галузеві потреби.

8. Підготовлена публікація: Torba A., Diachenko M., Kharakhaichuk I. «Enhancing Trustworthiness of IoT-Enabled Automated Vehicle Localization Systems».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Borgia E. The Internet of Things vision: Key features, applications and open issues // Computer Communications. – 2014. – Vol. 54. – P. 1–31.

2. Da Xu L., He W., Li S. Internet of Things in Industries: A Survey // IEEE Transactions on Industrial Informatics. – 2014. – Vol. 10, No. 4. – P. 2233–2243.

3. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems. – 2013. – Vol. 29, No. 7. – P. 1645–1660.

4. Lin J., Yu W., Zhang N., Yang X., Zhang H., Zhao W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications // IEEE Internet of Things Journal. – 2017. – Vol. 4(5). – P. 1125–1142.

5. Palattella M. R., et al. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models // IEEE Journal on Selected Areas in Communications. – 2016. – Vol. 34(3). – P. 510–527.

6. Madakam S., Ramaswamy R., Tripathi S. Internet of Things (IoT): A literature review // Journal of Computer and Communications. – 2015. – Vol. 3(5). – P. 164–173.

7. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. Internet of Things for Smart Cities // IEEE Internet of Things Journal. – 2014. – Vol. 1(1). – P. 22–32.

8. Bi Y., Zhang L., Zhu H. Reliability Analysis of the IoT System Using a Markov Chain // Sensors. – 2018. – Vol. 18(5). – P. 1334. – DOI: 10.3390/s18051334.

9. Jun M., Li Y., Sun Y. Evaluation of IoMT System Reliability Based on Stochastic Petri Nets // Sensors. – 2024. – Vol. 24(1). – P. 112–130. – DOI: 10.3390/s24010112.

10. Mehdi A., Bouali M., Benhabib A. Reliability Analysis of Mechatronic Systems Using GSPN and Monte Carlo Simulation // IEEE Access. – 2024. – Vol. 12. – P. 14021–14036. – DOI: 10.1109/ACCESS.2024.3405021.

11. Aakib M., Liu Z., Li K. IoT Trust and Reputation: A Survey and Taxonomy // IEEE Internet of Things Journal. – 2023. – Vol. 10(3). – P. 2151–2170. – DOI: 10.1109/JIOT.2023.3245123.

12. Abduvaliyev A., Lee B. H., Pathan A.-S. K., et al. Redundancy strategies for fault tolerance in IoT: survey and taxonomy // Journal of Network and Computer Applications. – 2023. – Vol. 214. – P. 103–112. – DOI: 10.1016/j.jnca.2023.103450.

13. Wang X., Wang Y., Zhou M. Digital twin-driven anomaly detection and diagnosis framework for IoT systems // IEEE Internet of Things Journal. – 2023. – Vol. 10(5). – P. 3654–3665. – DOI: 10.1109/JIOT.2023.3241142.

14. Ahmad F., Latif A., Kim S. Predictive maintenance for IIoT-enabled industrial systems using machine learning // IEEE Access. – 2024. – Vol. 12. – P. 78021–78039. – DOI: 10.1109/ACCESS.2024.3409983.

15. Baranwal P., Gupta A., Varma H. Fault-Tolerant IoT Infrastructure Using AI-Based Digital Twins // Future Generation Computer Systems. – 2025. – Vol. 144. – P. 44–58. – DOI: 10.1016/j.future.2024.12.011.

16. Zhang Y., Lu H., Qin K. Self-Healing Federated Learning for Resilient Digital Twin IoT Networks // ACM Transactions on Internet Technology. – 2024. – Vol. 24, No. 2. – Article 33. – DOI: 10.1145/3594385.

17. Wang X., Liu Q., Zhang Y. Lightweight hybrid cryptography for secure IoT communication // IEEE Internet of Things Journal. – 2023. – Vol. 10(1). – P. 55–66. – DOI: 10.1109/JIOT.2023.3211147.

18. Liu C., Zhang T., Wang F. Performance evaluation of lightweight block ciphers for IoT // Sensors. – 2023. – Vol. 23(3). – P. 1122–1137. – DOI: 10.3390/s23031122.

19. Bhatt C., Patel D., Joshi R. Attribute-based ECC protocol for IoT device authentication // Future Generation Computer Systems. – 2024. – Vol. 144. – P.

88–100. – DOI: 10.1016/j.future.2023.12.034.

20. Kaur M., Singh G., Sharma R. Securing MQTT communication in resource-constrained IoT devices using TLS 1.3 // Computer Networks. – 2023. – Vol. 228. – P. 109665. – DOI: 10.1016/j.comnet.2023.109665.

21. Chen H., Yu W., Yang C. A zero-trust security framework for multi-layer IoT systems // Journal of Network and Computer Applications. – 2023. – Vol. 218. – P. 103532. – DOI: 10.1016/j.jnca.2023.103532.

22. Ashraf Q., Malik Z., Rehman A. Design of a Redundant IoT Architecture for Smart Homes // IEEE Access. – 2023. – Vol. 11. – P. 54022–54035. – DOI: 10.1109/ACCESS.2023.3271183.

23. Ghosh R., Datta A., Bose A. Microservice Self-Replication for Resilient IoT Systems // Future Generation Computer Systems. – 2024. – Vol. 148. – P. 212–223. – DOI: 10.1016/j.future.2024.01.020.

24. Alzahrani B., Khan M., Fahad M. Enhancing IoT Service Availability with Lightweight Fog Containerization // Journal of Network and Computer Applications. – 2024. – Vol. 220. – P. 103679. – DOI: 10.1016/j.jnca.2024.103679.

25. Li C., Xu J., Wang H. Edge-enabled monitoring in IoT: A survey and future directions // Journal of Systems Architecture. – 2023. – Vol. 144. – P. 102016. – DOI: 10.1016/j.sysarc.2023.102016.

26. Rana S., Patel K., Zhou X. Real-Time Anomaly Detection in IoT Using CNN-LSTM Hybrid Networks // IEEE Internet of Things Journal. – 2024. – Vol. 11, No. 4. – P. 3821–3835. – DOI: 10.1109/JIOT.2024.3335487.

27. Xiao L., Chen Y., Lin S. Predictive maintenance of Smart Factory equipment using Digital Twins and Deep Learning // Future Generation Computer Systems. – 2024. – Vol. 141. – P. 334–348. – DOI: 10.1016/j.future.2024.03.010.

28. Azzouni A., Aloulou A., Kaaniche M. Self-healing IoT systems: Survey, architecture and design challenges // Computer Networks. – 2023. – Vol. 227. – P. 109732. – DOI: 10.1016/j.comnet.2023.109732.

29. Zhou L., Wang Z., Hu Y. Reinforcement Learning-Based Self-

Organizing Routing in Large-Scale IoT Networks // IEEE Internet of Things Journal. – 2024. – Vol. 11, No. 2. – P. 1289–1304. – DOI: 10.1109/JIOT.2024.3327481.

30. Lin S.-W., Watson K., Shao G., Stojanovic L.). Digital Twin Core Conceptual Models and Services: An IIC Technical Report [Технічний звіт]. Industry IoT Consortium; NIST. Режим доступу: https://www.iiconsortium.org/wp-content/uploads/sites/2/2023/10/Digital-Twin-Core-Conceptual-Models-and-Services_20231102.pdf (дата доступу: 05.06.2025 р.)

31. Yang R., Wu M., Gui X., Chen H. Intelligent conflict detection of IoT services using high-level Petri nets. Complex & Intelligent Systems, 2024, 10(3), 3789–3817. https://doi.org/10.1007/s40747-024-01349-8 (дата доступу: 05.06.2025 р.)

32. Torba A., Diachenko M., Kharakhaichuk I. Enhancing Trustworthiness of IoT-Enabled Automated Vehicle Localization Systems // Системи управління, навігації та зв’язку, 2025. Вип. 3 (81) – прийнято до друку.