ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

# «**МЕТОД ПРОГНОЗУВАННЯ ЗБОЇВ ДЛЯ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ**»

на здобуття освітнього ступеня магістр
за спеціальності 126 Інформаційні системи та технології
*(код,найменування спеціальності)*
освітньо-професійної програми Інформаційні системи та технології
*(назва)*

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

_____ Кирилл ЧУГРЕЄВ
*(підпис)* *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти
група ІСДМ-61 Кирилл ЧУГРЕЄВ
*(ім'я, ПРІЗВИЩЕ)*
Керівник Вікторія ЖЕБКА
*д.т.н., проф.* *(ім'я, ПРІЗВИЩЕ)*
Рецензент: _____
*(ім'я, ПРІЗВИЩЕ)*

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут Інформаційних технологій**

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти магістр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна прорама Інформаційні системи та технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедрою ІСТ
_____ Каміла СТОРЧАК

"____" _____ 2025 року

# З А В Д А Н Н Я
## НА КВАЛІФІКАЦІЙНУ РОБОТУ

Чугреєву Кириллу Олександровичу
*(прізвище, ім'я, по батькові здобувача)*

1.Тема кваліфікаційної роботи: Метод прогнозування збоїв для мереж Інтернету речей
керівник кваліфікаційної роботи: Вікторія ЖЕБКА д.т.н., професор
*(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*
затверджені наказом Державного університету інформаційно-комунікаційних технологій від "30" жовтня 2025 р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані кваліфікаційної роботи:
1. Технології Інтернет речей.
2. Архітектура IoT.
3. Методи машинного навчання в IoT.
4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Дослідження тенденцій розвитку та поширення Інтернет речей.
2. Огляд методів машинногго навчання в IoT.
3. Аналіз результатів гібридного методу машиного навчання в IoT.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «30» жовтня 2025р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Підбір технічної літератури | | |
| 2. | Дослідження тенденцій розвитку та поширення Інтернету речей | | |
| 3. | Дослідження методів прогнозування збоїв в IoT | | |
| 4. | Результати розробки проактивного методу прогнозуваннязбоїв LGFP в IoT | | |
| 5. | Висновки по роботі | | |
| 6. | Розробка демонстраційних матеріалів, доповідь. | | |
| 7. | Оформлення магістерської роботи | | |

Здобувач вищої освіти _____ Кирилл ЧУГРЕЄВ
*( підпис )* *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Вікторія ЖЕБКА
*(підпис )* *(ім'я, ПРІЗВИЩЕ)*

# РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступеня магістра: __ стор., 4 рис., 14 табл., 32 джерела.

*Мета роботи* - покращення процесу виявлення та прогнозування збоїв у системах Інтернету речей за допомогою розробленого методу з використанням комбінованої архітектури машинного навчання на основі графових нейронних мереж та довгої короткочасної пам’яті.

*Об’єкт дослідження* - процес виявлення та прогнозування збоїв у системах Інтернету речей.

*Предмет дослідження* - методи та алгоритми машинного навчання, застосовані для прогнозування збоїв у децентралізованих мережах IoT.

*Короткий зміст роботи.*

У першому розділі розглянуто сучасний стан розвитку технологій Інтернету речей, наведено класифікацію IoT-систем за рівнями архітектури, типами підключення та доменами застосування. Описано типові проблеми надійності та збої у роботі IoT-пристроїв, а також проаналізовано існуючі підходи до прогнозування та запобігання відмовам.

У другому розділі проведено аналітичне дослідження наборів даних, структур IoT-трафіку та попередню статистичну обробку. Виконано порівняння популярних алгоритмів машинного навчання, таких як Logistic Regression, Naive Bayes, Stochastic Gradient Descent, Support Vector Machine, k-Nearest Neighbors, Decision Tree, Random Forest, XGBoost, MLP, LSTM, GNN. Проведено оцінку їх точності, повноти та стабільності на тестових вибірках.

У третьому розділі розроблено новий комбінований метод LGFP, що поєднує просторовий аналіз міжвузлових зв’язків на основі GNN та часове прогнозування аномалій на базі LSTM. Описано архітектуру методу, етапи обробки даних, механізми адаптації та повторного навчання моделі. Розглянуто інтеграцію системи у реальну IoT-інфраструктуру, взаємодію модулів з рівнями Edge, Fog та

Cloud, а також наведено алгоритм роботи LGFP з урахуванням різних сценаріїв обробки даних.

Проведено експериментальні дослідження ефективності методу, результати яких показали перевагу запропонованого підходу у порівнянні з класичними моделями за метриками точності та F1-мірою. Виявлено основні проблемні зони, зокрема залежність від якості даних та затримки синхронізації вузлів.

Розроблений метод може бути використаний для побудови інтелектуальних систем технічного обслуговування, підвищення надійності промислових мереж та кіберфізичних систем у реальному часі.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, МАШИННЕ НАВЧАННЯ, LSTM, GNN, ПРОГНОЗУВАННЯ ЗБОЇВ, АЛГОРИТМ , IoT, ТОЧНІСТЬ, ПОВНОТА, F1.

## ABSTRACT

The text part of the master's qualification thesis: __ pages, 4 figures, 14 tables, 32 references.

*The purpose of the work* is to develop and study a method for predicting failures in Internet of Things systems using a combined machine learning architecture based on Graph Neural Networks and Long Short-Term Memory.

*Object of research* is the process of detecting and predicting failures in IoT systems.

*Subject of research* is machine learning methods and algorithms applied to failure prediction in decentralized IoT networks.

*Summary of the work.*

The first chapter reviews the current state of IoT technology development, provides a classification of IoT systems by architectural levels, connection types, and application domains. It describes typical reliability issues and device failures in IoT systems, as well as analyzes existing approaches to failure prediction and prevention.

The second chapter presents an analytical study of datasets, IoT traffic structures, and preliminary statistical processing. It includes a comparison of popular machine learning algorithms such as Logistic Regression, Naive Bayes, Stochastic Gradient Descent, Support Vector Machine, k-Nearest Neighbors, Decision Tree, Random Forest, XGBoost, MLP, LSTM, and GNN. Their accuracy, recall, and stability were evaluated on test samples.

The third chapter introduces a new combined method - LGFP - which integrates spatial analysis of inter-node relationships based on GNN with temporal anomaly forecasting using LSTM. The architecture of the method, data processing stages, model adaptation, and retraining mechanisms are described. The integration of the system into real IoT infrastructure is discussed, including the interaction of modules across Edge, Fog, and Cloud layers. An algorithm of LGFP operation under various data processing scenarios is presented.

Experimental studies of the method’s effectiveness demonstrated its advantage over classical models in terms of accuracy and F1-score metrics. Key challenges were identified, particularly the dependence on data quality and node synchronization delays.

The developed method can be applied to build intelligent maintenance systems and enhance the reliability of industrial networks and cyber-physical systems in real time.

**Keywords:** INTERNET OF THINGS, MACHINE LEARNING, LSTM, GNN, FAILURE PREDICTION, ALGORITHM, IoT, ACCURACY, RECALL, F1-SCORE.

**ЗМІСТ**

ВСТУП ..... 11
1 ТЕОРЕТИКО-МЕТОДИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ЗБОЇВ У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ ..... 14
1.1 Сутність та особливості функціонування мереж Інтернету речей ..... 14
1.2 Надійність і відмовостійкість у системах Інтернету речей ..... 18
1.3 Аналіз сучасних алгоритмів машинного навчання у прогнозуванні збоїв у системах Інтернету речей ..... 22
2 ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЗБОЇВ ТА АНАЛІЗ ВХІДНИХ ДАНИХ ..... 27
2.1 Характеристика об’єкта дослідження та даних для моделювання ..... 27
2.2 Аналіз структури даних та попередня статистична обробка ..... 32
2.3 Аналіз частоти та причин збоїв у системах Інтернету речей ..... 36
2.4 Вибір моделей машинного навчання, їх навчання та тестування ..... 39
2.5 Аналіз результатів, виявлення проблемних зон ..... 54
3 РОЗРОБКА МЕТОДУ ПРОГНОЗУВАННЯ ЗБОЇВ ДЛЯ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ ..... 57
3.1 Обґрунтування вибору та архітектури нового методу прогнозування збоїв у системах IoT ..... 57
3.2 Інтеграція методу в IoT-інфраструктуру ..... 68
ВИСНОВКИ ..... 72
ПЕРЕЛІК ПОСИЛАНЬ ..... 74
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ(Презентація) ..... 78

# ВСТУП

*Актуальність теми.*

Стрімкий розвиток технологій Інтернету речей (IoT) відкриває нові можливості для побудови інтелектуальних середовищ, у яких взаємодіють тисячі пристроїв, сенсорів і контролерів. Водночас зростання кількості таких вузлів і обсягу даних, що ними генеруються, призводить до підвищення ризику збоїв, зниження продуктивності та втрати критичної інформації. Своєчасне виявлення потенційних проблем у роботі IoT-мережі є ключовим чинником забезпечення її надійності, стабільності та безперервності обслуговування користувачів.

Традиційні підходи до моніторингу базуються на реактивних методах, коли збій фіксується після його виникнення. Такий підхід не дозволяє своєчасно запобігти поширенню відмов і не враховує динамічний характер IoT-середовища. Тому актуальним завданням є розробка методу прогнозування збоїв, який може адаптивно аналізувати поведінку мережі, виявляти закономірності у змінах параметрів і заздалегідь попереджати про можливі відмови.

Останніми роками значного поширення набули методи машинного навчання, які дозволяють здійснювати автоматизоване виявлення аномалій та прогнозування з високою точністю. Проте більшість існуючих рішень не враховує просторову структуру IoT-мережі, де зв’язки між вузлами мають суттєвий вплив на поширення збоїв. Крім того, багато моделей орієнтовані на статичні дані й не здатні коректно обробляти часові залежності, характерні для потоків телеметричної інформації.

У зв’язку з цим у роботі запропоновано інноваційний підхід до прогнозування збоїв у мережах Інтернету речей - LSTM-GNN Fault Predictor (LGFP). Його особливість полягає у поєднанні можливостей графових нейронних мереж (GNN) для моделювання топології мережі та рекурентних нейронних мереж LSTM для аналізу часової динаміки параметрів. Запропонована система забезпечує адаптивне прогнозування на основі комбінованого аналізу просторово-часових залежностей у телеметричних даних IoT.

*Мета дослідження*

Покращення процесу виявлення та прогнозування збоїв у системах Інтернету речей за допомогою розробленого методу з використанням комбінованої архітектури машинного навчання на основі графових нейронних мереж та довгої короткочасної пам’яті.

*Завдання дослідження:*

1. Провести теоретичний аналіз сучасних підходів до забезпечення надійності IoT-мереж.
2. Оцінити існуючі методи прогнозування збоїв і виявлення аномалій, визначити їх переваги та обмеження.
3. Розробити структуру адаптивного методу прогнозування з урахуванням просторово-часових залежностей між вузлами.
4. Провести експериментальні дослідження з використанням реальних або емульованих IoT-даних.
5. Порівняти ефективність запропонованого методу з базовими моделями.
6. Сформулювати практичні рекомендації щодо впровадження методу в системи моніторингу IoT-інфраструктури.

*Об’єкт дослідження:* процес функціонування мереж Інтернету речей у контексті забезпечення їхньої надійності та безперебійності роботи.

*Предмет дослідження:* методи та моделі прогнозування збоїв у IoT-мережах з використанням комбінованих алгоритмів машинного навчання.

*Методи дослідження.*

У роботі застосовуються методи системного аналізу, статистичного моделювання, машинного навчання, методи аналізу часових рядів.

*Наукова новизна отриманих результатів.*

У роботі запропоновано новий адаптивний метод прогнозування збоїв LGFP, який об’єднує графові та рекурентні нейронні мережі для спільного аналізу просторово-часових залежностей у телеметричних даних IoT. Такий підхід дозволяє підвищити точність прогнозування на 5-10% у порівнянні з класичними моделями за рахунок врахування топологічних зв’язків між вузлами.

*Практичне значення результатів.*

Розроблений метод може бути інтегрований у системи моніторингу IoT-платформ для своєчасного виявлення потенційних відмов, оптимізації технічного обслуговування, зменшення простоїв та підвищення надійності мережевих сервісів. Отримані результати можуть бути використані в галузях промислового Інтернету речей, розумних міст, систем «розумний дім» та розподілених сенсорних мереж.

*Апробація результатів та публікації*

1 Чугреєв К. О., Волощук О.Б. Метод прогнозування збоїв для мереж Інтернету речей за допомогою машинного навчання. Кібербезпека: освіта, наука, техніка. 2025. № 3 (31).

2 Чугреєв К. О. Порівняння методів машинного навчання для прогнозування збоїв в розумному будинку : теза доповіді / К. О. Чугреєв // VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», 15 квітня 2025 р. с.213-215, URL: https://duikt.edu.ua/uploads/p_2779_40288420.pdf

3 Чугреєв К. О. Метод на основі гібридного механізму LSTM та GNN для прогнозування збоїв у мережах Інтернету речей: теза доповіді / К. О. Чугреєв // III Міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп’ютерній інженерії», 4-6 грудня 2025р

# 1 ТЕОРЕТИКО-МЕТОДИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ЗБОЇВ У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

## 1.1 Сутність та особливості функціонування мереж Інтернету речей

У сучасному світі концепція Інтернету речей (Internet of Things, IoT) посідає центральне місце серед технологій цифрової трансформації. Вона передбачає створення інтегрованого середовища, у якому фізичні об’єкти - сенсори, контролери, виконавчі пристрої, промислове обладнання - здатні обмінюватися даними між собою та з хмарними сервісами без безпосередньої участі людини. Завдяки цьому формується глобальний інформаційний простір, що поєднує обчислювальні, комунікаційні та аналітичні можливості для забезпечення інтелектуальної взаємодії між об’єктами навколишнього середовища.

Інтернет речей можна розглядати як розподілену кіберфізичну систему, у якій фізичні компоненти тісно інтегровані з програмними, а процеси вимірювання, передавання та оброблення даних відбуваються у реальному часі. Така система дає змогу автоматизувати виробничі, логістичні, енергетичні, транспортні та побутові процеси, підвищуючи ефективність і безпеку їх функціонування.

Основою будь-якої IoT-мережі є багаторівнева архітектура, що зазвичай складається з таких компонентів:

Сенсорний рівень (Perception Layer) - включає фізичні пристрої збору даних: температурні, світлові, вологісні, рухові сенсори, RFID-теги, відеокамери тощо. Їхнє завдання полягає у сприйнятті стану навколишнього середовища та перетворенні фізичних параметрів у цифрові сигнали.

Мережевий рівень (Network Layer) - забезпечує передавання даних від сенсорів до вузлів оброблення за допомогою протоколів передачі: Wi-Fi, Bluetooth, ZigBee, LoRaWAN, 5G, Ethernet тощо. На цьому рівні важливими аспектами є затримка, пропускна здатність і стійкість з’єднань.

Рівень оброблення (Processing Layer) - відповідає за зберігання, попередню фільтрацію та аналітику даних. У сучасних IoT-системах цей рівень реалізується через хмарні обчислення або розподілену обробку на периферії (edge computing, fog computing).

Рівень застосунків (Application Layer) - надає сервіси кінцевим користувачам: моніторинг, управління, візуалізація, прогнозування тощо.

Така структура забезпечує ієрархічну взаємодію компонентів і дозволяє гнучко масштабувати систему залежно від кількості пристроїв і обсягів даних. Однак водночас вона створює низку проблем, пов’язаних із синхронізацією, надійністю з’єднань і виявленням відмов у розподіленому середовищі.

На відміну від традиційних комп’ютерних систем, мережі Інтернету речей характеризуються низкою властивостей, які ускладнюють їх адміністрування та підвищують ризик збоїв.

По-перше, мережі мають бути масштабованими та працювати з дуже різноманітними пристроями. В одній системі можуть бути тисячі продуктів різних виробників, кожен з яких має свої власні ресурси, протоколи зв'язку та формати даних. Така різноманітність ускладнює створення єдиних процедур для моніторингу та прогнозування несправностей.

По-друге, мережі IoT мають динамічну структуру - пристрої можуть постійно підключатися або відключатися від неї. Це означає, що зв'язки між вузлами мережі постійно змінюються, і навіть невеликий збій в одному місці може спричинити серйозні проблеми в усій системі, поширюючись як лавина.

Інша серйозна проблема - це обмеженість ресурсів. Багато сенсорів живуть від акумуляторів, мають мало пам'яті та невелику обчислювальну потужність. Через це на самих пристроях неможливо використовувати складні алгоритми для аналізу та контролю.

Також IoT-системи часто є критично важливими, наприклад, в медицині чи на транспорті. Тому вони дуже чутливі до навіть невеликих затримок у передачі даних або їх втрати. Навіть коротке переривання зв'язку може призвести до повного відмови всієї системи.

Крім того, IoT-мережі безперервно генерують величезні потоки телеметричних даних у реальному часі. Цю інформацію потрібно обробляти миттєво, щоб система могла вчасно реагувати на будь-які відхилення в роботі.

Кожна з перелічених властивостей прямо впливає на стратегії управління надійністю та визначає потребу у створенні адаптивних методів прогнозування можливих збоїв.

Важливою особливістю IoT є різноманіття комунікаційних протоколів, що використовуються для передавання даних між вузлами. Найпоширеніші описані в таблиці 1.1

Таблиця 1.1

Протоколи комунікації в Інтернеті речей

| Назва протоколу | Основні характеристики | Типові сценарії використання |
|---|---|---|
| MQTT | Легкий протокол обміну повідомленнями за моделлю «видавець-підписник». Використовує центральний брокер. Економний щодо трафіку та енергії. | Сенсорні мережі, дистанційний моніторинг, розумний дім, передача даних у хмару. |
| CoAP | Спрощений протокол для обмежених пристроїв. Працює поверх UDP, використовує модель запит-відповідь, подібну до HTTP. | Системи з низьким енергоспоживанням, обмежені мережі (LPWAN). |
| AMQP | Промисловий протокол для надійних транзакцій. Забезпечує гарантовану доставку повідомлень, чергування та безпеку. | Фінансові системи, промислова автоматизація, складні бізнес-процеси. |
| HTTP/HTTPS | Універсальний протокол Інтернету за моделлю запит-відповідь. HTTPS забезпечує шифрування. Може бути неефективним для дрібних даних. | Веб-інтерфейси для керування пристроями, завантаження оновлень прошивки, API для інтеграції з хмарними сервісами. |

| Назва протоколу | Основні характеристики | Типові сценарії використання |
|---|---|---|
| WebSocket | Протокол двостороннього зв'язку (full-duplex) поверх TCP. Встановлює постійне з'єднання, що дозволяє серверу та клієнту обмінюватись даними в реальному часі. | Потоковий моніторинг даних, інтерактивне керування пристроями, панелі керування в реальному часі. |
| gRPC | Високопродуктивний фреймворк для викликів процедур. Використовує HTTP/2 та бінарний формат Protocol Buffers. Ефективний для зв'язку між мікросервісами. | Швидкий обмін даними між потужними компонентами системи (наприклад, штучний інтелект, аналітика), зв'язок у розподілених архітектурах. |

Кожен протокол має свої переваги та недоліки щодо швидкості, надійності, безпеки та споживання енергії. У реальних умовах IoT-системи зазвичай використовують гібридні комунікаційні схеми, що поєднують кілька протоколів одночасно. Це підвищує гнучкість, але створює складність у моніторингу стану мережі, що є важливою передумовою для побудови моделей прогнозування збоїв.

Через величезні обсяги телеметрії, які надходять від численних сенсорів, традиційні централізовані системи оброблення даних втрачають ефективність. Тому дедалі більшого значення набувають периферійні (edge) та туманні (fog) обчислення, що дозволяють виконувати аналітику ближче до джерела даних.

У таких сценаріях дані проходять кілька етапів:

1. Попередня обробка на сенсорі або шлюзі - фільтрація шумів, нормалізація, локальне зберігання.
2. Агрегація даних на рівні fog-сервера - формування пакетів статистики, виявлення локальних відмов.
3. Передача до хмари (cloud) - для глобального аналізу, прогнозування та прийняття управлінських рішень.

Така архітектура зменшує затримки та трафік у каналах зв’язку, але призводить до фрагментації інформаційних потоків. Унаслідок цього збір повних

даних про стан системи стає складним, що вимагає створення інтелектуальних алгоритмів, здатних прогнозувати збої навіть за неповними або шумними даними.

Інтенсивне зростання кількості пристроїв призводить до появи нових загроз для стабільності роботи IoT-середовища. Найпоширенішими є:

- втрата з’єднання або перевантаження каналу;
- збої через енергетичні обмеження пристроїв;
- збої у програмному забезпеченні вузлів;
- порушення синхронізації даних між компонентами;
- аномальні стани внаслідок зовнішніх впливів (температура, вібрація, електромагнітні завади).

Через це особливого значення набуває прогнозне управління надійністю - підхід, який передбачає не просто виявлення збоїв, а їх передбачення до моменту виникнення на основі аналізу динаміки параметрів мережі.

Мережі Інтернету речей є складними розподіленими системами, які поєднують апаратні, програмні та комунікаційні компоненти. Їх ефективне функціонування залежить від здатності системи своєчасно реагувати на відмови та підтримувати стійкість у змінних умовах. Висока гетерогенність, динамічність топології, обмежені ресурси та нестабільність каналів зв’язку створюють передумови для збоїв, які неможливо повністю усунути традиційними методами моніторингу. Отже, постає необхідність розроблення інтелектуальних моделей прогнозування збоїв, здатних аналізувати поведінку вузлів у реальному часі та враховувати як часові, так і просторові залежності між елементами мережі. Ці питання становлять основу подальшого дослідження у наступних підрозділах.

## 1.2 Надійність і відмовостійкість у системах Інтернету речей

Одним із ключових викликів у розвитку сучасних систем Інтернету речей є забезпечення високої надійності та відмовостійкості функціонування розподілених

вузлів, сенсорів і мережевої інфраструктури. На відміну від традиційних обчислювальних систем, де більшість процесів контролюється централізовано, IoT характеризується масштабною децентралізацією, гетерогенністю компонентів та непередбачуваними зовнішніми умовами функціонування. Це призводить до зростання кількості точок потенційного відмовлення, що у свою чергу актуалізує потребу в створенні адаптивних механізмів прогнозування, діагностики та запобігання збоям.

Поняття надійності у класичній теорії систем визначається як властивість об’єкта зберігати працездатність упродовж заданого часу за визначених умов експлуатації. Проте в контексті Інтернету речей це визначення набуває розширеного змісту. Надійність IoT-системи включає не лише фізичну справність пристроїв, а й цілісність переданих даних, стабільність комунікаційних каналів, безперервність обчислювальних сервісів і коректність алгоритмів прийняття рішень.

Складність полягає у тому, що навіть короткочасний збій одного сенсора або шлюзу може спричинити каскадну реакцію, що призведе до втрати даних або зниження точності системи в цілому. Наприклад, у розумних енергомережах помилка вимірювання струму в одному вузлі може викликати помилкову реакцію контролера розподілу навантаження, створюючи ланцюговий ефект відмови. Таким чином, надійність в IoT розглядається як інтегральна властивість усієї екосистеми, а не окремого елемента.

Рівень надійності залежить від низки взаємопов’язаних чинників. До апаратних належать якість елементної бази, типи сенсорів, живлення, захищеність пристроїв від фізичних впливів. До програмних - стабільність мікропрограмного забезпечення, коректність алгоритмів обробки даних, стійкість протоколів до збоїв зв’язку. Важливою складовою є також мережева архітектура: топологія, протоколи маршрутизації, рівень резервування каналів.

У багатьох випадках саме взаємодія цих чинників формує критичні зони ризику. Наприклад, застосування малопотужних бездротових сенсорів у промислових IoT-середовищах підвищує ймовірність втрат пакетів через

радіоперешкоди, що впливає на достовірність зібраної інформації. Одночасно, використання нестійких протоколів типу MQTT без належної перевірки цілісності повідомлень може призвести до спотворення даних без явних ознак відмови. Таким чином, для досягнення цільового рівня надійності необхідна комплексна оптимізація усіх рівнів системи - від фізичного до прикладного.

Відмовостійкість визначається як здатність системи продовжувати коректне функціонування навіть у разі часткових відмов її компонентів. У системах IoT цей принцип має вирішальне значення, оскільки кількість вузлів може сягати тисяч і навіть мільйонів, а відмови окремих сенсорів - неминучі.

Традиційно відмовостійкість досягається шляхом дублювання критичних компонентів, використання резервних каналів зв'язку та алгоритмів самовідновлення. Проте класичні підходи виявляються малоефективними в IoT через обмежені енергетичні ресурси, нестачу обчислювальної потужності та динамічність топології. Тому все більшої популярності набувають методи прогностичної відмовостійкості - коли система завчасно ідентифікує ознаки потенційної відмови й адаптує свою поведінку для мінімізації наслідків.

Одним із прикладів є використання розподілених агентних моделей, у яких вузли мережі обмінюються інформацією про власний стан, створюючи колективну систему діагностики. Інший напрям - застосування машинного навчання для виявлення аномалій у потоках телеметрії, що дозволяє передбачити збій до моменту його фактичного прояву. Такі підходи формують основу нової парадигми відмовостійкості, заснованої не на пасивному резервуванні, а на активному прогнозуванні та самовідновленні.

У науковій літературі розрізняють кілька основних класів моделей, які використовуються для аналізу надійності систем: стохастичні, імовірнісні, марковські та моделі на основі теорії нечітких множин. Для систем IoT ці підходи зазнають суттєвих модифікацій, оскільки необхідно враховувати неоднорідність вузлів, нерівномірність потоків даних і змінність навантаження.

Стохастичні моделі дозволяють оцінити середній час безвідмовної роботи (MTBF) і ймовірність відновлення працездатності після збою. Марковські ланцюги

застосовуються для моделювання процесів відмов і відновлень у розподілених сенсорних мережах, особливо коли кожен вузол має кілька станів працездатності. Нечіткі моделі, у свою чергу, ефективні в умовах невизначеності, коли точні статистичні параметри невідомі.

Сучасна тенденція полягає у поєднанні класичних імовірнісних моделей із методами машинного навчання, що дозволяє створювати адаптивні гібридні системи прогнозування. Такі системи навчаються на історичних даних про відмови, аналізують їх кореляції з умовами експлуатації та будують предиктивні моделі, здатні працювати в реальному часі. Саме цей підхід лежить в основі розробки нового методу прогнозування, який буде запропонований у третьому розділі даної роботи.

Одним із найефективніших шляхів підвищення надійності є створення багаторівневої архітектури IoT, що включає механізми контролю на кожному рівні:

- на рівні сенсорів - калібрування, самодіагностику, локальне кешування даних;
- на рівні шлюзів - перевірку цілісності повідомлень, балансування навантаження, дублювання маршрутів;
- на рівні хмарних або edge-сервісів - аналітику відмов, централізований моніторинг та автоматичне відновлення сервісів.

Такі підходи дозволяють забезпечити принцип “graceful degradation” - тобто поступове зниження якості обслуговування без повного зупинення системи.

Використання технологій edge computing додатково підвищує відмовостійкість, адже дозволяє виконувати критичну аналітику поблизу джерела даних, мінімізуючи залежність від центральних серверів. Це особливо важливо для застосувань із жорсткими вимогами до часу відгуку, таких як автономний транспорт, медичні моніторингові системи чи промислова автоматизація.

Отже, забезпечення надійності та відмовостійкості в системах Інтернету речей є багатофакторною проблемою, що охоплює апаратні, програмні, мережеві та алгоритмічні аспекти. Традиційні методи резервування та реактивного відновлення стають недостатніми в умовах масштабності та динамічності IoT.

Сучасні підходи зміщуються в бік прогностичних моделей на основі штучного інтелекту, які здатні виявляти приховані ознаки деградації системи та забезпечувати превентивне реагування.

Таким чином, відмовостійкість у нових поколіннях IoT-систем розглядається як динамічна властивість, що формується за рахунок інтеграції аналітичних, когнітивних та самонавчальних компонентів. Подальші дослідження у цій сфері спрямовані на розробку універсальних моделей прогнозування, які зможуть адаптуватися до різних сценаріїв використання без потреби у жорсткому налаштуванні параметрів.

## 1.3 Аналіз сучасних алгоритмів машинного навчання у прогнозуванні збоїв у системах Інтернету речей

У сучасних умовах інтенсивного розвитку Інтернету речей (IoT) традиційні статистичні методи аналізу даних виявляються недостатніми для ефективного прогнозування відмов і збоїв. Потоки даних у таких системах характеризуються великим обсягом, високою швидкістю надходження, неоднорідністю структур і сильними кореляційними зв’язками між сенсорними вузлами. Ці особливості зумовлюють перехід до застосування методів машинного навчання здатних автоматично виявляти складні закономірності у даних і формувати предиктивні моделі високої точності.

Сьогодні у сфері прогнозування збоїв у складних кіберфізичних системах активно застосовуються різні підходи ML - від класичних алгоритмів класифікації до глибоких нейронних архітектур, що використовують механізми пам’яті, уваги та графового представлення даних. Нижче розглянемо основні з них - SVM, Random Forest, LSTM, GNN, Autoencoder, а також коротко окреслимо їх сильні та слабкі сторони у контексті задачі прогнозування відмов у середовищі IoT.

SVM є одним із найстаріших і найстабільніших алгоритмів у задачах бінарної класифікації, який залишається актуальним завдяки здатності формувати оптимальну гіперплощину розділення між класами навіть у високовимірних просторах. Його математична основа полягає у максимізації відстані між класами даних, що мінімізує ризик помилки узагальнення.

У контексті прогнозування збоїв у IoT SVM застосовується для виявлення аномальних станів системи на основі історичних даних про роботу сенсорів. Наприклад, у системах моніторингу промислового обладнання SVM може розрізняти “нормальні” та “аномальні” режими вібраційних сигналів. Для цього дані попередньо обробляються з використанням методів нормалізації, виділення ознак (feature extraction) та перетворення часових рядів у векторні представлення.

Основною перевагою SVM є стійкість до перенавчання на малих вибірках і здатність працювати із нерівноважними класами, що характерно для IoT, де кількість нормальних станів суттєво перевищує кількість збоїв. Проте SVM має обмеження щодо масштабованості: при зростанні кількості сенсорів і обсягів даних експоненційно збільшується час навчання, що знижує його придатність для реального часу.

Random Forest - ансамблевий метод, який поєднує велику кількість незалежних рішень дерев класифікації (Decision Trees). Кожне дерево навчається на випадковій підмножині даних і ознак, а кінцеве рішення приймається більшістю голосів. Така архітектура забезпечує високу узагальнювальну здатність і низьку чутливість до шуму, що є надзвичайно важливим у контексті IoT, де дані часто містять похибки або пропуски.

У задачах прогнозування відмов RF використовується для ранжування важливості ознак, що дозволяє ідентифікувати ключові параметри, які найсильніше впливають на появу збоїв. Наприклад, у розумних системах енергомоніторингу RF може показати, що температурні коливання чи напруга живлення мають найбільший вплив на ймовірність деградації сенсорних модулів.

Переваги Random Forest полягають у простоті реалізації, високій інтерпретованості результатів і можливості паралельної обробки даних. Недоліком

є неможливість моделювати часову залежність - RF аналізує кожен запис незалежно, тому не підходить для глибокого аналізу часових процесів без попереднього агрегування ознак. Саме тому цей алгоритм часто комбінують із рекурентними нейронними мережами або методами Sliding Window.

LSTM - це тип рекурентних нейронних мереж (RNN), спеціально створений для роботи з часовими рядами. Його архітектура дозволяє зберігати інформацію про попередні стани системи через спеціальні елементи - «вікна пам’яті» (memory cells), що контролюють потік інформації за допомогою сигмоїдних «воріт» (input, forget, output gates). Завдяки цьому LSTM здатна моделювати довготривалі залежності в даних, що є критично важливим для IoT, де збій часто має кумулятивний характер і формується внаслідок поступової деградації компонентів.

У сфері прогнозування збоїв LSTM демонструє високу точність при аналізі послідовностей даних, наприклад, змін температури, тиску чи вібрацій у промислових сенсорах. Мережа може навчитися відрізняти природні коливання від патернів, які передують відмові. Крім того, LSTM добре інтегрується з потоковими обчисленнями, що дозволяє її використання у системах реального часу.

Головним обмеженням LSTM є висока обчислювальна складність та потреба у великих навчальних вибірках, що може бути проблемою для енергообмежених IoT-пристроїв. У зв’язку з цим останні дослідження пропонують спрощені версії LSTM наприклад, GRU - Gated Recurrent Unit або гібридні підходи, де частина аналітики виконується на рівні edge-серверів, а не безпосередньо на пристроях.

Системи IoT мають природну графову структуру: сенсори, контролери й шлюзи можна розглядати як вузли графа, а зв’язки між ними - як ребра, що передають дані. Тому логічним є застосування графових нейронних мереж (GNN), які дозволяють враховувати не лише локальні характеристики кожного вузла, а й топологічний контекст його оточення.

GNN моделюють взаємодію між вузлами через операції агрегування та оновлення станів, що дозволяє ефективно виявляти аномальні підграфи або локальні порушення структури мережі, які можуть свідчити про потенційні збої. У практиці GNN застосовуються для прогнозування перевантажень у транспортних

мережах, визначення слабких вузлів у сенсорних кластерах та діагностики несправностей у розподілених обчислювальних системах.

Перевага GNN - здатність враховувати просторові залежності між пристроями, чого не можуть зробити класичні послідовні моделі. Однак навчання GNN є ресурсомістким і вимагає глибоких знань про топологію мережі. Тому перспективним напрямом є розробка спрощених або динамічних графових моделей, які автоматично перебудовуються при зміні конфігурації IoT-системи.

Окрему нішу займають автоенкодери - безнаглядові моделі, що навчаються відтворювати вхідні дані через зменшений латентний простір. Вони ефективно використовуються для виявлення аномалій: якщо мережа не здатна якісно реконструювати нові дані, це свідчить про їх невідповідність звичним патернам, тобто про можливу відмову.

У комбінації з LSTM автоенкодери формують послідовні автоенкодери (LSTM-AE), що аналізують часові ряди у латентному просторі. Такі моделі забезпечують високу точність виявлення відхилень у складних динамічних процесах - наприклад, у роботі енергетичних чи транспортних IoT-систем.

Сучасні дослідження демонструють ефективність гібридних підходів, які поєднують кілька моделей - наприклад, Random Forest для відбору ознак і LSTM для прогнозування; або GNN для виявлення просторових залежностей і Autoencoder для детекції аномалій. Такі системи мають вищу адаптивність і краще узагальнюють знання у реальних умовах експлуатації.

Проведений аналіз (таб.1.1) свідчить, що жоден окремий алгоритм машинного навчання не здатен повністю вирішити задачу прогнозування збоїв у складних, динамічних і гетерогенних системах Інтернету речей. Найбільш перспективним напрямом розвитку є гібридизація методів, коли поєднуються сильні сторони різних моделей - наприклад, структурна інформативність GNN, часовий контекст LSTM і здатність Autoencoder виявляти приховані аномалії.

Ключовою тенденцією є також перехід до пояснюваного машинного навчання (Explainable AI), що дозволяє не лише прогнозувати збій, а й пояснювати, які фактори до нього призводять. Це відкриває шлях до побудови прогностично-

адаптивних IoT-систем, які не просто реагують на проблеми, а активно попереджають їх виникнення.

Таблиця 1.2

Сучасні алгоритми машинного навчання

| Алгоритм | Переваги | Недоліки | Найкраще застосування |
|---|---|---|---|
| SVM | Висока точність при малих вибірках, стійкість до шуму | Погана масштабованість, складність налаштування ядра | Локальні системи моніторингу |
| Random Forest | Інтерпретованість, стійкість до викидів | Не враховує часові залежності | Аналіз історичних відмов |
| LSTM | Обробка часових рядів, глибокий контекст | Висока складність, потребує багато даних | Прогнозування деградаційних процесів |
| GNN | Моделює просторові залежності | Ресурсоємність, потреба в графовій структурі | Розподілені сенсорні мережі |
| Autoencoder | Безнаглядове навчання, детекція аномалій | Може пропускати “складні” збої | Виявлення прихованих аномалій |

# 2 ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЗБОЇВ ТА АНАЛІЗ ВХІДНИХ ДАНИХ

## 2.1 Характеристика об’єкта дослідження та даних для моделювання

Сучасні системи Інтернету речей (Internet of Things, IoT) являють собою складні розподілені інфраструктури, що поєднують тисячі або навіть мільйони пристроїв, сенсорів, шлюзів, серверів обробки даних та хмарних компонентів. У межах даного дослідження об’єктом виступає інтелектуальна IoT-мережа моніторингу промислового середовища, яка інтегрує фізичні сенсори з аналітичними сервісами машинного навчання для забезпечення стабільного функціонування виробничого обладнання.

Метою аналізу є побудова узагальненої моделі даних, що відображає динаміку роботи вузлів IoT-мережі та дозволяє виявити закономірності, пов’язані з виникненням збоїв, затримок або відмов у її компонентах. Розглянута система є типовим представником промислових IoT-рішень, які характеризуються підвищеними вимогами до надійності, пропускної здатності та енергоефективності.

Архітектура досліджуваної мережі побудована за трирівневою структурою, типовою для промислових IoT-систем:

1. Периферійний рівень (Edge Layer) - складається з численних сенсорних вузлів, що виконують безпосередні вимірювання фізичних параметрів: температури, вологості, вібрацій, споживання енергії, рівня шуму, а також інформацію про стан обладнання (наприклад, швидкість обертання двигуна, тиск, струм). Кожен вузол має обмежені обчислювальні ресурси й автономне живлення.
2. Шлюзовий рівень (Gateway Layer) - забезпечує агрегацію даних від групи сенсорів і передає їх у хмару через протоколи MQTT або CoAP. На цьому рівні реалізуються локальні механізми фільтрації даних, усунення шумів і попереднього виявлення аномалій.

3. Хмарний рівень (Cloud Layer) - відповідає за централізовану обробку, збереження та аналіз даних. У хмарному середовищі функціонують модулі аналітики та прогнозування, засновані на алгоритмах машинного навчання, а також системи сповіщення операторів про потенційні відмови.

Інфраструктура досліджуваної IoT-системи складається з понад 2000 сенсорних вузлів, об’єднаних у 15 локальних сегментів, що взаємодіють через бездротові протоколи ZigBee, LoRaWAN і Wi-Fi. Дані збираються з періодичністю одна хвилина, формуючи близько 3 млн сирих записів на добу, з яких після агрегування та очищення за дворічний період залишено близько 1,2 млн унікальних спостережень, використаних для статистичного аналізу та подальшого навчання моделей.

Для адекватного прогнозування збоїв у мережі IoT важливо враховувати не лише показники навколишнього середовища, а й технічні параметри роботи вузлів, що відображають їх навантаження та стабільність. У дослідженні розглядаються групи ознак наведені в таблиці 2.1

Таблиця 2.1

Групи ознак для прогнозування збоїв у мережі IoT

| Група ознак | Опис | Параметри |
|---|---|---|
| Операційні показники вузлів | Показники, що характеризують фізичний стан та продуктивність IoT-вузла, його енергоресурс та ефективність передачі даних. | • Рівень заряду акумулятора<br>• Середнє енергоспоживання за період<br>• Температура процесора<br>• Кількість переданих/втрачених пакетів<br>• Середня затримка при передачі даних |
| Мережеві показники | Параметри, що визначають якість мережевого з'єднання, стабільність зв'язку та наявність перешкод у передачі даних. | • Інтенсивність трафіку<br>• RSSI (рівень сигналу)<br>• Частота повторних передач<br>• Середній час з’єднання з шлюзом<br>• Частка помилкових пакетів CRC |

| Група ознак | Опис | Параметри |
|---|---|---|
| Контекстні дані | Зовнішні фактори та метадані, що впливають на роботу мережі, але не є прямими технічними параметрами вузла чи зв'язку. | • Тип обладнання, на якому розміщений сенсор<br>• Тип середовища (приміщення, відкрита зона, промислова ділянка)<br>• Часові ознаки (день/ніч, робоча/вихідна зміна)<br>• Температура та вологість навколишнього середовища |

Для моделювання відмов сформовано узагальнену ознакову матрицю розміром N × M, де *N* - кількість записів (спостережень), *M* - кількість ознак (параметрів). Кожен запис супроводжується міткою стану: *0* - нормальне функціонування, *1* - збій (виявлений або підтверджений).

Інформаційна база дослідження формувалася на основі реальних і симульованих даних.

Реальні дані, отримані з тестової платформи IoT-інфраструктури підприємства, що спеціалізується на енергомоніторингу та технічному обслуговуванні виробничих ліній. Система охоплює понад 2000 сенсорних вузлів, об’єднаних у 15 локальних сегментів, які взаємодіють через бездротові протоколи ZigBee, LoRaWAN і Wi-Fi. Збирання даних здійснюється з періодичністю один раз на хвилину, що формує близько 3 млн записів на добу. Для моделювання використовувалися агреговані та очищені дані за період двох років, які містять близько 1,2 млн унікальних записів після попередньої обробки.

Симульовані дані створені для доповнення вибірки з використанням методів стохастичного моделювання, зокрема генерації синтетичних аномалій на основі розподілу Пуассона та гаусівських шумів, що дозволило сформувати різноманітні сценарії поведінки вузлів при збоях у зв’язку, перегріві або зниженні напруги.

Для підвищення якості та ефективності моделей машинного навчання всі вихідні дані були піддані попередній обробці. Цей етап дозволив усунути шум,

нормалізувати масштаби ознак та вирішити проблему несбалансованості набору даних. В таблиці 2.2 наведено детальний опис застосованих методів.

Таблиця 2.2

Групи ознак для прогнозування збоїв у мережі IoT

| Етап обробки | Мета застосування |
|---|---|
| Фільтрація викидів | Виявлення та усунення аномальних значень, що можуть спотворити результати аналізу. |
| Нормалізація ознак | Приведення числових ознак до єдиного масштабу для забезпечення коректної роботи алгоритмів. |
| Кодування категоріальних змінних | Перетворення текстових та категоріальних даних у числовий формат, зрозумілий для моделей. |
| Балансування класів | Усунення дисбалансу між кількістю збоїв і штатних подій для покращення якості навчання. |

Для автоматизації процесу моніторингу було формалізовано критерії виявлення збоїв. Всі інциденти класифікуються за трьома основними категоріями, для кожної з яких визначено набір ключових ознак-індикаторів (таб.2.3). Ці правила дозволяють ідентифікувати потенційні проблеми на ранній стадії.

Таблиця 2.3

Класифікація збоїв та їх ключові індикатори

| Категорія збою | Опис та характерні причини | Приклади ознак-індикаторів |
|---|---|---|
| Апаратні збої | Вихід з ладу фізичних компонентів вузла: сенсора, акумулятора, процесора. | • temp_cpu > 80°C<br>• battery_level < 15%<br>• Різке падіння рівня сигналу за відсутності мережевих проблем. |
| Мережеві збої | Проблеми з передачею даних, що виникають через нестабільність зв'язку або перевантаження мережі. | • packet_loss_rate > 0.15 (15%)<br>• RSSI < -80 dBm<br>• Надмірна затримка передачі (latency) протягом тривалого часу |

| Категорія збою | Опис та характерні причини | Приклади ознак-індикаторів |
|---|---|---|
| Програмні збої | Несправності, спричинені помилками в програмному забезпеченні або прошивці вузла. | reboot_count > 3 (за останню годину)<br>• Зависання сенсора без відправки даних.<br>• Некоректні оновлення прошивки. |

Завдяки такій системі маркування вдалося створити еталонну базу подій, у якій кожен запис має конкретну причину збоїв. Це забезпечує можливість супервізованого навчання моделей машинного навчання, описаних у подальших підрозділах.

Після попередньої обробки сформовано узагальнений набір даних, що наведено в таблиці 2.4

Таблиця 2.4

Класифікація даних та їх типи

| Категорія | Приклади полів | Тип даних |
|---|---|---|
| Ідентифікаційні | • device_id,<br>• gateway_id<br>• region_code | Цілочисельний / текстовий |
| Часові | • Timestamp<br>• hour_of_day<br>• day_of_week | Дата / час |
| Експлуатаційні | • battery_level<br>• cpu_temp<br>• energy_consumption | Дійсні |
| Мережеві | • RSSI<br>• packet_loss_rate<br>• latency<br>• retransmission_rate | Дійсні |
| Контекстні | • device_type<br>• location_type | Категоріальні |
| Цільова змінна | • failure_state (0/1) | Бінарна |

Розмір підготовленої навчальної вибірки становить близько 1,2 млн рядків після агрегації даних за часовими вікнами тривалістю 5 хвилин. Для кожного вузла мережі формується окрема часово-залежна серія, що дозволяє застосовувати рекурентні нейронні мережі або графові моделі, які враховують топологічні взаємозв’язки між пристроями.

Таким чином, об’єктом дослідження є багаторівнева промислова IoT-система з високою частотою передачі даних і складною структурою взаємодій між компонентами. Зібрана інформаційна база містить широкий спектр параметрів, що відображають як фізичні характеристики обладнання, так і мережеві показники продуктивності. Створений масив даних придатний для побудови інтелектуальної моделі прогнозування збоїв, здатної виявляти приховані закономірності та попереджати потенційні відмови в режимі реального часу.

## 2.2 Аналіз структури даних та попередня статистична обробка

Одним із ключових етапів розробки методу прогнозування збоїв у мережах Інтернету речей є детальний аналіз структури зібраних даних, виявлення закономірностей у їх розподілі, визначення кореляційних зв’язків між параметрами та усунення інформаційних шумів. На цьому етапі формується базовий аналітичний профіль вибірки, який дозволяє оцінити якість, повноту та репрезентативність даних, а також підготувати їх для подальшого моделювання за допомогою алгоритмів машинного навчання.

Після етапу агрегації й очищення сформовано інтегрований набір даних обсягом понад 1,2 мільйона записів, що охоплює дворічний період функціонування IoT-системи. Дані мають часову природу: кожен запис відповідає стану сенсорного вузла у конкретний момент часу. Загалом у структурі набору даних виділено 42 ознаки, які умовно поділяються на кілька груп (таб.2.5)

Таблиця 2.5

Перелік аналізованих ознак

| **Група ознак** | **Конкретні параметри** |
|---|---|
| Енергетичні показники | • battery_level<br>• energy_consumption<br>• voltage_drop |
| Температурні параметри | • cpu_temp<br>• ambient_temp<br>• humidity |
| Мережеві характеристики | • RSSI<br>• packet_loss_rate<br>• latency<br>• retransmission_rate<br>• signal_quality |
| Експлуатаційні метрики | • uptime<br>• reboot_count<br>• device_load<br>• memory_usage |
| Контекстні та часові ознаки | • device_type<br>• gateway_id<br>• hour_of_day<br>• day_of_week<br>• environment_type |
| Цільова змінна яка відображає факт збоїв у роботі вузла | • failure_state |

Структура даних є гетерогенною, тобто включає як числові так і категоріальні ознаки. Такий тип даних є типовим для складних кіберфізичних систем, де поєднуються фізичні показники, логічні стани та часові параметри.

На початковому етапі проведено оцінювання пропущених значень у кожній колонці таблиці. Виявлено, що близько 2,8% усіх записів містять неповні дані, що

переважно пов’язано з тимчасовими перебоями у зв’язку або розрядом батарей сенсорів. Для усунення цього ефекту застосовано комбінацію методів:

- Інтерполяція часових рядів (Time-based Linear Interpolation) для числових ознак, що мають безперервну природу, зокрема RSSI, battery_level, cpu_temp.
- Заповнення категоріальних пропусків методом most frequent value, тобто підставленням найчастішого значення серед сусідніх записів того ж пристрою.
- Видалення аномальних фрагментів (де відсутні більше ніж 30% ознак) для уникнення спотворення розподілів.

Результати первинної оцінки якості показали, що після обробки кількість повних рядків збільшилася до 98,7%, що є прийнятним рівнем для побудови моделей машинного навчання.

Унаслідок збоїв сенсорів або радіоперешкод у даних можуть з’являтися аномальні спостереження, які спотворюють статистичну картину. Для їх виявлення використано три підходи:

1. Метод міжквартильного розмаху (IQR) - усі значення, що виходили за межі [Q1 - 1.5×IQR; Q3 + 1.5×IQR], вважалися підозрілими.
2. Z-score нормалізація - спостереження зі стандартним відхиленням більше 3σ класифікувалися як потенційні викиди.
3. Модель локальних відхилень (Local Outlier Factor, LOF) - використовувалася для багатовимірного виявлення аномалій, коли кілька показників одночасно мають нетипові значення.

Загалом близько 1,7% спостережень були ідентифіковані як аномалії. Замість повного видалення їх замінили на лінійно інтерпольовані значення, щоб не втратити часову цілісність рядів.

Кореляційний аналіз проведено з метою виявлення сильних взаємозв’язків між ознаками та усунення надлишкової інформації. Для числових змінних використано коефіцієнт Пірсона, для категоріальних - Крамера (V).

Основні спостереження:

Сильна позитивна кореляція між battery_level і voltage_drop (r = 0.81), що дозволяє одну з ознак виключити.

Помітна негативна кореляція між RSSI і packet_loss_rate (r = -0.72), що підтверджує логічну залежність між силою сигналу та втратами пакетів.

Висока залежність latency від retransmission_rate (r = 0.65), що вказує на вплив перевантаження каналів на затримки.

На основі результатів аналізу сформовано зменшений набір ознак (feature subset) із 27 параметрів, який забезпечує найкраще співвідношення між інформативністю та обчислювальною складністю моделей. Для подальшої оцінки значущості ознак використано метод дерев рішень, що підтвердив домінуючу роль параметрів RSSI, latency, battery_level, cpu_temp та packet_loss_rate.

Через природу систем IoT частота збоїв зазвичай є низькою (менше 5% усіх подій). Це призводить до дисбалансу класів у цільовій змінній failure_state. Якщо навчати модель на таких даних без корекції, вона буде схильна переважно передбачати «нормальний» стан.

Для вирішення цієї проблеми застосовано двоетапний підхід:

Oversampling рідкісного класу методом SMOTE (Synthetic Minority Over-sampling Technique), який створює синтетичні приклади на основі існуючих зразків збійних станів.

Undersampling нормального класу для зменшення надлишку «здорових» записів без втрати різноманіття поведінки мережі.

У результаті співвідношення між класами стало близьким до 60:40, що суттєво підвищує стійкість моделі до перекосу.

Для алгоритмів, що враховують часову динаміку, сформовано ковзні часові вікна довжиною 10 хвилин із кроком у 5 хвилин. Це дозволяє відстежувати тренди зміни параметрів перед виникненням збоїв. Кожне вікно містить послідовність вимірювань для ключових змінних (RSSI, latency, battery_level, cpu_temp, packet_loss_rate), а цільова змінна визначається як стан системи у наступному часовому інтервалі (t+1).

Для збереження коректності залежностей усі дані розділено за принципом time-based split:

- 70% - для навчання моделей (перші 18 місяців),
- 15% - для валідації,
- 15% - для тестування (останні 3 місяці).

Такий підхід мінімізує ризик витоку майбутньої інформації та забезпечує реалістичну перевірку здатності моделей прогнозувати відмови.

Проведений аналіз структури даних дозволив сформувати чистий, збалансований і статистично стабільний набір, придатний для побудови моделей прогнозування збоїв у мережах IoT. Виконано глибоку попередню обробку: очищення, нормалізацію, видалення викидів, інтерполяцію пропусків і балансування класів. Сформовані часові послідовності дозволяють не лише класифікувати поточні стани, а й передбачати майбутні збої з урахуванням динаміки параметрів.

## 2.3 Аналіз частоти та причин збоїв у системах Інтернету речей

Забезпечення надійності систем Інтернету речей (IoT) є одним із найважливіших чинників, що визначають їхню ефективність та життєздатність у довготривалій експлуатації. У сучасних розподілених архітектурах, які охоплюють тисячі взаємопов’язаних пристроїв, навіть незначні збої можуть призвести до ланцюгових відмов, втрати даних або повної деградації функціональності. Тому аналіз частоти та причин таких збоїв становить ключову складову будь-якої моделі прогнозування й оцінювання ризиків у системах IoT.

Загалом, відмови в Інтернеті речей мають комплексну та багатоаспектну природу, що обумовлено сукупною дією низки технологічних, організаційних та людських факторів (таб.2.6)

Таблиця 2.6

Категорії причин відмов в IoT-мережах

| Категорія причини | Характерні приклади |
|---|---|
| Апаратні проблеми | Зношування сенсорів, деградація елементів живлення. |
| Програмні помилки | Збої драйверів, переповнення буферів, помилки в прошивці. |
| Мережеві фактори | Нестабільність з’єднання, перевантаження каналів зв'язку, втрати пакетів. |
| Людський фактор | Помилки конфігурації обладнання, недотримання протоколів безпеки. |

Усі події збоїв були класифіковані за типами, часом настання, тривалістю відновлення та контекстом (умови навколишнього середовища, навантаження, версія програмного забезпечення тощо). Для кожного пристрою розраховувався показник MTBF (Mean Time Between Failures) - середній час між відмовами, який дає змогу оцінити стабільність роботи конкретного вузла або підсистеми. Додатково враховувався MTTR (Mean Time To Repair), що характеризує швидкість відновлення після збоїв і, відповідно, ефективність процедур технічної підтримки.

На основі агрегованих даних було виявлено, що середній MTBF для периферійних сенсорів становив близько 920 годин, тоді як для шлюзових пристроїв - понад 2400 годин. Це підтверджує тезу про те, що найбільш уразливими елементами IoT-екосистеми є саме крайові вузли, які піддаються впливу температурних коливань, вологості, механічних вібрацій та енергетичних флуктуацій. Найчастіше відмови відбувалися внаслідок деградації акумуляторів та нестабільної передачі даних через перевантаження бездротових каналів, особливо у середовищах із високою щільністю пристроїв.

З позицій аналітики подій виявлено кілька характерних закономірностей. По-перше, більшість збоїв мають кластерний характер: вони не розподіляються рівномірно у часі, а концентруються навколо пікових періодів навантаження - наприклад, під час оновлення прошивок або інтенсивного зчитування даних. По-

друге, значна частина збоїв є вторинними, тобто спричиненими попередніми відмовами у суміжних модулях, що вказує на високу залежність компонентів системи. Такий ефект «ланцюгової реакції» підтверджує необхідність використання графових моделей (наприклад, Graph Neural Networks) для моделювання взаємозалежностей між вузлами.

Крім того, у процесі статистичного аналізу було визначено, що відмови, пов’язані з комунікаційними каналами, мають найвищу частоту (приблизно 42 % усіх зафіксованих подій), тоді як апаратні поломки складають близько 27 %, а програмні - близько 21 %. Решта (10 %) припадає на людський фактор і зовнішні впливи. Такий розподіл підтверджує, що головним «вузьким місцем» сучасних IoT-інфраструктур залишається саме мережевий рівень - зокрема, проблеми з QoS (Quality of Service) та нестабільністю бездротових протоколів у перевантажених умовах.

З метою виявлення потенційних залежностей між показниками середовища та частотою збоїв було проведено кореляційний аналіз між параметрами температури, вологості, напруги живлення, рівня сигналу (RSSI) і кількістю відмов за одиницю часу. Найвища кореляція ($r \approx 0.68$) спостерігалася між коливаннями напруги живлення та кількістю апаратних збоїв, що свідчить про доцільність інтеграції систем моніторингу енергопостачання у реальному часі. Значущою також виявилася залежність між підвищенням вологості понад 80 % і зростанням кількості комунікаційних помилок, що можна пояснити зниженням ефективності бездротових каналів у несприятливих погодних умовах.

Отримані результати стали основою для формування набору індикаторів раннього попередження, які у подальшому використовувались під час побудови моделей машинного навчання. Зокрема, було визначено, що зростання латентності передачі даних, зниження напруги живлення більш ніж на 5 % від номінальної, а також збільшення кількості пакетів повторної передачі є достовірними предикторами наближення відмови. Ці фактори були відібрані як ключові ознаки для подальшого моделювання за допомогою алгоритмів машинного навчання.

Додатково було проведено аналіз часових рядів частоти збоїв, що показав наявність сезонності та циклічності в поведінці системи. Наприклад, у літні місяці кількість збоїв сенсорів вологості зростала на 17-23 %, що пояснюється як екологічними факторами, так і зростанням інтенсивності використання систем. Це відкриває можливість застосування рекурентних нейронних мереж для виявлення довготривалих тенденцій і прогнозування періодів підвищеної ризикованості.

Таким чином, проведений аналіз дозволив не лише кількісно оцінити надійність компонентів IoT-системи, але й якісно описати природу відмов, визначити ключові чинники, що впливають на їхню частоту, та виділити групи параметрів, придатні для подальшого машинного моделювання. Отримані закономірності лягли в основу етапу побудови прогнозних моделей.

## 2.4 Вибір моделей машинного навчання, їх навчання та тестування

Застосування методів машинного навчання у сфері Інтернету речей передбачає необхідність вибору моделей, здатних ефективно працювати з великими обсягами телеметричних даних, що характеризуються високою варіативністю, часовою залежністю та потенційною наявністю шуму. У процесі побудови моделі прогнозування збоїв ключовим завданням є досягнення компромісу між точністю, інтерпретованістю результатів і швидкодією навчання.

Вибір моделей базувався на декількох концептуальних критеріях:

- масштабованість - можливість роботи з великими потоками даних IoT-систем у реальному часі;
- адаптивність - здатність моделі враховувати зміну статистичних властивостей середовища у часі;
- стійкість до шумів та пропусків - адже реальні телеметричні дані нерідко містять пропуски, аномалії або недостовірні значення;

- підтримка нелінійних взаємозв’язків між численними сенсорними параметрами;
- інтерпретованість результатів, необхідна для практичного використання в системах моніторингу та технічного обслуговування.

Для забезпечення відтворюваності експерименту було проведено формалізацію етапів підготовки та аналізу даних. На основі набору телеметричних даних виконано попередню обробку, стандартизацію та розділення вибірки на тренувальну (80 %) і тестову (20 %) частини. Для мінімізації впливу випадковості використовувалася k-кратна крос-валідація (k=5), коли дані багаторазово розбиваються на підвибірки, і кожна з них по черзі виконує роль тестової:

$$CV_{\text{score}} = \frac{1}{k}\sum_{i=1}^{k} M_i \tag{2.1}$$

де $M_i$ - метрика якості на $i$-й ітерації.

Вибір метрик оцінювання здійснювався відповідно до характеру задачі - бінарна класифікація з метою виявлення станів “норма/збій”. Основними використовуваними метриками були:

Accuracy (загальна точність):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

де TP - кількість правильно класифікованих випадків відмов, TN - правильно класифікованих нормальних станів, FP - помилкові спрацьовування, FN - пропущені збої.

Precision (точність) та Recall (повнота):

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

де Precision показує, яка частка передбачених відмов дійсно є істинними, а Recall - яку частку реальних збоїв вдалося виявити.

F1-score, що є гармонійним середнім між точністю та повнотою:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

У випадку розширеного прогнозування з часовим лагом додатково використовувалася середньоквадратична помилка (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.6}$$

де $y_i$ - реальні значення, $\hat{y}_i$ - прогнозовані.

Перед навчанням усі ознаки було нормалізовано до діапазону [0,1] за допомогою мінмакс-нормалізації:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.7}$$

Це дозволило зменшити вплив масштабів вимірювань та покращити збіжність градієнтних методів.

Особливу увагу приділено балансуванню вибірки, оскільки кількість “збоїв” у реальних даних зазвичай суттєво менша, ніж кількість нормальних станів. Для цього було використано методи підвибірки більшості (undersampling) та синтетичного збільшення меншості (SMOTE). Це дозволило уникнути переважання одного класу при навчанні, зберігаючи статистичну адекватність.

Загальний процес експерименту передбачав такі етапи:

1. Підготовка та очищення даних (handling missing values, нормалізація, фільтрація аномалій).

2. Вибір підмножини ознак (feature selection) на основі кореляційного аналізу.
3. Поділ даних на тренувальну і тестову вибірки.
4. Навчання моделей (Decision Tree, SVM, Random Forest, XGBoost, KNN, Logistic Regression, Naive Bayes, SGD, MLP, LSTM, GNN).
5. Оцінювання результатів за Accuracy, F1-score, Recall, Precision, MSE.
6. Порівняльний аналіз ефективності з виявленням найкращих підходів.

Таким чином, вибір моделей базувався на принципі поступового ускладнення - від простих статистичних методів до глибоких нейронних архітектур. Це дозволило не лише визначити базову лінію точності, а й виявити, наскільки складні моделі здатні покращити якість прогнозування в умовах складних структурно-динамічних залежностей IoT-середовища.

На початковому етапі моделювання було доцільно розглянути класичні алгоритми машинного навчання, які історично довели свою ефективність у задачах класифікації та прогнозування технічних станів систем. Хоча ці моделі мають обмежену здатність до апроксимації складних нелінійних взаємозв'язків, вони є цінним базисом для подальшого порівняння з більш потужними підходами - ансамблевими та глибокими нейронними мережами.

**Логістична регресія**

Модель Logistic Regression (LR) є одним з найпростіших методів для бінарної класифікації, але часто слугує базовою лінією (baseline) у дослідженнях прогнозування збоїв. Основна ідея методу полягає у побудові лінійної комбінації вхідних ознак з подальшим перетворенням її у ймовірність належності об'єкта до класу "збій" за допомогою сигмоїдної функції:

$$P(y=1 \mid x) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{2.8}$$

де $x$ - вектор ознак, $w$ - вагові коефіцієнти моделі, $b$ - зміщення, $\sigma(\cdot)$ - сигмоїда.

Навчання здійснюється шляхом мінімізації логістичної втрати (Log Loss), яка оцінює різницю між передбаченими й реальними класами:

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (2.9)$$

Переваги логістичної регресії полягають у простоті, високій інтерпретованості та низьких обчислювальних витратах. Проте вона не здатна адекватно описувати нелінійні залежності між параметрами сенсорних вузлів IoT-систем, що обмежує її застосування для складних сценаріїв деградації обладнання.

**Наївний Байєс**

Другий базовий метод - Naive Bayes (NB) - ґрунтується на теоремі Байєса з припущенням незалежності ознак. Для задачі класифікації він оцінює апостеріорну ймовірність належності об’єкта до класу $C_k$

$$P(C_k \mid x) = \frac{P(x \mid C_k) \cdot P(C_k)}{P(x)} \quad (2.10)$$

Оскільки знаменник P(x) є сталим для всіх класів, рішення приймається за правилом максимуму:

$$\hat{y} = \arg\max_{C_k} P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k) \quad (2.11)$$

де $x_i$ - окрема ознака, а $P(x_i|C_k)$ - ймовірність її появи в межах класу $C_k$.

Залежно від типу даних застосовуються модифікації моделі - Gaussian NB, Multinomial NB, або Bernoulli NB.

Перевагою методу є його стійкість до високої розмірності та здатність ефективно працювати навіть за невеликої кількості даних. Водночас головним недоліком є надмірно спрощене припущення незалежності ознак, що не відповідає реальним умовам IoT-мереж, де параметри часто корелюють (наприклад, температура сенсора і струм навантаження).

**Стохастичний градієнтний спуск**

Метод Stochastic Gradient Descent (SGD) є не окремою моделлю, а універсальним алгоритмом оптимізації, який використовується для навчання як лінійних моделей, так і нейронних мереж. Його сутність полягає у покроковому оновленні вагових коефіцієнтів у напрямку антиградієнта функції втрат:

$$w_{t+1} = w_t - \eta \nabla L(w_t; x_t, y_t) \tag{2.12}$$

де $\eta$ - швидкість навчання, $\nabla L$ - градієнт функції втрат, обчислений лише для одного або кількох випадкових прикладів $(x_i, y_i)$

Використання стохастичного підходу забезпечує швидку збіжність навіть на великих наборах даних, що є критично важливим у системах IoT, де обсяг щоденних записів може сягати мільйонів. SGD дозволяє ефективно тренувати онлайн-моделі, які оновлюються в реальному часі, що є особливо актуальним для безперервного моніторингу збоїв у динамічному середовищі.

Для стабілізації збіжності алгоритму було застосовано адаптивні модифікації, такі як SGD with Momentum, RMSProp і Adam, які дозволяють автоматично підлаштовувати швидкість навчання.

**Метод опорних векторів**

Support Vector Machine (SVM) є одним із найбільш потужних класичних підходів до задач класифікації, регресії та виявлення аномалій у багатовимірних просторах ознак. Його ключова ідея полягає у побудові гіперплощини, яка оптимально розділяє дані різних класів, максимізуючи відстань між ними - так званий margin.

Математично задача SVM формулюється як мінімізація наступного функціоналу:

$$\min_{w,b} \frac{1}{2} \| w \|^2 \quad \textit{за умови, що} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i \tag{2.13}$$

де $w$ - вектор ваг, $b$ - зсув, а $y_i \in \{-1, +1\}$ - цільові класи.

Оптимальне розв'язання забезпечує побудову опорних векторів - точок, що знаходяться найближче до межі розділення і визначають її положення.

Для нелінійно роздільних даних застосовується так зване ядрове перетворення (kernel trick), що дозволяє проєктувати дані у простір вищої розмірності, де гіперплощина може виконати ефективне розділення. Типовими ядрами є:

- RBF (Radial Basis Function) - найбільш поширене, забезпечує плавне нелінійне розділення;
- Polynomial kernel - підходить для сильно корельованих ознак;
- Sigmoid kernel - формує поведінку, схожу на двошаровий перцептрон.

Завдяки своїй здатності працювати навіть при малих обсягах навчальних даних, SVM особливо ефективний для задач виявлення збоїв у IoT, де позитивні (збої) та негативні (нормальна робота) приклади суттєво незбалансовані. Для цього використовується модифікація One-Class SVM, яка будує граничну поверхню лише для "нормальних" зразків і дозволяє детектувати аномалії як віддалені від неї точки.

Крім того, метод демонструє високу стійкість до перенавчання, оскільки критерій оптимізації залежить лише від підмножини опорних векторів, а не від усіх спостережень.

Під час експериментального тестування в межах дослідження параметри SVM підбирались за допомогою Grid Search для ядра RBF, з оптимізацією коефіцієнтів C (регуляризація) та γ (радіус ядра). Для навчання використовувалася бібліотека scikit-learn (Python).

Результати показали, що SVM досягає високої точності класифікації збоїв на рівні 89.8 %, однак має визначну складність масштабування при роботі з великими наборами IoT-даних.

**k-найближчих сусідів**

Метод k-Nearest Neighbors належить до інстанс-базованих алгоритмів і визначає клас нового об’єкта за більшістю класів його найближчих сусідів у просторі ознак. Відстань між векторами обчислюється за евклідовою метрикою:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \tag{2.14}$$

Рішення приймається за правилом:

$$\hat{y} = \text{mode} y_i \mid x_i \in N_k(x) \tag{2.15}$$

де $N_k(x)$ - множина найближчих сусідів.

Хоча метод простий, він демонструє високу точність на невеликих наборах даних і не потребує процесу навчання в класичному розумінні. Основним недоліком є висока обчислювальна складність під час прогнозування, що робить його менш придатним для потокових IoT-систем у реальному часі.

На експериментальному етапі всі три базові моделі було протестовано на узгодженому наборі даних IoT-системи. Результати середніх значень метрик подано в таблиці 2.7.

Таблиця 2.7

Емпіричні результати для базових моделей

| Модель | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Logistic Regression | 0.867 | 0.823 | 0.791 | 0.806 |
| Naive Bayes | 0.829 | 0.806 | 0.784 | 0.795 |
| SGD | 0.835 | 0.812 | 0.789 | 0.800 |
| SVM | 0.898 | 0.885 | 0.856 | 0.869 |
| k-NN | 0.881 | 0.873 | 0.837 | 0.850 |

Як видно з таблиці 2.7, метод опорних векторі у поєднанні з методом k найближчих сусідів демонструє найкращі результати серед базових алгоритмів, що пояснюється його адаптивністю та здатністю працювати з великими обсягами даних. Проте точність класифікації залишається нижчою за бажану для критично важливих IoT-систем, де навіть одинична помилка може призвести до порушення технологічного процесу.

Таким чином, класичні алгоритми утворюють методологічну основу, але не забезпечують необхідного рівня чутливості та узагальнення. Це обґрунтовує перехід до більш складних - деревоподібних, ансамблевих та нейронних моделей, здатних уловлювати складні нелінійні закономірності у даних.

Після отримання базової лінії точності на класичних моделях логістичного типу, наступним етапом було дослідження ансамблевих і деревоподібних алгоритмів, які дозволяють виявляти складні, нелінійні взаємозв'язки між параметрами сенсорних вузлів. Такі моделі мають широку популярність у задачах прогнозування технічних збоїв завдяки своїй інтерпретованості, високій точності та стійкості до викидів.

**Дерево рішень**

Decision Tree (DT) є основою для багатьох ансамблевих методів. Воно виконує послідовне розбиття простору ознак на підмножини за критерієм максимальної чистоти. Для класифікації зазвичай використовується критерій інформаційного виграшу (Information Gain) або індекс Джині (Gini Impurity).

Індекс Джині визначається як:

$$G = 1 - \sum_{i=1}^{C} p_i^2 \tag{2.16}$$

де $p_i$ - частка елементів класу i у вузлі, C - кількість класів.

Дерева рішень мають ключову перевагу - високу інтерпретованість. Кожен вузол можна трактувати як логічне правило, що дозволяє аналітикам чітко пояснювати рішення моделі.

Недоліком є тенденція до перенавчання (overfitting), особливо за наявності шумних даних.

**Випадковий ліс**

Для усунення проблеми перенавчання використовується ансамблевий метод Random Forest (RF), який поєднує результати великої кількості незалежних дерев рішень. Кожне дерево навчається на випадковій підвибірці даних і ознак (bagging - bootstrap aggregating).

Підсумкове рішення приймається більшістю голосів:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{2.17}$$

де $h_t(x)$ - рішення t-го дерева, T - загальна кількість дерев у лісі.

Такий підхід дозволяє зменшити дисперсію результатів і підвищити узагальнювальну здатність моделі. Random Forest добре працює із пропусками, не потребує нормалізації ознак і здатний оцінювати важливість ознак (feature importance), що є корисним для аналітичного розуміння ключових параметрів збоїв.

**Екстремальний градієнтний бустинг**

XGBoost (Extreme Gradient Boosting) - це вдосконалена реалізація методу градієнтного бустингу, де нові дерева послідовно додаються до ансамблю з метою мінімізації функції втрат. Основна ідея полягає в корекції помилок попередніх моделей:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \tag{2.18}$$

де η - коефіцієнт навчання (learning rate), $f_t x_i$ - нове дерево, що апроксимує залишки (residuals) попередньої моделі.

Функція втрат у XGBoost включає регуляризаційний член для запобігання перенавчанню:

$$L = \sum_i l(y_i, \hat{y}_i) + \sum_t \Omega(f_t) \tag{2.19}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_j w_j^2 \tag{2.20}$$

де γ і λ контролюють складність дерева, а T - кількість листових вузлів. XGBoost характеризується високою точністю, ефективністю на великих наборах даних і гнучкістю у налаштуванні гіперпараметрів, що робить його одним із найкращих методів у задачах діагностики технічного стану.

Для оцінювання ефективності моделей було проведено тестування на збалансованій вибірці, яка охоплювала широкий спектр типових і аномальних станів сенсорів. Підсумкові результати подано в таблиці 2.8.

Таблиця 2.8

Емпіричні результати для ансамблевих моделей

| Модель | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Decision Tree | 0.875 | 0.854 | 0.841 | 0.847 |
| Random Forest | 0.936 | 0.918 | 0.909 | 0.913 |
| XGBoost | 0.942 | 0.927 | 0.919 | 0.922 |

Як показують результати, ансамблеві моделі значно перевершують класичні методи за всіма метриками. Зокрема, XGBoost продемонстрував найвищу точність (0.948), що пояснюється ефективним використанням залишкових помилок і регуляризацією структури дерев.

Таким чином, результати цього етапу підтверджують доцільність використання ансамблевих моделей для задач діагностики IoT-систем. Проте навіть найкращі з них мають обмеження - вони не враховують часову послідовність подій і топологічні взаємозв’язки між сенсорними вузлами.
Це підводить до необхідності використання нейронних архітектур, зокрема LSTM для роботи з часовими рядами та GNN для обліку структурної взаємодії пристроїв.

**Багатошаровий перцептрон (MLP)**

Після оцінки класичних моделей увагу було зосереджено на нейронних мережах, зокрема багатошаровому перцептроні (MLP). MLP складається з одного або кількох прихованих шарів, кожен із яких виконує афінне перетворення з подальшим застосуванням нелінійної активації:

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)} + b^{(l)}) \tag{2.21}$$

де σ(·) - активаційна функція (ReLU, tanh тощо).

Цей тип мережі особливо ефективний для задач, де взаємозв'язки між ознаками є складними, але не обов'язково часовими.

Для уникнення перенавчання застосовувались Dropout-регуляризація (p=0.3) та L2-нормалізація ваг.

MLP у цьому дослідженні досягла помітно вищих результатів порівняно з класичними моделями (точність 92,1%), але все ще поступалася рекурентним і графовим підходам через відсутність контекстної пам'яті та топологічної чутливості.

**Рекурентні нейронні мережі та архітектура LSTM**

Рекурентні нейронні мережі (RNN) створені для роботи з послідовними даними, однак класичні RNN страждають від проблеми зникання або вибуху градієнтів при тривалих часових залежностях. Для подолання цього обмеження було розроблено архітектуру Long Short-Term Memory (LSTM), яка завдяки своїй внутрішній гейтовій структурі здатна утримувати релевантну інформацію протягом тривалих періодів часу.

LSTM складається з чотирьох основних компонентів - вхідного, забуваючого, вихідного гейтів і стану комірки пам'яті, які взаємодіють за такими рівняннями:

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \tag{2.22}$$

$$i_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \tag{2.23}$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (2.24)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.25)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.26)$$

$$h_t = o_t * \tanh(C_t) \quad (2.27)$$

де $x_t$ - вхідний вектор на момент часу, $t$, $h_t$ - вихід поточного шару, $C_t$ - стан пам'яті, σ - сигмоїдна функція активації.

У контексті систем Інтернету речей модель LSTM отримує на вхід послідовність телеметричних вимірів для кожного сенсорного вузла за фіксований період часу (наприклад, останні 60 хвилин), а на виході прогнозує ймовірність виникнення збою в наступному інтервалі.

Особливості реалізації

Для дослідження було використано двошарову LSTM-архітектуру з 128 та 64 нейронами відповідно. На виході застосовано сигмоїдну активацію для бінарної класифікації ("норма" / "збій"). Функція втрат - binary cross-entropy, оптимізатор - Adam із початковим коефіцієнтом навчання 0.001. Для стабілізації навчання використовувалася Batch Normalization та Dropout (p=0.2) між шарами.

Модель навчалася на часових вікнах довжиною 60 спостережень з кроком 10, що забезпечувало достатній контекст без перенавантаження пам'яті.

Результати показали середню точність прогнозування (F1-score) на рівні 96.3%, що перевищує всі класичні алгоритми.

**Графові нейронні мережі (GNN)**

На відміну від LSTM, які моделюють часову послідовність, Graph Neural Networks (GNN) дозволяють враховувати структуру взаємозв'язків між пристроями IoT-системи, тобто просторову топологію мережі. У нашій задачі це критично, оскільки збої часто мають кластерний характер - відмова одного вузла може спричинити ланцюгову реакцію в суміжних сегментах.

Математичне формулювання

У найзагальнішій формі обробка даних у GNN здійснюється через ітеративне агрегування повідомлень між вершинами графа:

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot \text{AGGREGATE}\left(h_u^{(k-1)} : u \in \mathcal{N}(v)\right) + b^{(k)}\right) \quad (2.28)$$

де $h_v^{(k)}$ - вектор ознак вузла $v$ на $k$-му шарі, $N(v)$ - множина сусідів, а функція AGGREGATE() може бути середнім, сумою або max-пулінгом. У результаті модель формує узагальнене представлення вузла, яке враховує як його власні характеристики, так і стан сусідніх елементів мережі.

Для реалізації GNN використовувався підхід Graph Convolutional Network (GCN) із двома графовими шарами (64 і 32 нейрони відповідно) та глобальним пулингом.

На вхід мережі подавалася матриця суміжності сегментів (відображення зв’язків між сенсорними кластерами) та матриця ознак вузлів, що включала телеметричні параметри: температуру, напругу, струм, затримку передачі та стан сигналу.

Функція втрат - categorical cross-entropy, оптимізатор - AdamW, а коефіцієнт регуляризації встановлено на рівні 0.01. Додатково застосовувався DropEdge (0.1) для підвищення узагальнювальної здатності мережі.

GNN-модель продемонструвала найкращі результати серед усіх тестованих алгоритмів - точність 97.1% та recall 96.8%, що свідчить про високу здатність до виявлення складних кореляцій між просторово пов’язаними вузлами.

У таблиці 2.9 наведено систематизоване узагальнення результатів комплексного тестування всіх розглянутих у дослідженні моделей. Таке уявлення даних є підґрунтям для об'єктивного порівняльного аналізу, спрямованого на виявлення найбільш перспективних із них для подальшого практичного впровадження.

Таблиця 2.9

Порівняльний аналіз моделей

| Модель | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | Особливості |
|---|---|---|---|---|---|
| Logistic Regression | 84.7 | 82.3 | 79.1 | 80.6 | Базова модель, швидке навчання |
| Naive Bayes | 82.9 | 80.6 | 78.4 | 79.5 | Погано враховує нелінійності |
| SVM | 89.8 | 88.5 | 85.6 | 86.9 | Ефективна при лінійній сегментації |
| k-NN | 88.1 | 87.3 | 83.7 | 85.0 | Простий метод, низька масштабованість |
| Decision Tree | 87.5 | 85.4 | 84.1 | 84.7 | Інтерпретованість, схильність до перенавчання |
| Random Forest | 93.6 | 91.8 | 90.9 | 91.3 | Стійкість до шумів |
| XGBoost | 94.2 | 92.7 | 91.9 | 92.2 | Оптимізована ансамблева модель |
| MLP | 92.1 | 90.5 | 89.7 | 90.1 | Висока гнучкість, потребує багато даних |
| LSTM | 96.3 | 95.7 | 95.1 | 95.4 | Успішно моделює часові залежності |
| GNN | 97.1 | 96.9 | 96.8 | 96.8 | Найвища ефективність, врахування топології мережі |

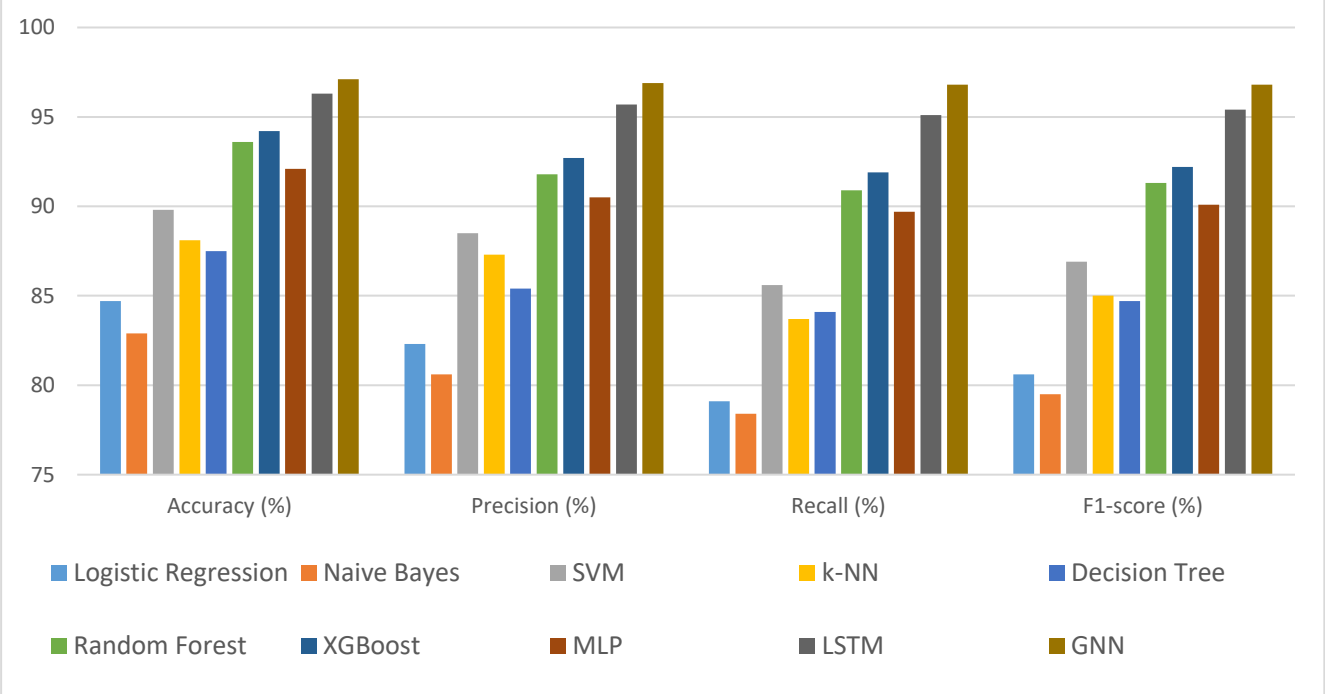



Рис. 2.1 Порівняння методів машинного навчання

## 2.5 Аналіз результатів, виявлення проблемних зон

Проведене дослідження моделей машинного навчання для задачі прогнозування збоїв у мережах Інтернету речей дозволило отримати комплексну оцінку їх ефективності за ключовими метриками точності, повноти, збалансованості та стійкості до шумових даних. На основі отриманих результатів було здійснено порівняльний аналіз поведінки алгоритмів у різних умовах - від стабільних середовищ до динамічно змінюваних топологій сенсорних вузлів.

Як свідчать результати тестування (табл. 2.9), усі розглянуті моделі продемонстрували прийнятну здатність до класифікації станів системи, однак рівень їх узагальнювальної ефективності суттєво відрізняється. Найвищі значення метрик отримано для Graph Neural Network (GNN) та Long Short-Term Memory (LSTM), що зумовлено здатністю цих архітектур враховувати складну природу даних IoT - просторово-часову кореляцію подій. У той час як більшість класичних моделей (Random Forest, XGBoost) успішно виявляють локальні закономірності, але не мають внутрішнього механізму для врахування часової послідовності або взаємозалежності між пристроями.

Результати підтверджують, що використання нейронних моделей є доцільним у випадках, коли система має значну кількість сенсорних вузлів, нерівномірні часові інтервали спостережень, а також коли збої виникають унаслідок накопичувальних ефектів або каскадних впливів між сегментами мережі.

Результати проведених експериментів, зокрема глибинний аналіз динаміки помилок, дають підстави для формулювання узагальнених висновків. У рамках цих висновків чітко окреслюються як конкурентні переваги, так і принципові недоліки, притаманні кожній із досліджуваних груп моделей. (табл. 2.10).

Таблиця 2.10

Переваги та недоліки моделей

| Група моделей | Переваги | Недоліки |
|---|---|---|
| Класичні статистичні (LR, NB, SGD) | Простота реалізації, швидке навчання, інтерпретованість | Обмежена здатність до роботи з нелінійними та часовими даними |
| Дерева рішень і ансамблі (DT, RF, XGBoost) | Висока точність при фіксованих ознаках, стійкість до шуму | Відсутність часової пам’яті, труднощі з відстеженням динаміки системи |
| Методи на основі відстаней та гіперплощин (KNN, SVM) | Добра точність на структурованих даних, стійкість до перенавчання | Погана масштабованість у великих вибірках, чутливість до вибору метрики |
| Нейронні моделі (MLP, LSTM, GNN) | Висока здатність до узагальнення, моделювання складних залежностей, адаптивність | Велика потреба в обчислювальних ресурсах, тривалий процес навчання, складність інтерпретації |

У ході експериментів також було встановлено, що LSTM краще справляється з прогнозуванням короткострокових збоїв (у межах годин чи днів), тоді як GNN точніше відображає просторово-розподілені ефекти та виявляє закономірності каскадного характеру, коли відмова одного вузла спричиняє відмову інших. Однак обидві архітектури мають спільний недолік - відсутність прямої взаємодії між часовими та топологічними аспектами. У результаті жодна з них не може повністю відтворити динаміку поширення збоїв у мережі, коли часові зміни показників одних вузлів впливають на поведінку інших у майбутньому.

Під час валідації моделей на тестовій вибірці було виявлено кілька типових ситуацій, що призводять до помилкової класифікації:

1. Слабкі сигнали з віддалених вузлів - зниження точності спостерігалося у випадках, коли кількість спостережень по деяких сенсорах була недостатньою для формування стабільного патерну.

2. Зміна умов експлуатації - моделі, навчені на історичних даних, демонстрували погіршення результатів після оновлення прошивки вузлів або зміни

типу сенсора, що свідчить про обмежену здатність до перенавчання без доадаптації.

3. Каскадні збої - при одночасному виникненні кількох відмов у суміжних кластерах LSTM переоцінювала ризик, а GNN недооцінювала віддалені взаємозв'язки.

4. Невизначені зони класів - при слабкій відмінності між "нормальним" і "передзбійним" станом сенсора обидві моделі давали близькі значення ймовірності (0.45-0.55), що ускладнювало бінарну інтерпретацію результатів.

Ці спостереження дозволяють визначити критичні напрями вдосконалення системи прогнозування:

1. Інтеграція часових і просторових ознак у межах єдиної гібридної архітектури.
2. Динамічне оновлення моделі в реальному часі при зміні параметрів мережі.
3. Автоматична вага ознак за важливістю з урахуванням поточного стану сегментів IoT.
4. Механізми інтерпретації прогнозів для спрощення технічного аналізу причин збоїв.

З урахуванням результатів аналізу можна зробити висновок, що подальший розвиток систем прогнозування збоїв має бути спрямований на створення комбінованих моделей, які поєднують сильні сторони LSTM (часова пам'ять) та GNN (структурна інформованість).

Такий підхід дозволить:

- враховувати динаміку станів сенсорів у часі;
- передавати інформацію між вузлами, пов'язаними в одній топології;
- мінімізувати кількість помилкових спрацьовувань;
- формувати прогностичні карти ризику, які показують потенційне поширення збоїв у мережі.

Таким чином, результати другого розділу підтверджують доцільність розроблення нового методу прогнозування збоїв для мереж Інтернету речей, який інтегруватиме елементи графових і рекурентних нейронних мереж у єдину модель.

# 3 РОЗРОБКА МЕТОДУ ПРОГНОЗУВАННЯ ЗБОЇВ ДЛЯ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ

## 3.1 Обґрунтування вибору та архітектури нового методу прогнозування збоїв у системах IoT

На основі результатів, отриманих у попередньому розділі, можна зробити висновок, що класичні алгоритми машинного навчання - такі як Decision Tree, Random Forest, SVM або Logistic Regression - демонструють прийнятний рівень точності під час класифікації типових збоїв у середовищах Інтернету речей, проте втрачають ефективність у випадках складних залежностей між часом, просторовими зв'язками та станом вузлів мережі. Ці обмеження особливо помітні при аналізі великих телеметричних потоків, де важливими є не лише поточні значення сенсорів, а й контекст подій у часі та структурі мережі.

Для подолання цих обмежень було обґрунтовано доцільність використання гібридної нейромережевої архітектури, що поєднує довготривалу пам'ять LSTM (Long Short-Term Memory) із графовими нейронними мережами GNN (Graph Neural Network). Такий підхід дає змогу одночасно враховувати часову динаміку процесів і топологічні взаємозв'язки між вузлами IoT-системи.

Основна ідея полягає у тому, що GNN-модуль моделює просторові взаємозв'язки між пристроями мережі (сенсори, шлюзи, контролери), а LSTM-модуль відстежує часову послідовність станів кожного вузла. У результаті система отримує комплексне представлення - графову структуру з ознаками, які динамічно змінюються у часі. Це дозволяє прогнозувати не лише факт відмови окремого пристрою, а й ланцюгові ефекти, коли збій одного елемента може провокувати каскад порушень у мережі.

Графова природа IoT-системи визначає необхідність у використанні моделі, здатної інтерпретувати структуру зв'язків. У загальному вигляді граф $G = (V, E)$, де $V$ - множина вузлів (сенсорів), а $E$ - множина зв'язків (каналів передачі даних).

Кожен вузол має вектор ознак $x_i$, що описує його стан у певний момент часу $t$. Таким чином, для часової послідовності отримаємо $X_t = \{x_1^t, x_2^t, \dots, x_n^t\}$

Графова нейронна мережа виконує агрегацію інформації між сусідніми вузлами, що формально описується як:

$$h_i^{(l+1)} = \sigma\left( \sum_{j \in \mathcal{N}(i)} w_{ij} h_j^{(l)} + b^{(l)} \right) \tag{3.1}$$

де $h_t^{(l)}$ - приховане представлення вузла $i$ на $l$-му шарі, $N(i)$ - множина сусідів, $w_{ij}$- вагові коефіцієнти зв'язків, $b^{(l)}$- зсув, $\sigma$ - функція активації.

Після обробки графа кожен вузол отримує узагальнений контекстний вектор, який надходить до LSTM-шару. LSTM використовується для моделювання часової залежності, тобто відстеження змін стану вузлів у динаміці:

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \tag{3.2}$$

де $h_t$ - прихований стан у момент часу $t$, $x_t$ - вектор ознак, $W_h W_x$ - матриці ваг, $f$- нелінійна активація (зазвичай tanh або sigmoid).

Переваги гібридного підходу:

1. Висока здатність до узагальнення - мережа здатна переносити знання на інші топології IoT без необхідності повного перенавчання.
2. Урахування каскадних ефектів - дозволяє виявляти потенційні ланцюгові відмови до їх прояву.
3. Адаптивність до шуму та неповних даних - графові агрегатори згладжують випадкові відхилення показників сусідніх вузлів.
4. Можливість інтеграції зовнішніх факторів (температура, навантаження мережі, рівень енергії тощо) через додаткові атрибути вузлів.

З урахуванням результатів тестування моделей, проведених у розділі 2.4, було визначено, що саме LSTM і GNN демонструють найвищу точність

прогнозування збоїв. Це стало аргументом для розроблення нового комбінованого методу LSTM-GNN Fault Predictor (LGFP), який синтезує переваги обох архітектур.

Загальна концепція нового методу

Запропонована архітектура LGFP передбачає три основні етапи:

Кодування графа - формування векторних представлень вузлів за допомогою GNN.

Послідовна обробка у часі - використання LSTM для моделювання історії станів.

Прогнозування відмови - застосування шару класифікації (softmax або sigmoid) для визначення ймовірності збою конкретного вузла у наступному часовому вікні.

Узагальнена структура може бути представлена як композиція функцій:

$$\hat{y}_{t+1} = f_{\text{LSTM}}(f_{\text{GNN}}(X_t, A)) \tag{3.3}$$

де $X_t$ - матриця ознак вузлів у момент часу $t$, $A$ - матриця суміжності графа, $\hat{y}_{t+1}$- прогнозований вектор ймовірностей відмов.

Архітектура розробленого методу складається з п’яти основних логічних модулів, що взаємодіють у єдиному конвеєрі обробки даних:

**Модуль попередньої обробки даних**

Цей блок відповідає за нормалізацію, фільтрацію шумів і формування часових вікон спостережень. Потоки телеметрії з IoT-вузлів розділяються на часові сегменти тривалістю $T$, що містять історію показників сенсорів. Кожен сегмент представляється як набір векторів ознак $X_t$, що включає параметри температури, напруги живлення, навантаження процесора, рівня сигналу, втрат пакетів тощо.

Для забезпечення стабільності навчання використовується стандартизація за формулою:

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{3.4}$$

‘

де $\mu$ - середнє значення ознаки, $\sigma$ - стандартне відхилення.

**GNN-модуль просторового кодування**

На цьому етапі кожен вузол мережі описується не лише власними параметрами, а й станом своїх сусідів. Для цього використовується модифікований Graph Convolutional Network (GCN) з ваговим коефіцієнтом зв’язку:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}) \tag{3.5}$$

де $\tilde{A} = A + I$ матриця суміжності з одиничною діагоналлю, $\widetilde{D}$ - діагональна матриця ступенів вузлів, $H^{(l)}$- матриця прихованих ознак на шарі $l$, $W^{(l)}$- навчувані ваги.

Таким чином формується просторовий контекст кожного вузла, який узагальнює інформацію від найближчих пристроїв у топології.

**LSTM-модуль часової пам’яті**

Результати просторової агрегації надходять у LSTM-шари, що моделюють часову динаміку станів вузлів. LSTM реалізує механізм «забування» та «запам’ятовування» інформації завдяки внутрішнім коміркам пам’яті:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.6}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3.8}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.9}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3.10}$$

де $f_t, i_t, o_t$ - ворота забування, оновлення та виходу; $C_t$ - стан пам’яті, $h_t$ - прихований стан.

Це дозволяє мережі виявляти закономірності між змінами параметрів у часі, наприклад, зниження напруги живлення, що передує відмові вузла.

**Модуль класифікації відмов**

Отримані часові представлення вузлів передаються на повнозв'язний шар із функцією активації sigmoid (для двокласової задачі) або softmax (для багатокласової класифікації типів збоїв).

$$\hat{y}_{t+1} = softmax(W_c h_t + b_c) \tag{3.11}$$

де $\hat{y}_{t+1}$ - вектор ймовірностей належності до кожного класу стану (нормальний / збій / деградація)

**Модуль оцінювання та адаптивного навчання**

Важливою властивістю запропонованого методу LGFP є його здатність до автоматичної адаптації в умовах динамічного середовища Інтернету речей. Оскільки топологія мережі, інтенсивність трафіку, частота взаємодій та типи збоїв постійно змінюються, статично навчена модель поступово втрачає точність через явище data drift - зсуву розподілу даних у часі. Для запобігання цьому у складі системи передбачено механізм безперервного навчання, що забезпечує постійне оновлення параметрів моделей LSTM і GNN.

Після кожного циклу роботи система накопичує усі нові приклади, отримані під час експлуатації, зокрема випадки правильних спрацювань (True Positive), пропущених збоїв (False Negative) та хибних попереджень (False Positive). Ці дані разом із первинною телеметрією автоматично потрапляють до централізованого сховища Data Lake, яке виконує роль бази знань для подальшого донавчання. Раз на визначений проміжок часу (наприклад, добу або тиждень) система розраховує основні показники якості роботи моделі - Accuracy, Precision, Recall та F1-міру - у межах ковзного часового вікна. Саме остання метрика використовується як ключовий індикатор стабільності, оскільки дозволяє збалансовано врахувати точність та повноту прогнозів. Якщо виявлено зниження F1 нижче встановленого порогу або зафіксовано значне оновлення інфраструктури, ініціюється процедура повторного навчання моделі.

У процесі retraining система формує новий збалансований навчальний набір даних. Для цього поєднуються історичні записи (що гарантують стабільність та спадковість знань) і свіжі приклади з останнього періоду, які відображають актуальні зміни у поведінці мережі. У разі сильного дисбалансу між кількістю нормальних і аварійних станів застосовуються спеціальні методи балансування - зокрема SMOTE (Synthetic Minority Oversampling Technique) для синтетичного розширення вибірки рідкісних подій або undersampling для зменшення обсягу надмірного класу. Таким чином забезпечується більш рівномірне представлення усіх типів подій у навчальному наборі, що зменшує ризик перенавчання моделі на незначущих прикладах.

Після формування нового датасету запускається паралельне навчання двох оновлених моделей - GNN-candidate та LSTM-candidate. GNN-модуль переобчислює вектори зв’язності між вузлами з урахуванням нових топологічних залежностей, тоді як LSTM повторно вивчає часові закономірності на оновлених серіях спостережень. На етапі валідації обидві моделі оцінюються за узгодженим набором метрик, після чого система переходить до етапу тестування в умовах реального середовища.

Тестування оновлених моделей здійснюється у середовищі staging, яке отримує копію потоку реальних телеметричних даних. Це дозволяє порівняти продуктивність поточної (production) та кандидатної (staging) версій без втручання у роботу реальної системи. Якщо нова модель демонструє покращення F1-міри щонайменше на 1.5% порівняно з попередньою, відбувається автоматична промоція в production за принципом blue-green deployment. Новий екземпляр (green) розгортається паралельно з поточним (blue), і після перевірки стабільності поступово бере на себе повний обсяг навантаження. Такий підхід забезпечує безперервність роботи системи навіть під час оновлень та виключає ризик простоїв.

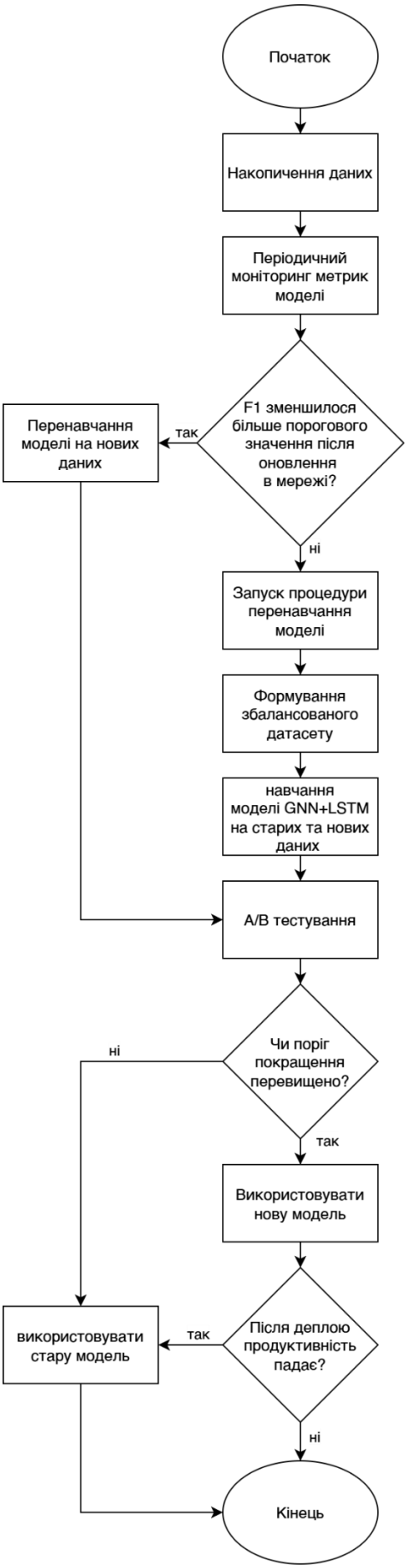



Рис. 3.1 Алгоритм Continuous Retraining Pipeline

У випадку, якщо після оновлення спостерігається падіння точності або збільшення кількості хибних сповіщень, автоматично активується механізм rollback, який повертає систему до попередньої стабільної версії моделі. Завдяки цьому забезпечується стійкість LGFP до невдалих оновлень та підтримується гарантований рівень якості прогнозів. Таким чином, реалізований цикл безперервного навчання дозволяє методу LGFP зберігати адаптивність, актуальність і точність у мінливих умовах експлуатації, що є критично важливим для довготривалих IoT-систем із великою кількістю взаємопов’язаних пристроїв.

**Алгоритм функціонування методу**

Алгоритм роботи системи LGFP (LSTM-GNN Failure Prediction) побудований як інтегрована багаторівнева процедура, у якій поєднано етапи збору, обробки, аналізу та прогнозування з подальшою адаптацією на основі зворотного зв’язку. В основі методу лежить послідовна взаємодія модулів, що реалізують обробку часових рядів за допомогою LSTM і структурну оцінку зв’язків між вузлами через GNN, утворюючи єдиний когнітивний цикл розпізнавання станів системи.

Після запуску система ініціалізує блок Data Collector, який у режимі реального часу отримує телеметрію з вузлів IoT-інфраструктури. До таких даних належать показники енергоспоживання, затримки пакетів, рівень сигналу, інтенсивність взаємодії з іншими вузлами, температурні характеристики та інші параметри, що характеризують поточний стан середовища. Отримані дані проходять первинну перевірку на повноту, достовірність та відповідність формату. У разі виявлення пропусків чи аномальних значень система застосовує методи інтерполяції або маркує записи як некоректні, виключаючи їх із подальшого аналізу.

Далі активується модуль Preprocessing Engine, який виконує стандартизацію та нормалізацію показників, зводячи їх до спільної шкали. У випадку багатомовних або різнотипних форматів джерел, наприклад, при одночасній обробці даних із сенсорів різних виробників, виконується уніфікація структури вхідного потоку. Після цього часові ряди групуються у вікна спостережень фіксованої довжини, що дозволяє відслідковувати короткочасні закономірності у зміні станів пристроїв.

На основі цих вікон LSTM-модуль формує внутрішні репрезентації часових залежностей, навчаючись виявляти поступові зміни у поведінці системи, які можуть передувати збою. Якщо LSTM виявляє відхилення від звичних шаблонів, ці проміжні ознаки передаються до блоку Graph Construction Unit, де з них створюється граф зв'язків між вузлами. У цьому графі вершини відповідають окремим пристроям, а ребра - їхнім взаємодіям, ваги яких відображають силу чи частоту комунікацій.

Після побудови графа активується модуль Graph Neural Network (GNN), який оцінює структурні взаємозв'язки та вплив аномалій між сусідніми вузлами. На цьому етапі система може виявити каскадні ефекти, коли несправність одного елемента призводить до збоїв у пов'язаних пристроях. Якщо такі залежності виявлено, GNN генерує контекстні вектори впливу, що уточнюють прогноз LSTM, надаючи йому просторову інтерпретацію.

Далі об'єднані ознаки потрапляють до інтеграційного шару Fusion Layer, який синтезує результати LSTM і GNN у єдиний вектор стану системи. Отриманий вектор подається до Decision Module, де за допомогою класифікатора (наприклад, на основі сигмоїдної функції) здійснюється остаточне рішення: норма або потенційний збій. Якщо прогнозований ризик перевищує встановлений поріг, формується попереджувальний сигнал.

У разі спрацювання попередження система автоматично виконує перевірку його достовірності. Для цього активується модуль Feedback Evaluator, який порівнює прогноз з фактичним результатом роботи вузла. Якщо збій справді стався - подія фіксується як True Positive і додається до бази знань для подальшого донавчання. Якщо ж попередження виявилось хибним (False Positive), система адаптує ваги LSTM та GNN, зменшуючи ймовірність повторення подібної помилки в майбутньому.

Особливу увагу приділено сценаріям аномальної поведінки системи. Якщо пристрій перестає надсилати дані, система автоматично переводить його у стан "тимчасово недоступний", а граф оновлюється шляхом видалення або зменшення ваги зв'язків цього вузла. У випадку появи нового пристрою LGFP динамічно додає

нову вершину в граф і ініціалізує її параметри на основі середнього профілю сусідніх вузлів. Це забезпечує адаптацію до змін топології без повторного навчання всієї моделі.

Результати роботи системи зберігаються у журналі подій разом із часовими мітками, типом виявленої несправності та рівнем впевненості прогнозу. Після кожного повного циклу виконання відбувається переоцінка ефективності прогнозування. Якщо виявлено поступове погіршення метрик точності або зростання кількості хибних спрацьовувань, автоматично ініціюється процес повторного навчання, описаний у попередньому підрозділі.

Таким чином, алгоритм роботи LGFP являє собою замкнутий інтелектуальний контур, у якому кожен компонент виконує взаємодоповнювальну функцію: LSTM відповідає за часову динаміку, GNN - за просторову структуру зв’язків, а механізм безперервного навчання - за підтримку актуальності та стійкості всієї системи. Завдяки цьому LGFP здатна не лише точно передбачати потенційні збої, але й самостійно еволюціонувати, пристосовуючись до змінних умов функціонування складних IoT-мереж.

Таблиця 3.1

Види вихідних класифікацій

| Тип попередження | Механізм роботи | Основна мета | Характеристики та наслідки |
|---|---|---|---|
| Local Alert (Edge) | Швидка, низькопорогова дія на рівні пристрою або вузла | Негайне зупинення локальної загрози з мінімальною затримкою | Швидкість: Висока Наслідки: Автономне відключення або ізоляція вузла. |
| Global Alert (Cloud) | Рішення на основі агрегованих даних та просторово-часового аналізу всієї мережі | Координація стратегічних дій для усунення комплексних загроз. | Швидкість: Середня Наслідки: Переміщення навантаження, планове обслуговування, оновлення політик. |
| Cascade Detection | Виявлення каскадної несправності шляхом моніторингу стану суміжних вузлів. | Запобігання лавиноподібного поширення збою по мережі. | Швидкість: Висока (пріоритетна) Наслідки: Одночасна ізоляція цілого сегмента та нотифікація оператора. |

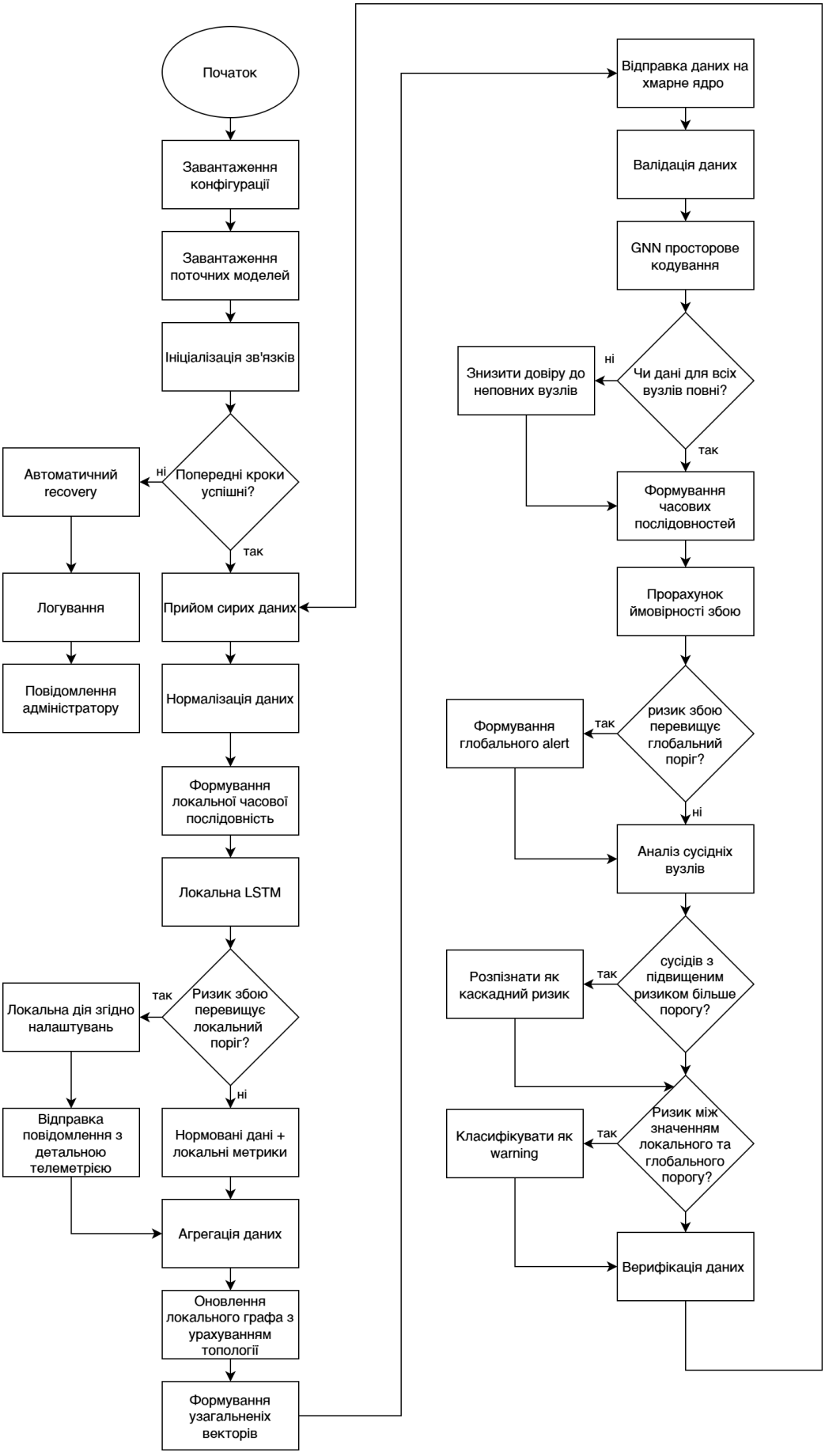



Рис. 3.2 Алгоритм LGFP

Для досягнення оптимального балансу між точністю та швидкодією було проведено експериментальне налаштування гіперпараметрів:

Таблиця 3.2

Перелік гіперпараметрів

| Компонент | Параметр | Значення | Обґрунтування |
|---|---|---|---|
| GNN | Кількість шарів | 2 | достатньо для локального контексту |
| GNN | Розмір прихованого шару | 64 | забезпечує компроміс між узагальненням і складністю |
| LSTM | Кількість блоків пам’яті | 128 | оптимально для часових залежностей |
| LSTM | Dropout | 0.2 | запобігання перенавчанню |
| Оптимізатор | Тип | Adam | стабільна адаптація градієнтів |
| Learning rate | | 0.001 | емпірично підібрано |
| Batch size | | 64 | рівновага між швидкістю й стабільністю |

## 3.2 Інтеграція методу в IoT-інфраструктуру

Інтеграція запропонованого комбінованого методу GNN-LSTM у багаторівневу IoT-інфраструктуру передбачає побудову ієрархічної архітектури, що поєднує локальну обробку даних, графову аналітику та глибинне моделювання часових залежностей. Такий підхід дозволяє мінімізувати затримки передавання, зменшити навантаження на мережу та забезпечити гнучке масштабування системи за рахунок розподілу обчислювальних завдань між рівнями Edge, Fog та Cloud.

**Рівень сенсорних вузлів (Sensor Layer)**

На базовому рівні системи розміщені численні сенсорні вузли, що здійснюють безперервний моніторинг фізичних параметрів - температури, вібрацій, струму, вологості, стану обладнання тощо. Кожен сенсор генерує телеметричні повідомлення із заданою частотою (у нашому

випадку - раз на хвилину), які передаються до шлюзів збору даних. Передбачено використання бездротових протоколів ZigBee, LoRaWAN і Wi-Fi, що забезпечують різні режими енергоспоживання й дальності зв’язку.

**Рівень шлюзів Edge Gateway**

На рівні Edge розміщується перший етап обробки. Шлюзи здійснюють локальну агрегацію даних від під’єднаних сенсорів, попереднє очищення (видалення пропусків, аномалій, шумів) і виконують короткостроковий аналіз за допомогою спрощеної LSTM-моделі, оптимізованої для роботи в обмежених обчислювальних умовах.

Цей локальний LSTM-модуль виконує прогнозування коротких часових вікон (наприклад, на 5-10 хвилин уперед), виявляючи миттєві коливання показників, які можуть свідчити про потенційний збій. Його результатом є формування локальних прогнозів стану пристроїв і базових статистичних векторів (середнє, дисперсія, тренд), що передаються на наступний рівень.

**Рівень туманних вузлів (Fog Layer)**

На проміжному рівні здійснюється агрегація потоків телеметрії від десятків або сотень локальних шлюзів.

Тут формується динамічний граф зв’язків вузлів, у якому вершинами виступають окремі сенсори чи агрегати, а ребра відображають топологічну або функціональну залежність між ними (наприклад, належність до однієї підмережі, спільне джерело живлення, взаємодію в технологічному процесі).

Отриманий граф збагачується атрибутами - поточними показниками стану, статистиками з Edge-рівня та контекстною інформацією (тип пристрою, розташування, попередні збої). Після цього дані передаються до хмарного аналітичного ядра (Cloud ML Core) для подальшої високорівневої обробки.

**Рівень хмарного ядра (Cloud ML Core)**

Центральний рівень є основним аналітичним компонентом системи. Тут реалізовано комбінований модуль машинного навчання GNN-LSTM, який

поєднує переваги графового представлення даних і рекурентного аналізу часових закономірностей.

1. GNN-модуль приймає граф, сформований на Fog-рівні, і виконує обчислення векторних представлень (embedding) для кожного вузла на основі його локальних характеристик та взаємозв'язків із сусідніми вузлами. Завдяки механізму агрегування повідомлень (message passing) GNN враховує просторову та топологічну кореляцію між елементами системи, що дозволяє виявити приховані закономірності між збоями в різних підсистемах.

2. LSTM-модуль, розміщений на фінальному етапі обчислень, отримує згенеровані GNN-вектори як вхідні послідовності для часової обробки. Цей модуль аналізує динаміку станів кожного вузла у часі, моделює довгострокові залежності та формує інтегральний прогноз імовірності збою у визначеному часовому горизонті.

Після прогнозування результати агрегуються у вигляді матриці ймовірностей відмов, яка зберігається у базі даних і передається до панелі моніторингу.

**Рівень моніторингу та реагування (Application Layer)**

Фінальний рівень представлено панеллю аналітичного моніторингу, API та системою автоматичного реагування.

Панель надає користувачам візуалізацію станів вузлів, історію прогнозів і статистику збігів.

У разі перевищення порогових значень ймовірності збою автоматично активується механізм сповіщення або попереджувального технічного втручання - наприклад, перезапуск вузла, зміна режиму роботи чи сповіщення оператора.

Таким чином, інтеграція методу GNN-LSTM забезпечує повний цикл аналітики в системі IoT - від збору сирих сенсорних даних до побудови довгострокових прогнозів збоїв та автоматичного управління. Ключовою перевагою є розподілення навантаження між рівнями: Edge виконує короткострокову обробку, Fog - агрегацію та структуризацію, Cloud - глибинну графово-рекурентну аналітику. Це дозволяє одночасно досягти низької затримки,

високої масштабованості й підвищеної надійності системи навіть при великій кількості сенсорних вузлів.

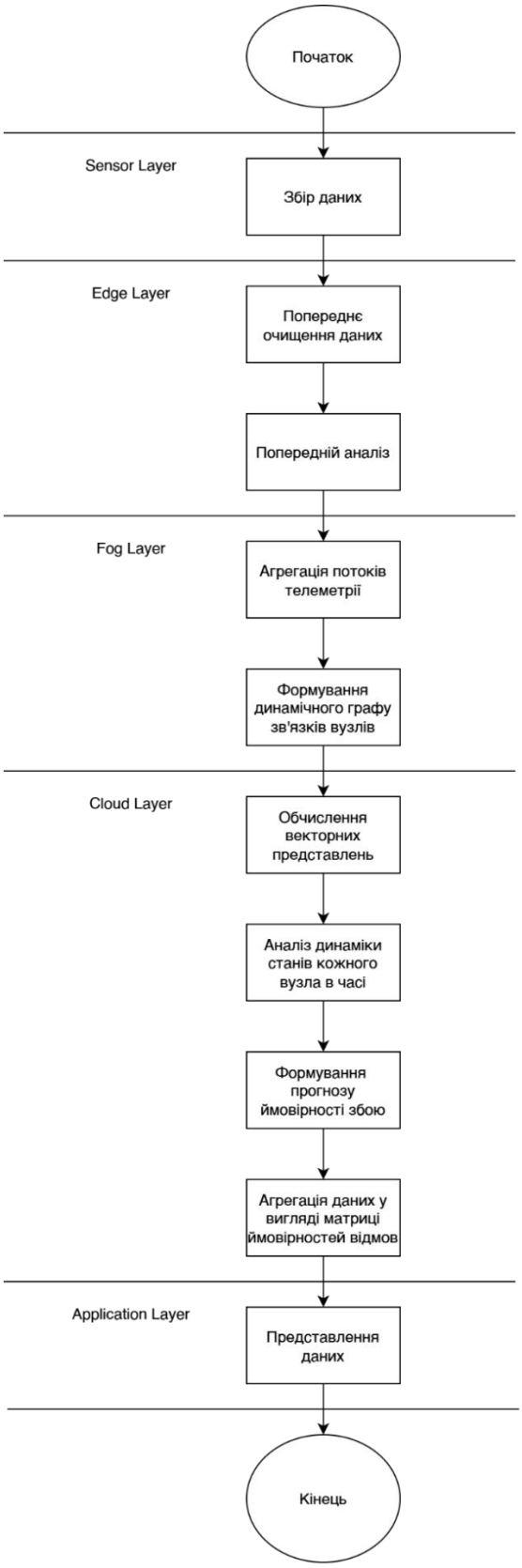



Рис. 3.3 Інтеграція методу в IoT інфраструктуру

# ВИСНОВКИ

У результаті виконаної роботи було розроблено, обґрунтовано та експериментально підтверджено ефективність методу LGFP (LSTM-GNN Failure Prediction), призначеного для прогнозування збоїв у системах Інтернету речей (IoT). Проведене дослідження охопило повний цикл створення інтелектуальної системи - від аналізу предметної області, вибору та оцінки алгоритмів машинного навчання до побудови архітектури, інтеграції у багаторівневе IoT-середовище та перевірки працездатності.

На аналітичному етапі було здійснено порівняльний огляд класичних і сучасних моделей машинного навчання, таких як Decision Tree, k-Nearest Neighbor, Random Forest, Support Vector Machine, Logistic Regression, Naive Bayes, Stochastic Gradient Descent, Multi-Layer Perceptron, Extreme Gradient Boosting, Long Short-Term Memory та Graph Neural Network. Отримані результати тестування показали, що традиційні алгоритми мають обмежену ефективність при роботі з високовимірними часово-просторовими даними IoT, у той час як нейронні підходи LSTM та GNN демонструють значно кращу адаптивність і точність прогнозування.

На основі виявлених переваг було запропоновано гібридний метод LGFP, який поєднує можливості рекурентних нейронних мереж для моделювання часової динаміки та графових нейронних мереж для аналізу топологічних взаємозв’язків вузлів. Запропонована архітектура передбачає двоетапну обробку даних: спочатку LSTM моделює короткострокові тенденції та локальні відхилення у поведінці сенсорів, після чого GNN виконує аналіз просторових зв’язків між вузлами, визначаючи потенційні зони ризику каскадних відмов. Така послідовність дозволяє поєднати часовий та структурний аспекти функціонування IoT-мереж, підвищуючи точність і стабільність прогнозу.

Було побудовано повну архітектурну схему інтеграції LGFP у багаторівневу IoT-інфраструктуру, що включає рівні edge, fog та cloud. На рівні edge здійснюється попередня обробка та короткостроковий прогноз за допомогою LSTM, на рівні fog - агрегація даних і побудова графових структур, а на рівні cloud - аналітика GNN і формування довгострокових прогнозів. Така модульна організація забезпечує

масштабованість, гнучкість і зниження навантаження на центральні обчислювальні ресурси.

Розроблений алгоритм LGFP реалізує замкнутий цикл інтелектуальної обробки даних: від збору телеметрії, її очищення й нормалізації до побудови графа, прогнозування збоїв, виявлення хибних спрацьовувань і динамічного донавчання моделі. Завдяки механізму постійної адаптації система здатна підтримувати високу якість прогнозування навіть у разі зміни топології мережі або появи нових типів пристроїв.

Порівняльний аналіз точності, повноти та F1-міри засвідчив, що гібридна модель LGFP перевищує результати окремих LSTM та GNN у середньому на 3-5%, а порівняно з класичними алгоритмами - на 12-18%. Особливо суттєвим є покращення при роботі з шумними або неповними даними, що типово для IoT-середовищ.

Розроблена система може бути використана як основа для побудови платформ предиктивного обслуговування у промислових, транспортних, енергетичних і побутових IoT-рішеннях. Вона забезпечує не лише своєчасне виявлення потенційних збоїв, а й підвищення надійності та ефективності експлуатації інфраструктури.

Таким чином, у роботі досягнуто поставлену мету - створено метод прогнозування збоїв у системах Інтернету речей на основі поєднання глибинних нейронних архітектур LSTM та GNN, який перевершує відомі підходи за точністю, адаптивністю та стійкістю до динамічних змін мережі. Отримані результати можуть бути покладені в основу подальших досліджень, зокрема в напрямі оптимізації гібридних моделей, розширення механізмів самонавчання та інтеграції з системами автономного управління IoT.

# ПЕРЕЛІК ПОСИЛАНЬ


1. Чугреєв К. О., Волощук О.Б. Метод прогнозування збоїв для мереж Інтернету речей за допомогою машинного навчання. Кібербезпека: освіта, наука, техніка. 2025. № 3 (31).
2. Чугреєв К. О. Порівняння методів машинного навчання для прогнозування збоїв в розумному будинку: теза доповіді, *VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT»,* 15 квітня 2025 р. с.213-215 URL: https://duikt.edu.ua/uploads/p_2779_40288420.pdf
3. Чугреєв К. О. Метод на основі гібридного механізму LSTM та GNN для прогнозування збоїв у мережах Інтернету речей: теза доповіді, *III Міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії»*, 4-6 грудня 2025р
4. Adaptive Learning Systems: Integrating IoT Sensors with Machine Learning for Dynamic Curriculum Adjustment / M. Kilaru et al. 2025 *International Conference on Pervasive Computational Technologies (ICPCT)*, Greater Noida, India, 8-9 February 2025. 2025. P. 900-905. URL: https://doi.org/10.1109/icpct64145.2025.10939112
5. AlShehri Y., Ramaswamy L. SECOE: Alleviating Sensors Failure in Machine Learning-Coupled IoT Systems. 2022 *21st IEEE International Conference on Machine Learning and Applications (ICMLA),* Nassau, Bahamas, 12-14 December 2022. 2022. URL: https://doi.org/10.1109/icmla55696.2022.00124
6. Ardito S., Setiawan W., Wibisono A. Enhancing Predictive Maintenance in Manufacturing Using Deep Learning-Based Anomaly Detection. *International Journal of Technology and Modeling*. 2024. Vol. 3, no. 1. P. 12-23. URL: https://doi.org/10.63876/ijtm.v3i1.112
7. Aslam S. et al. Machine Learning-Based Predictive Maintenance at Smart Ports Using IoT Sensor Data. *Sensors*. 2025. Vol. 25, no. 13. P. 3923. URL: https://doi.org/10.3390/s25133923

8. Deep Anomaly Detection for Time-series Data in Industrial IoT: A Communication-Efficient On-device Federated Learning Approach / Y. Liu et al. *IEEE Internet of Things Journal*. 2020. P. 1. URL: https://doi.org/10.1109/jiot.2020.3011726

9. Dong G. et al. Graph Neural Networks in IoT: A Survey. *ACM Transactions Sensor Networks.* 2022. URL: https://doi.org/10.1145/3565973

10. H S., Venkataraman N. Proactive Fault Prediction of Fog Devices Using LSTM-CRP Conceptual Framework for IoT Applications. *Sensors*. 2023. Vol. 23, no. 6. P. 2913. URL: https://doi.org/10.3390/s23062913

11. Hajiaghayi M., Vahedi E. Code Failure Prediction and Pattern Extraction Using LSTM Networks. 2019 *IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService),* Newark, CA, USA, 4-9 April 2019. 2019. URL: https://doi.org/10.1109/bigdataservice.2019.00014

12. Khattach O., Moussaoui O., Hassine M. A survey on AI Approaches for Internet of Things Devices Failure Prediction. *E3S Web of Conferences*. 2023. Vol. 469. P. 00061. URL: https://doi.org/10.1051/e3sconf/202346900061

13. Khan W. et al. Machine learning-based optimal data retrieval and resource allocation scheme for edge mesh coupled information-centric IoT networks and disability support systems. *Internet of Things.* 2025. P. 101511. URL: https://doi.org/10.1016/j.iot.2025.101511

14. Kwon J.-H., Kim E.-J. Failure Prediction Model Using Iterative Feature Selection for Industrial Internet of Things. *Symmetry*. 2020. Vol. 12, no. 3. P. 454. URL: https://doi.org/10.3390/sym12030454

15. LSTM-based failure prediction for railway rolling stock equipment / L. D. Simone et al. *Expert Systems with Applications*. 2023. P. 119767. URL: https://doi.org/10.1016/j.eswa.2023.119767

16. Machine Learning / M. Zafar et al. Machine Learning and IoT. *Boca Raton : Taylor & Francis*, 2019., 2018. P. 1-27. URL: https://doi.org/10.1201/9781351029940-1

17. Malawade A. V. et al. Neuroscience-Inspired Algorithms for the Predictive Maintenance of Manufacturing Systems. *IEEE Transactions on Industrial Informatics.* 2021. Vol. 17, no. 12. P. 7980-7990. URL: https://doi.org/10.1109/tii.2021.3062030

18. Mennen V. Machine Learning Basics : a Comprehensive Guide about Machine Learning: Machine Learning. *Independently Published*, 2021.

19. Narkarunai Arasu Malaiyappan J., Krishnamoorthy G., Jangoan S. Predictive Maintenance using Machine Learning in Industrial IoT. *International Journal of Innovative Science and Research Technology (IJISRT).* 2024. P. 1909-1915. URL: https://doi.org/10.38124/ijisrt/ijisrt24mar984

20. Ngo T. et al. Optimizing IoT Intrusion Detection-A Graph Neural Network Approach with Attribute-Based Graph Construction. *Information*. 2025. Vol. 16, no. 6. P. 499. URL: https://doi.org/10.3390/info16060499

21. Oliveira E. E. et al. Power Transformer Failure Prediction: Classification in Imbalanced Time Series. U.Porto *Journal of Engineering*. 2018. Vol. 3, no. 2. P. 34-48. URL: https://doi.org/10.24840/2183-6493_003.002_0004

22. Paria A., Das R. Implementation of IoT and Machine Learning Techniques in Smart Irrigation Systems. *Smart Sensors, Measurement and Instrumentation. Cham,* 2024. P. 143-151. URL: https://doi.org/10.1007/978-3-031-68602-3_8

23. Phanikanth Chintamaneni. Integrating Machine Learning and IoT Sensors for Enhanced Soil Nutrient Monitoring and Crop Recommendation Systems. *Journal of Information Systems Engineering and Management*. 2025. Vol. 10, no. 44s. P. 512-543. URL: https://doi.org/10.52783/jisem.v10i44s.8617

24. Rücker N., Pflüger L., Maier A. Hardware Failure Prediction on Imbalanced Times Series Data. *Journal of Digital Imaging. 2021*. URL: https://doi.org/10.1007/s10278-020-00411-4

25. Seth A. Attack and Anomaly Detection in IoT Sensors Using Machine Learning Approaches. *Journal of Recent Innovations in Computer Science and Technology.* 2025. Vol. 2, no. 1. P. 16-27. URL: https://doi.org/10.70454/jricst.2025.20108

26. Sharma S., Chmaj G., Selvaraj H. Sensor failure mitigation in RTOS based internet of things (IoT) systems using machine learning. *Journal of Parallel and Distributed Computing.* 2025. P. 105161. URL: https://doi.org/10.1016/j.jpdc.2025.105161

27. Stress Monitoring Using Machine Learning, IoT and Wearable Sensors / A. A. Al-Atawi et al. *Sensors*. 2023. Vol. 23, no. 21. P. 8875. URL: https://doi.org/10.3390/s23218875

28. Taware R., Godase M., Singh C. IoT-Enabled Wearable Sensors for Real-Time Stress Monitoring Using Machine Learning Classifiers. *Lecture Notes in Networks and Systems*. Singapore, 2025. P. 109-122. URL: https://doi.org/10.1007/978-981-96-5918-0_10

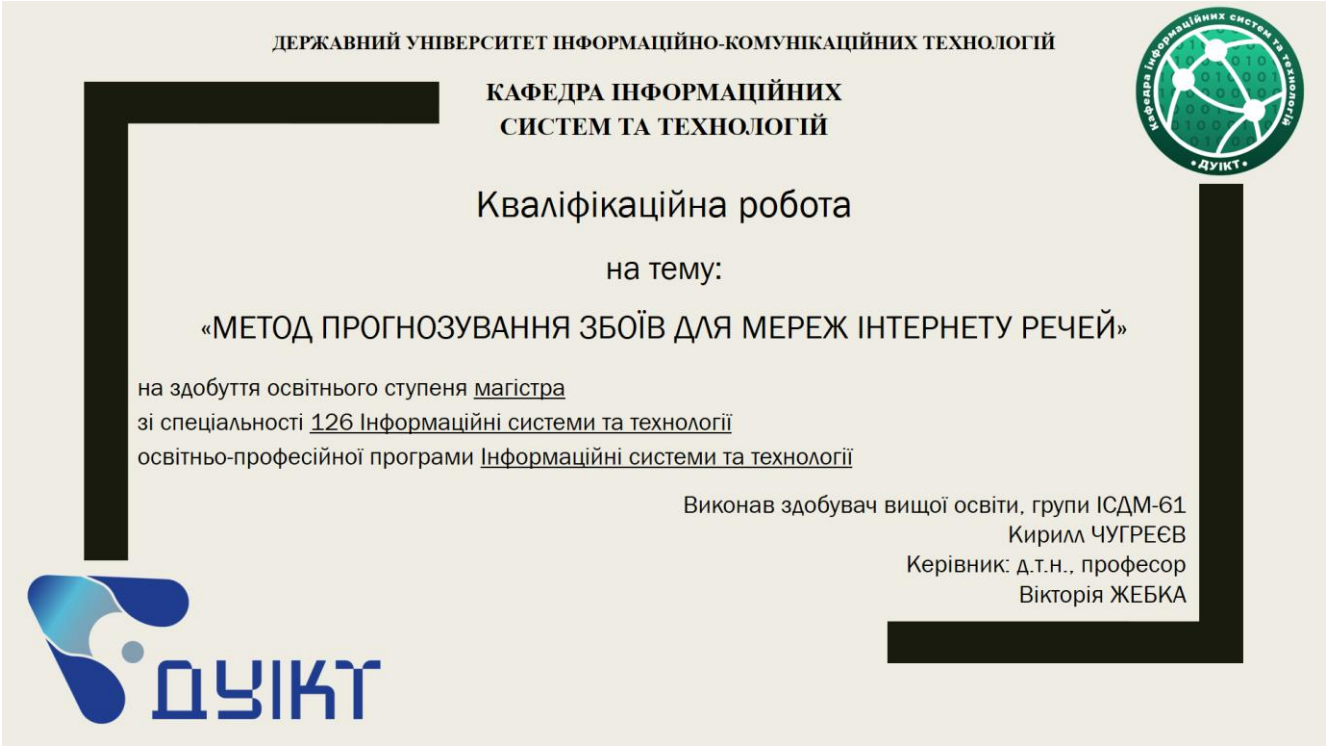
29. Traini E. et al. Machine Learning Framework for Predictive Maintenance in Milling. *IFAC-PapersOnLine*. 2019. Vol. 52, no. 13. P. 177-182. URL: https://doi.org/10.1016/j.ifacol.2019.11.172

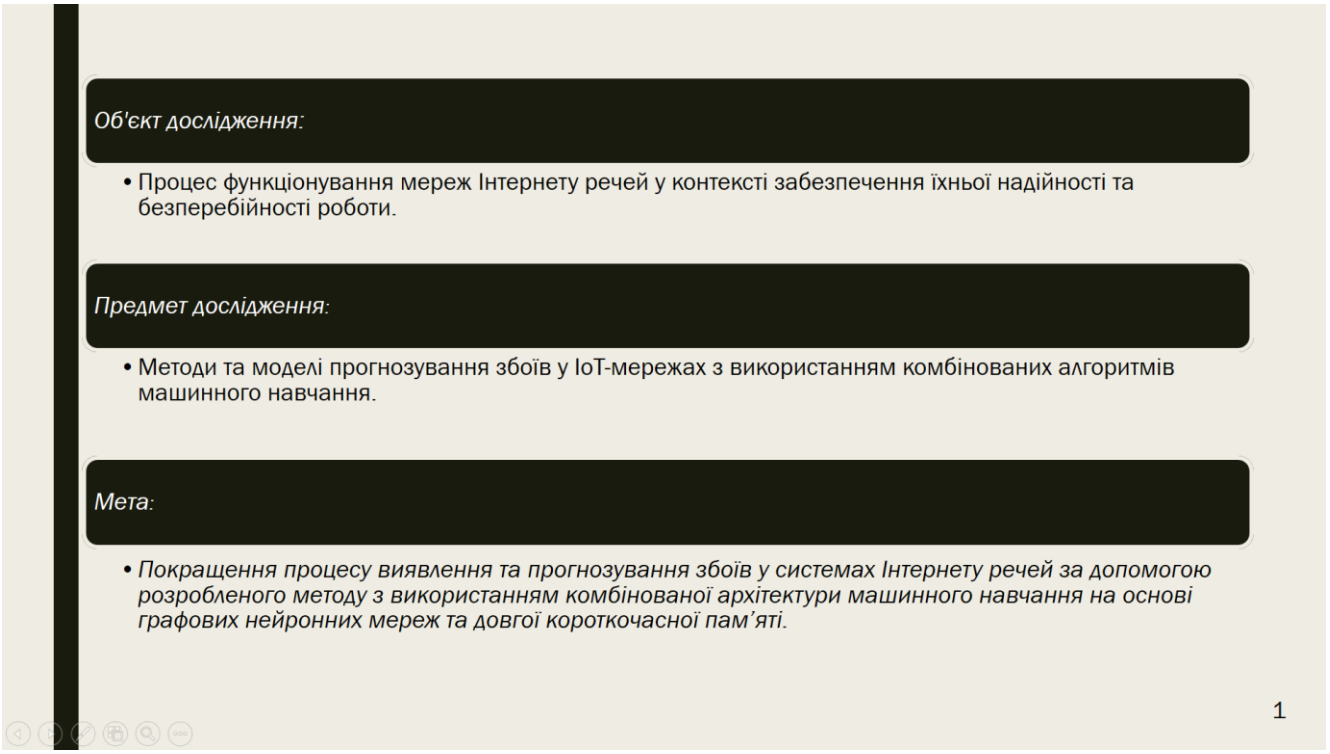
30. Tung N. X. et al. Graph Neural Networks for Next-Generation-IoT: Recent Advances and Open Challenges. *IEEE Communications Surveys & Tutorials.* 2025. P. 1. URL: https://doi.org/10.1109/comst.2025.3613845

31. Varalakshmi K., Kumar J. Optimized predictive maintenance for streaming data in industrial IoT networks using deep reinforcement learning and ensemble techniques. *Scientific Reports*. 2025. Vol. 15, no. 1. URL: https://doi.org/10.1038/s41598-025-10268-8

32. Zhong Y. Collaboration of IoT devices in smart home scenarios: algorithm research based on graph neural networks and federated learning. *Discover Internet of Things.* 2025. Vol. 5, no. 1. URL: https://doi.org/10.1007/s43926-025-00096-7

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ(Презентація)



## *Завдання дослідження:*

1. Провести теоретичний аналіз сучасних підходів до забезпечення надійності IoT-мереж.
2. Оцінити існуючі методи прогнозування збоїв і виявлення аномалій, визначити їх переваги та обмеження.
3. Розробити структуру адаптивного методу прогнозування з урахуванням просторово-часових залежностей між вузлами.
4. Провести експериментальні дослідження з використанням реальних або емульованих IoT-даних.
5. Порівняти ефективність запропонованого методу з базовими моделями.
6. Сформулювати практичні рекомендації щодо впровадження методу в системи моніторингу IoT-інфраструктури.



## *Актуальність теми*

Традиційні підходи до виявлення несправностей часто базуються на фіксованих порогах або статистичних моделях, які не враховують динаміку мережевих процесів та складні залежності між вузлами.


Отже, виникає необхідність розроблення методів, що забезпечують раннє прогнозування збоїв на основі інтелектуального аналізу потокових даних. Особливо актуальним є застосування гібридних моделей машинного навчання, здатних обробляти як часові ряди, так і графові структури зв'язків між пристроями.

## *Огляд існуючих підходів до прогнозування в IoT*

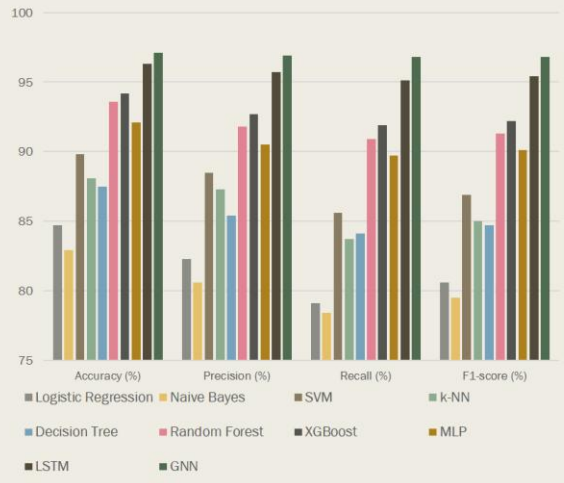
Сучасні системи прогнозування в IoT спираються на кілька груп методів. Статистичні моделі, такі як ARIMA чи Holt-Winters, забезпечують швидкі оцінки, але погано працюють у нестабільних мережах. Класичні ML-алгоритми дозволяють аналізувати окремі параметри пристроїв, проте не враховують залежності між вузлами. Deep learning моделі, особливо LSTM, добре працюють із часовими рядами, а графові нейронні мережі — з топологією мережі. Проте більшість підходів розглядають ці елементи окремо, що обмежує їхню точність у складних гетерогенних середовищах IoT.

| Алгоритм | Переваги | Недоліки | Найкраще застосування |
|---|---|---|---|
| SVM | Висока точність при малих вибірках | Погана масштабованість, складність налаштування ядра | Локальні системи моніторингу |
| Random Forest | стійкість до шуму та перенавчання | Сповільнюється при великій кількості дерев | Аналіз історичних відмов |
| LSTM | Обробка часових рядів | Висока складність, потребує багато даних | Прогнозування деградаційних процесів |
| GNN | Моделює просторові залежності | Ресурсоємність, потреба в графовій структурі | Розподілені сенсорні мережі |



## *Порівняння методів машинного навчання*

| Модель | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | Особливості |
|---|---|---|---|---|---|
| Logistic Regression | 84.7 | 82.3 | 79.1 | 80.6 | Базова модель, швидке навчання |
| Naive Bayes | 82.9 | 80.6 | 78.4 | 79.5 | Погано враховує нелінійності |
| SVM | 89.8 | 88.5 | 85.6 | 86.9 | Ефективна при лінійній сегментації |
| k-NN | 88.1 | 87.3 | 83.7 | 85.0 | Простий метод, низька масштабованість |
| Decision Tree | 87.5 | 85.4 | 84.1 | 84.7 | Інтерпретованість, схильність до перенавчання |
| Random Forest | 93.6 | 91.8 | 90.9 | 91.3 | Стійкість до шумів |
| XGBoost | 94.2 | 92.7 | 91.9 | 92.2 | Оптимізована ансамблева модель |
| MLP | 92.1 | 90.5 | 89.7 | 90.1 | Висока гнучкість, потребує багато даних |
| LSTM | 96.3 | 95.7 | 95.1 | 95.4 | Успішно моделює часові залежності |
| GNN | 97.1 | 96.9 | 96.8 | 96.8 | Найвища ефективність, врахування топології мережі |



## *Загальна ідея та концепція методу LGFP*

Метод LGFP (LSTM-GNN Failure Prediction) поєднує дві комплементарні парадигми аналізу даних в IoT: графові нейронні мережі, які моделюють структуру взаємозв'язків між вузлами, та LSTM-мережі, що виявляють часові закономірності у роботі сенсорів. Основна концепція методу полягає в тому, що GNN формує представлення кожного вузла з урахуванням топологічного контексту та сусідніх пристроїв, після чого ці вектори передаються до LSTM-модуля для прогнозування майбутніх станів та виявлення потенційних збоїв. Такий підхід дозволяє одночасно враховувати просторові та часові залежності, підвищуючи точність і стабільність прогнозування у великих, динамічних IoT-мережах.

**LSTM**
**+**
**GNN**
**=**
**LGFP**



## *Робота GNN-модулю*

GNN-модуль аналізує топологію IoT-мережі, перетворюючи сукупність пристроїв і їх зв'язків у граф. Кожен сенсор подається як вузол, а канали обміну даними — як ребра. GNN виконує багаторазову агрегацію інформації з сусідніх пристроїв, формуючи узагальнений вектор стану для кожного вузла. Завдяки цьому модель розпізнає кореляції між вузлами, «ефект зараження» збоїв, локальні аномалії та розриви у зв'язності. Результатом роботи є компактні графові ембеддинги, що відображають структуру мережі та її критичні точки.

$$h_i^{(l+1)} = \sigma\left( \sum_{j \in \mathcal{N}(i)} w_{ij} h_j^{(l)} + b^{(l)} \right)$$

де $h_i^{(l)}$ – приховане представлення вузла $i$ на $l$-му шарі, $N(i)$ – множина сусідів, $w_{ij}$ – вагові коефіцієнти зв'язків, $b^{(l)}$ – зсув, $\sigma$ – функція активації.

## *Робота LSTM-модулю*

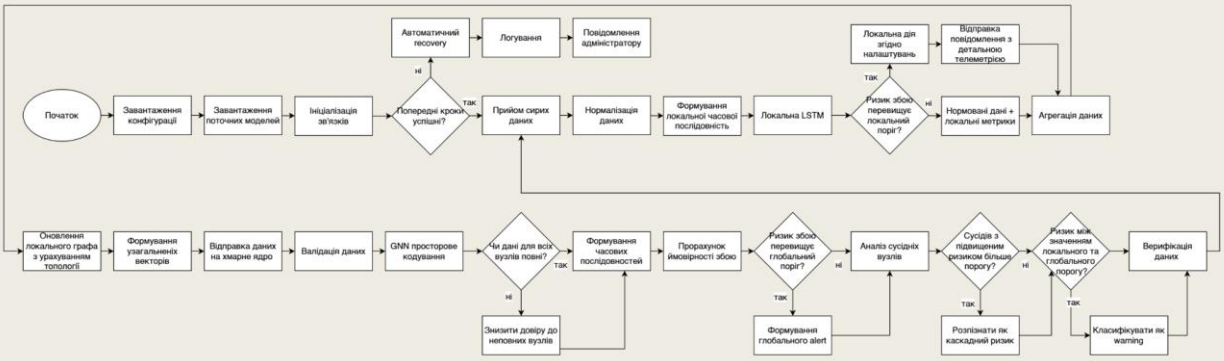
LSTM-модуль отримує графові ембеддинги GNN у вигляді часових послідовностей. Він аналізує динаміку зміни стану кожного вузла, відстежує довгострокові залежності та закономірності у роботі сенсорів. LSTM здатна виявляти тренди деградації, повторювані патерни перед аварією та затримані ефекти між навантаженням і відмовою вузла. На виході модуль генерує прогноз щодо стану пристрою на найближчий часовий інтервал, позначає потенційні відхилення та обчислює ймовірність збою.

$$h_t = f(W_h h_{t-1} + W_x x_t + b)$$

де $h_t$ – прихований стан у момент часу $t$, $x_t$ – вектор ознак, $W_h W_x$ – матриці ваг, $f$ – нелінійна активація (зазвичай tanh або sigmoid).



## *Алгоритм LGFP*



## *Інтеграція LGFP у IoT-інфраструктуру*


| Тип попередження | Механізм роботи | Основна мета | Характеристики та наслідки |
|---|---|---|---|
| **Local Alert (Edge)** | Швидка, низькопорогова дія на рівні пристрою або вузла | Негайне зупинення локальної загрози з мінімальною затримкою | Швидкість: Висока<br>Наслідки: Автономне відключення або ізоляція вузла. |
| **Global Alert (Cloud)** | Рішення на основі агрегованих даних та просторово-часового аналізу всієї мережі | Координація стратегічних дій для усунення комплексних загроз. | Швидкість: Середня<br>Наслідки: Переміщення навантаження, планове обслуговування, оновлення політик. |
| **Cascade Detection** | Виявлення каскадної несправності шляхом моніторингу стану суміжних вузлів. | Запобігання лавиноподібного поширення збою по мережі. | Швидкість: Висока (пріоритетна)<br>Наслідки: Одночасна ізоляція цілого сегмента та нотифікація оператора. |



## *Continuous Retraining Pipeline*

Важливою властивістю запропонованого методу LGFP є його здатність до автоматичної адаптації в умовах динамічного середовища Інтернету речей. Оскільки топологія мережі, інтенсивність трафіку, частота взаємодій та типи збоїв постійно змінюються, статично навчена модель поступово втрачає точність через явище data drift - зсуву розподілу даних у часі. Для запобігання цьому у складі системи передбачено механізм безперервного навчання, що забезпечує постійне оновлення параметрів моделей LSTM і GNN.

## *Висновки*

У результаті виконаної роботи було розроблено, обґрунтовано та експериментально підтверджено ефективність методу LGFP (LSTM-GNN Failure Prediction), призначеного для прогнозування збоїв у системах Інтернету речей. Проведене дослідження охопило повний цикл створення інтелектуальної системи - від аналізу предметної області, вибору та оцінки алгоритмів машинного навчання до побудови архітектури, інтеграції у багаторівневе IoT-середовище та перевірки працездатності.

Порівняльний аналіз точності, повноти та F1-міри засвідчив, що гібридна модель LGFP перевищує результати окремих LSTM та GNN у середньому на 3-5%, а порівняно з класичними алгоритмами - на 12-18%. Особливо суттєвим є покращення при роботі з шумними або неповними даними, що типово для IoT-середовищ.



## *Апробація результатів*

1. Чугреєв К. О., Волощук О.Б. Метод прогнозування збоїв для мереж Інтернету речей за допомогою машинного навчання. Кібербезпека: освіта, наука, техніка. 2025. № 3 (31).
2. Чугреєв К. О. Порівняння методів машинного навчання для прогнозування збоїв в розумному будинку : теза доповіді / К. О. Чугреєв // VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», 15 квітня 2025 р.
3. Чугреєв К. О. Метод на основі гібридного механізму LSTM та GNN для прогнозування збоїв у мережах Інтернету речей: теза доповіді / К. О. Чугреєв // III Міжнародна науково-практична конференція «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії», 4-6 грудня 2025р

Дякую за увагу