

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Олена НЕЧИПОРУК

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний комплекс діагностування технічних систем за допомогою
глибинного навчання

Виконавець: _____ Тимур ВОЙЦЕХІВСЬКИЙ

Керівник: _____ Наталія АПЕНЬКО

Нормоконтролер: _____ Євгеній ТУПОТА

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Факультет _____ комп'ютерних наук та технологій
Кафедра інтелектуальних кібернетичних систем _____
Спеціальність 126 “Інформаційні системи та технології” _____
Освітньо професійна програма «Інформаційні системи та технології» _____
Форма навчання _____ денна _____

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Олена НЕЧИПОРУК

« _____ » _____ 2025 р.

ЗАВДАННЯ на виконання кваліфікаційної роботи

_____ Войцехівського Тимура Михайловича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. **Тема проєкту (роботи):** Програмний комплекс діагностування технічних систем за допомогою глибинного навчання

затверджена наказом ректора від " 28 " _____ серпня 2025 року № 1575/ст

2. Термін виконання роботи: з 29.09.2025 до 31.12.2025

3. **Вихідні дані до роботи:** 1) наукова та навчально-методична література;

2) текстові та зображувальні матеріали за тематикою дослідження;

3) технічне завдання.

4. **Зміст пояснювальної записки** (перелік питань, що підлягають розробці):

1) теоретичні основи діагностування технічних систем на основі IoT-телеметрії та глибинного навчання;

2) інтеграція програмного модуля дігностування роботи IoT пристроїв в систему освітлення;

3) програмний модуль діагностування в системі управління мережею IoT-пристроїв.

5. **Перелік обов'язкового графічного матеріалу:**

1) компоненти системи дистанційного управління освітленням;

2) компоненти серверної частини системи управління пристроями освітлення;

3) вікна програмного модуля управління мережею IoT-пристроїв;

4) схема алгоритму обробки команд користувача в мережі IoT-пристроїв;

5) схема алгоритму діагностування роботи мережі IoT-пристроїв.

6. Календарний план

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1	Ознайомитись з постановкою задачі на кваліфікаційна робота	29.09.25 – 30.09.25	
2	Вивчити спеціальну літературу і технічну документацію	01.10.25 – 10.10.25	
3	Проаналізувати існуючі системи, основні поняття і їх створення	11.10.25 – 12.10.25	
4	Написати розділ 1	13.10.25 – 20.10.25	
5	Написати розділ 2	21.10.25 – 31.10.25	
6	Виконати розробку діагностичної системи	01.11.25 – 20.11.25	
7	Написати розділ 3	21.11.25 – 29.11.25	
8	Оформити пояснювальну записку	30.11.25 – 11.12.25	
9	Підготувати графічний демонстраційний матеріал	12.12.25 – 13.12.25	

7. Дата видачі завдання « 29 » вересня 2025 р.

Керівник кваліфікаційної роботи

_____ (підпис)

Наталія АПЕНЬКО

Завдання прийняв до виконання

_____ (підпис студента)

Тимур ВОЙЦЕХІВСЬКИЙ

РЕФЕРАТ

Пояснювальна записка до кваліфікаційного проєкту «Програмний комплекс діагностування технічних систем за допомогою глибинного навчання» містить 83 сторінки, 13 рисунків, 11 таблиць, список бібліографічних посилань використаних джерел з 32 найменувань, 1 додаток.

IoT-СИСТЕМА, ДІАГНОСТУВАННЯ СИСТЕМ, ПРОГРАМНИЙ МОДУЛЬ, НЕЙРОННА МЕРЕЖА, ПРОПУСКНА ЗДАТНІСТЬ, НАДІЙНІСТЬ, *SMART ENVIRONMENT*.

Мета дослідження – розробити програмний комплекс діагностування технічних систем на основі телеметричних даних *IoT* із використанням моделі глибинного навчання класу *LSTM Autoencoder* для виявлення аномалій у часових рядах параметрів стану.

Об’єкт дослідження – процес моніторингу та діагностування технічних систем на основі телеметрії, що надходить із розподіленої мережі *IoT*-пристроїв.

Предмет дослідження – методи та програмні засоби виявлення аномальних режимів роботи технічних систем за допомогою автоенкодерних нейронних мереж на багатовимірних часових рядах телеметричних параметрів.

Наукова новизна полягає у розробці методу діагностування технічних систем, який використовує *LSTM*-автоенкодер для виявлення аномалій у багатовимірній *IoT*-телеметрії без необхідності попереднього маркування аварійних станів, із формуванням інтегрального діагностичного показника на основі помилки реконструкції та адаптивного порогоування.

Практичне значення роботи полягає у можливості застосування розробленого програмного комплексу для раннього виявлення деградації та нестабільних режимів у технічних системах, що контролюються *IoT*-мережею, а також для підтримки рішень щодо профілактичного обслуговування та підвищення експлуатаційної надійності. В основу комплексу може бути покладена існуюча інфраструктура збору та аналізу метрик, описана у вихідній роботі, із додаванням нейромережевого діагностичного модуля.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	6
ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ДІАГНОСТУВАННЯ ТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ <i>IoT</i> -ТЕЛЕМЕТРІЇ ТА ГЛИБИННОГО НАВЧАННЯ	10
1.1. Технічні системи як об’єкти діагностування.....	10
1.2. Поняття та характеристика <i>IoT</i> -систем	16
1.3. Ключові параметри продуктивності в <i>IoT</i> -мережах та їх вплив на діагностування	24
1.4. Аналіз сучасних технологій та готових рішень дистанційного управління на прикладі систем освітлення	34
1.5. Висновки до розділу	39
РОЗДІЛ 2 ІНТЕГРАЦІЯ ПРОГРАМНОГО МОДУЛЯ ДІАГНОСТУВАННЯ РОБОТИ <i>IoT</i> ПРИСТРОЇВ В СИСТЕМУ ОСВІТЛЕННЯ	41
2.1. Загальна архітектура програмного комплексу діагностування.....	43
2.2. Розробка модуля попередньої обробки даних.....	48
2.3. Проєктування та реалізація нейронної мережі <i>LSTM Autoencoder</i> ..	54
2.4. Висновки до розділу	57
РОЗДІЛ 3 ПРОГРАМНИЙ МОДУЛЬ ДІАГНОСТУВАННЯ В СИСТЕМІ УПРАВЛІННЯ МЕРЕЖЕЮ <i>IoT</i> -ПРИСТРОЇВ	59
3.1. Опис експериментального стенду та вихідних даних.....	59
3.2. Діагностування системи освітлення.....	62
3.3. Тестування мобільного застосунку для управління пристроями освітлення	68
3.4. Експериментальне дослідження програмного модуля діагностування	75
3.5. Висновки до розділу	77
ВИСНОВКИ.....	79
СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81
ДОДАТОК А.....	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

DHCP – Dynamic Host Configuration Protocol

FTTH – Fiber To The Home

MAC – Media Access Control

OLT – Optical Line Terminal

ONU – Optical Network Unit

PON – Passive Optical Network

QoS – Quality of Service

RCI MDU – RCI Multi-Dwelling Unit

Rx – Receive

SC/UPC – Subscriber Connector / Ultra Physical Contact

SFP – Small Form-factor Pluggable

SNMP – Simple Network Management Protocol

Tx – Transmit

VLAN – Virtual Local Area Network

WDM – Wavelength Division Multiplexing

ДБЖ – Джерело безперебійного живлення

ВСТУП

В умовах цифровізації інженерних об'єктів та переходу до кіберфізичних систем ключовим стає не лише керування технічними пристроями, а й своєчасне виявлення деградації, прихованих відмов і передвідмовних станів. Багато технічних систем – від інженерних мереж будівель до промислових вузлів – працюють у режимі тривалої експлуатації, де критичним є раннє попередження про аномальні режими, що передують відмовам.

Поширення *IoT* призвело до зростання обсягів телеметрії, яка відображає поточний стан системи в режимі близькому до реального часу. Проте класичні підходи технічної діагностики часто потребують або детальних фізичних моделей об'єкта, або маркованих наборів даних із прикладами відмов, що на практиці є складним – відмови є рідкісними, різноманітними, а їх фіксація потребує ресурсів і часу. Це формує потребу у методах, які здатні будувати діагностичні рішення на основі даних нормальної роботи та автоматично виявляти відхилення без попереднього маркування аварійних станів.

Перспективним напрямом є застосування глибинного навчання, зокрема автоенкодерних моделей, для виявлення аномалій у багатовимірних часових рядах телеметрії. Логіка таких підходів полягає у навчанні моделі відтворювати «нормальну» динаміку показників, а діагностичним критерієм стає помилка реконструкції, яка зростає при появі нехарактерної поведінки системи. Для *IoT*-систем цей підхід є особливо доречним, оскільки телеметрія природно формується як часовий ряд параметрів мережі, обчислювального навантаження, якості зв'язку, енергоспоживання та доступності компонентів.

Актуальність теми зумовлена необхідністю підвищення надійності та безпеки технічних систем, що експлуатуються в умовах змінних навантажень і нестабільних комунікацій, а також потребою у програмних комплексах, здатних здійснювати інтелектуальну діагностику в режимі реального часу на основі потоків *IoT*-даних. На практиці це дозволяє скоротити час простою, зменшити витрати на обслуговування та підвищити якість прийняття рішень у експлуатації.

Мета дослідження – розробити програмний комплекс діагностування

технічних систем на основі телеметричних даних *IoT* із використанням моделі глибинного навчання класу *LSTM Autoencoder* для виявлення аномалій у часових рядах параметрів стану.

Об'єкт дослідження – процес моніторингу та діагностування технічних систем на основі телеметрії, що надходить із розподіленої мережі *IoT*-пристроїв.

Предмет дослідження – методи та програмні засоби виявлення аномальних режимів роботи технічних систем за допомогою автоенкодерних нейронних мереж на багатовимірних часових рядах телеметричних параметрів.

Завдання дослідження:

- проаналізувати підходи до технічної діагностики в умовах *IoT*-моніторингу та визначити вимоги до програмного комплексу діагностування;
- сформуванати набір інформативних телеметричних параметрів (метрик) і методику їх попередньої обробки для задачі діагностики;
- розробити модель *LSTM Autoencoder* для навчання на нормальних режимах роботи та визначити діагностичний критерій на основі помилки реконструкції;
- обґрунтувати метод встановлення порогів аномальності та правила інтерпретації результатів як діагностичних повідомлень;
- спроектувати архітектуру програмного комплексу (збір – обробка – *inference* – інтерфейс) та інтегрувати модуль нейромережевої діагностики;
- провести експериментальне дослідження якості діагностування (точність виявлення, частота хибних спрацювань, чутливість до деградаційних сценаріїв) і подати результати у вигляді таблиць та графіків.

Наукова новизна полягає у розробці методу діагностування технічних систем, який використовує *LSTM*-автоенкодер для виявлення аномалій у багатовимірній *IoT*-телеметрії без необхідності попереднього маркування аварійних станів, із формуванням інтегрального діагностичного показника на основі помилки реконструкції та адаптивного порогоування.

Практичне значення роботи полягає у можливості застосування розробленого програмного комплексу для раннього виявлення деградації та нестабільних режимів у технічних системах, що контролюються *IoT*-мережею, а також для підтримки рішень щодо профілактичного обслуговування та підвищення

експлуатаційної надійності. В основу комплексу може бути покладена існуюча інфраструктура збору та аналізу метрик, описана у вихідній роботі, із додаванням нейромережевого діагностичного модуля.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ДІАГНОСТУВАННЯ ТЕХНІЧНИХ СИСТЕМ НА ОСНОВІ ІОТ-ТЕЛЕМЕТРІЇ ТА ГЛИБИННОГО НАВЧАННЯ

1.1. Технічні системи як об'єкти діагностування

1.1.1. Поняття технічної системи та її функціональні стани

Технічна система розглядається як сукупність взаємопов'язаних елементів апаратного, програмного та інформаційного характеру, які функціонують спільно з метою виконання визначених задач у заданих умовах експлуатації. До таких систем належать інженерні мережі, промислові комплекси, автоматизовані установки, інформаційно-керувальні системи, а також сучасні кіберфізичні та IoT-орієнтовані рішення. Важливою особливістю технічних систем є наявність внутрішньої структури, взаємодії між компонентами та залежність їх поведінки від зовнішнього середовища і режимів навантаження.

Функціонування технічної системи відбувається у часі та характеризується зміною її станів. Під станом технічної системи розуміють сукупність значень параметрів, що описують її працездатність, ефективність і відповідність заданим вимогам у конкретний момент часу. Такі параметри можуть включати фізичні величини, обчислювальні показники, мережеві характеристики, енергоспоживання та інші вимірювані або обчислювані ознаки, які відображають поточну поведінку системи.

У теорії технічної діагностики традиційно виділяють нормальний функціональний стан, за якого система виконує всі задані функції в межах встановлених допусків. Для цього стану характерна стабільність параметрів, передбачувана динаміка та відсутність критичних відхилень.

<i>Кафедра ІКС</i>				<i>КАІ 25 02 90 000 ПЗ</i>			
<i>Виконав</i>	<i>Войцехівський Т.М.</i>			<i>Теоретичні основи діагностування технічних систем на основі IoT-телеметрії та глибинного навчання</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	<i>10</i>	<i>83</i>
<i>Консульт.</i>					<i>M-126-24-1-IT</i>		
<i>Норм. контр.</i>	<i>Тупота С.В.</i>						
<i>Зав. Каф.</i>	<i>Нечипорук О.П.</i>						

Саме нормальний стан є базовим для формування еталонних моделей поведінки системи, на основі яких здійснюється подальше діагностування та виявлення аномалій.

Окрім нормального стану, у процесі експлуатації технічна система може переходити у стан деградації, який характеризується поступовим погіршенням окремих параметрів без повної втрати працездатності. Деградація може бути зумовлена фізичним зношенням компонентів, накопиченням помилок у програмному забезпеченні, зростанням навантаження, впливом зовнішніх факторів або нестабільністю комунікацій. Важливою особливістю деградаційних процесів є їх повільний і часто прихований характер, що ускладнює виявлення традиційними пороговими методами.

Наступним функціональним станом є передвідмовний стан, у якому система ще формально виконує свої функції, проте її параметри наближаються до граничних значень. У цьому стані різко зростає ймовірність виникнення відмови, а будь-яке додаткове навантаження або зовнішній вплив можуть призвести до порушення роботи. Передвідмовний стан є критично важливим з точки зору експлуатації, оскільки саме на цьому етапі можливе ефективне профілактичне втручання.

Таблиця 1.1

Класифікація основних видів відмов у технічних системах

Тип відмови	Джерело виникнення	Характер прояву	Потенційні наслідки
Фізична	Апаратні компоненти	Поступовий або раптовий	Втрата працездатності вузла
Інформаційна	Програмне забезпечення	Нестабільна поведінка	Некоректні рішення системи
Мережева	Канали зв'язку	Зростання затримок, втрати пакетів	Порушення обміну даними
Енергетична	Система живлення	Зниження напруги, розряд	Часткова або повна зупинка

Відмова технічної системи визначається як стан, за якого система втрачає здатність виконувати хоча б одну з заданих функцій відповідно до встановлених вимог. Відмови можуть мати різну природу, бути повними або частковими, раптовими або такими, що є результатом накопиченої деградації. У сучасних складних системах, зокрема *IoT*-орієнтованих, відмова одного компонента не завжди призводить до повної зупинки системи, але може суттєво вплинути на її ефективність і безпеку.

Особливістю сучасних технічних систем є їх нелінійна та стохастична поведінка, обумовлена великою кількістю взаємодіючих компонентів, асинхронністю процесів і мінливістю зовнішніх умов. Це призводить до того, що межі між нормальним, деградаційним і передвідмовним станами часто є розмитими. У таких умовах класичні жорстко визначені критерії діагностики виявляються недостатньо ефективними.

Саме тому в задачах діагностування сучасних технічних систем дедалі більшого значення набувають підходи, що базуються на аналізі поведінкових моделей і часової динаміки параметрів. Замість фіксованого визначення станів за окремими порогоми використовується оцінка відхилення поточної поведінки системи від її типової нормальної динаміки. Такий підхід створює передумови для застосування методів машинного та глибинного навчання, зокрема автоенкодерних нейронних мереж, які здатні формувати узагальнену модель нормального функціонування та виявляти аномальні режими на ранніх етапах.

1.1.2. Основні види відмов і деградацій у технічних системах

У процесі експлуатації технічні системи піддаються впливу різноманітних факторів, які можуть призводити до порушення їх нормального функціонування. Ці порушення проявляються у вигляді відмов або деградацій, що відрізняються за причинами виникнення, швидкістю розвитку та наслідками для роботи системи. Аналіз видів відмов є фундаментальною складовою технічної діагностики, оскільки дозволяє сформувати набір інформативних ознак для виявлення аномальних станів.

Фізичні відмови пов'язані з механічним або електронним зношенням компонентів технічної системи. До них належать пошкодження апаратних елементів, деградація контактів, старіння матеріалів, перегрів, вплив вологи або пилу. Такі відмови часто мають поступовий характер і супроводжуються зміною фізичних параметрів, що може бути зафіксовано за допомогою сенсорів температури, струму, напруги або вібрацій.

Інформаційні відмови виникають у програмному забезпеченні або логіці керування системою. Вони можуть проявлятися у вигляді помилок обробки даних, некоректної інтерпретації сигналів, накопичення збоїв у пам'яті, витоків ресурсів або некоректної взаємодії між програмними модулями. Особливістю інформаційних відмов є їх складна відтворюваність та залежність від тривалості роботи системи і сценаріїв навантаження.

Мережеві відмови характерні для розподілених технічних систем, зокрема *IoT*-орієнтованих. Вони пов'язані з порушенням передачі даних між компонентами системи внаслідок втрат пакетів, зростання затримок, нестабільності каналів зв'язку або перевантаження мережевої інфраструктури. Такі відмови не завжди призводять до повної зупинки системи, проте суттєво впливають на якість обслуговування та достовірність зібраної телеметрії.

Таблиця 1.2

Відображення деградаційних процесів у телеметрії *IoT*-систем

Параметр телеметрії	Типова деградація	Діагностичний прояв
Затримка передачі	Перевантаження мережі	Повільне зростання <i>latency</i>
Втрати пакетів	Нестабільний канал	Поява періодичних піків
Навантаження <i>CPU</i>	Програмна деградація	Тренд до зростання
Рівень живлення	Розряд акумулятора	Поступове падіння напруги

Енергетичні відмови виникають унаслідок нестабільного або недостатнього живлення компонентів технічної системи. Для *IoT*-пристроїв, що працюють автономно, критичним фактором є зниження заряду акумуляторів, деградація елементів живлення або порушення режимів енергозбереження. Такі відмови часто

мають прихований характер і можуть проявлятися через зменшення частоти передачі даних або збільшення затримок реакції системи.

Окрему категорію становлять деградаційні процеси, які не є відмовами у класичному розумінні, але призводять до поступового погіршення характеристик системи. Деградація може торкатися як апаратних, так і програмних компонентів, а також комунікаційних каналів. Найбільш небезпечним є те, що деградація часто не супроводжується різкими змінами параметрів, а відхилення накопичуються поступово, що ускладнює їх виявлення традиційними методами.

Для задач інтелектуальної діагностики важливим є розуміння того, що різні види відмов і деградацій можуть проявлятися у телеметричних даних у вигляді складних багатовимірних патернів. Один і той самий тип відмови може по-різному відображатися в залежності від контексту роботи системи, що потребує аналізу часової динаміки параметрів, а не лише їх миттєвих значень.

1.1.3. Класичні підходи до технічної діагностики

Класичні методи технічної діагностики формувалися в умовах відносно простих і добре формалізованих технічних систем, для яких були відомі фізичні моделі процесів і допустимі межі параметрів. Основною метою таких методів є визначення поточного стану системи та виявлення відхилень від нормального режиму на основі вимірюваних показників.

Таблиця 1.3

Порівняння класичних методів технічної діагностики

Метод	Принцип роботи	Переваги	Обмеження
Пороговий	Порівняння з граничними значеннями	Простота, швидкодія	Нечутливість до деградацій
Модельний	Порівняння з математичною моделлю	Висока точність	Складність побудови
Статистичний	Аналіз розподілів і трендів	Виявлення відхилень	Погано працює з нестационарністю

У практиці експлуатації ці підходи й досі широко застосовуються завдяки простоті реалізації та зрозумілій інтерпретації результатів.

Найпоширенішими є порогові методи діагностики, які ґрунтуються на порівнянні значень параметрів із наперед заданими граничними рівнями. Якщо параметр виходить за межі допустимого інтервалу, фіксується аномалія або відмова. Перевагою такого підходу є його низька обчислювальна складність та можливість швидкого впровадження навіть у системах з обмеженими ресурсами. Водночас порогові методи є малочутливими до повільних деградацій і не враховують взаємозв'язок між параметрами.

Модельно-орієнтовані методи базуються на використанні математичних або фізичних моделей технічної системи, які описують її нормальну поведінку. Діагностика здійснюється шляхом порівняння фактичних вимірювань із результатами моделювання. Відхилення між моделлю та реальними даними інтерпретуються як ознака несправності. Такий підхід дозволяє досягти високої точності, проте вимагає глибокого знання внутрішньої структури системи та є складним для масштабування у випадку гетерогенних або динамічно змінних систем.

Статистичні методи діагностики використовують аналіз розподілів, кореляцій та трендів параметрів у часових рядах. До них належать контрольні карти, аналіз середніх значень, дисперсій та автокореляційних функцій. Ці методи дозволяють виявляти відхилення від типової поведінки без необхідності жорсткого задання порогів. Проте їх ефективність знижується у випадку нестационарних процесів і багатовимірних даних, характерних для сучасних *IoT*-систем.

У складних розподілених системах, де одночасно взаємодіють десятки або сотні параметрів, класичні підходи стикаються з проблемою масштабованості. Аналіз кожного параметра окремо не дозволяє врахувати комплексний характер деградаційних процесів, коли аномалія проявляється не у зміні одного показника, а у зміні їхньої взаємної динаміки. Це призводить до зростання кількості хибних спрацювань або, навпаки, до пропуску критичних станів.

Додатковим обмеженням класичних методів є необхідність ручного налаштування параметрів діагностики. Встановлення порогів, побудова моделей

або вибір статистичних критеріїв часто виконуються експертами і залежать від конкретних умов експлуатації. У реальних системах ці умови можуть змінюватися з часом, що потребує регулярного перегляду налаштувань і знижує адаптивність діагностичних рішень.

У контексті *IoT*-орієнтованих технічних систем класичні методи виявляються недостатньо гнучкими через високу варіативність даних, наявність шумів, пропусків та асинхронності вимірювань. Це стимулює пошук підходів, здатних автоматично формувати модель нормальної поведінки системи без явного задання правил і порогів, а також адаптуватися до змін умов експлуатації.

Таблиця 1.4

Обмеження класичних методів у *IoT*-системах

Характеристика <i>IoT</i>	Вплив на діагностику
Багатовимірність	Неможливість аналізу параметрів ізольовано
Нестаціонарність	Зміна нормальних режимів
Шум і пропуски	Зростання хибних спрацювань
Масштабованість	Складність ручного налаштування

1.2. Поняття та характеристика *IoT*-систем

Інтернет речей (англ. *Internet of Things*, *IoT*) є концепцією побудови розподілених інформаційно-технічних систем, у межах яких фізичні об'єкти оснащуються вбудованими обчислювальними модулями, сенсорними елементами, засобами зв'язку та керування з метою автоматизованого збору, обміну й обробки даних без безпосередньої участі людини. Ключовою ідеєю *IoT* є створення безперервного інформаційного потоку між фізичним середовищем і цифровими системами, що відкриває можливості для моніторингу, аналізу та інтелектуального керування технічними об'єктами в режимі, близькому до реального часу [3].

Розвиток концепції Інтернету речей відбувався поетапно та був тісно пов'язаний із еволюцією засобів ідентифікації, зв'язку та обробки даних. Початкові передумови *IoT* сформувалися ще у 1990-х роках із появою технологій радіочастотної ідентифікації (*RFID*) та рішень класу *machine-to-machine* (*M2M*), які

дозволяли автоматизувати обмін інформацією між пристроями. Використання *RFID* стало основою для ідентифікації та відстеження об'єктів у логістиці, промисловості та складських системах, що згодом трансформувалося у ширшу парадигму взаємодії “речей”.

Формалізація терміна *Internet of Things* відбулася наприкінці ХХ століття, коли Кевін Ештон у 1999 році запропонував концепцію світу, в якому фізичні об'єкти здатні самостійно збирати та передавати інформацію про свій стан. Подальший розвиток мережевих технологій, мікроелектроніки та програмного забезпечення призвів до того, що *IoT* поступово вийшов за межі задач ідентифікації й перетворився на універсальну платформу для побудови розподілених кіберфізичних систем.

У 2010-х роках концепція Інтернету речей зазнала якісного стрибка завдяки інтеграції з хмарними обчисленнями, технологіями обробки великих даних та методами машинного навчання. Масове підключення сенсорних пристроїв призвело до формування значних обсягів телеметричної інформації, аналіз якої став неможливим без автоматизованих інтелектуальних методів. Саме на цьому етапі *IoT* перетворюється з інфраструктури збору даних на основу для аналітичних і прогностичних систем.

Важливу роль у поширенні *IoT* відіграв розвиток бездротових технологій зв'язку та доступних апаратних платформ. Поява енергоефективних протоколів і недорогих мікроконтролерів дозволила створювати автономні пристрої, здатні працювати тривалий час без обслуговування та бути інтегрованими у великі мережі. Це суттєво розширило сфери застосування *IoT* – від побутових і медичних пристроїв до промислових, транспортних і енергетичних систем.

На сучасному етапі Інтернет речей розглядається як складова частина кіберфізичних систем і концепції Індустрії 4.0. *IoT* забезпечує зв'язок між фізичними процесами та цифровими моделями, формуючи основу для цифрових двійників, автономних виробничих ліній, “розумних” міст і адаптивних технічних комплексів. У такому контексті *IoT* виступає не лише джерелом даних, а й активним елементом системи управління, здатним до самоаналізу та оптимізації [4].

Архітектура *IoT*-систем зазвичай реалізується у вигляді багаторівневої структури, що дозволяє розділити функції збору, передачі, обробки та використання даних. Такий підхід забезпечує масштабованість, гнучкість і можливість інтеграції різнорідних технологій у єдине інформаційне середовище. Типова архітектура включає кілька логічних рівнів, кожен з яких відповідає за певний етап життєвого циклу даних.

Рівень пристроїв (*Perception Layer*) відповідає за безпосередній контакт із фізичним середовищем та формування первинних даних за допомогою сенсорів і виконавчих механізмів. Мережевий рівень (*Network Layer*) забезпечує надійну передачу телеметричної інформації до центрів обробки з урахуванням вимог безпеки та якості обслуговування [5]. Рівень обробки даних (*Data Processing Layer*) реалізує зберігання, агрегацію та аналітику даних, включаючи застосування методів машинного та глибинного навчання. Рівень застосувань (*Application Layer*) надає інтерфейси взаємодії з користувачем і зовнішніми системами, а рівень управління та безпеки забезпечує контроль доступу, оновлення та захист *IoT*-інфраструктури [6].

Узгоджена взаємодія зазначених рівнів формує замкнений цикл “вимірювання – аналіз – прийняття рішення – вплив”, що створює передумови для побудови інтелектуальних систем діагностування. Саме на рівні обробки даних виникає необхідність застосування нейромережевих методів, здатних аналізувати багатовимірні часові ряди та виявляти складні аномальні й деградаційні процеси, що є предметом подальшого дослідження в цій роботі.

1.2.1. Структура *IoT*-систем та особливості формування телеметрії

IoT-системи являють собою багаторівневі розподілені комплекси, що поєднують фізичні пристрої, мережеву інфраструктуру та програмні засоби обробки даних. Їх ключовою функцією є безперервний збір телеметричної інформації про стан контрольованих об’єктів і передачу цієї інформації для подальшого аналізу та прийняття рішень. На відміну від традиційних автоматизованих систем, *IoT*-архітектури орієнтовані на масштабованість,

асинхронну взаємодію та роботу в умовах обмежених ресурсів на периферійних вузлах.

Типова *IoT*-система складається з рівня фізичних пристроїв, рівня передачі даних та рівня обробки і представлення інформації. На нижньому рівні розташовані сенсори та виконавчі пристрої, які здійснюють вимірювання параметрів середовища або стану технічних компонентів. Ці пристрої часто мають обмежені обчислювальні можливості та енергетичні ресурси, що визначає характер телеметрії — невеликі пакети даних, які передаються з певною періодичністю або за подіями.

Мережевий рівень *IoT*-системи забезпечує транспортування телеметричних даних від пристроїв до серверної інфраструктури. Передача може здійснюватися за допомогою різних протоколів і технологій зв'язку, що відрізняються за пропускну здатністю, затримками та енергоефективністю. Унаслідок цього телеметричні потоки часто є асинхронними, нерівномірними у часі та схильними до втрат або затримок, що ускладнює їх безпосередній аналіз.

На серверному рівні здійснюється агрегація, зберігання та попередня обробка телеметрії. Дані надходять від великої кількості пристроїв і формують багатовимірні часові ряди, у яких кожен вимір відповідає певному параметру стану системи. У сучасних *IoT*-платформах для цього використовуються бази даних часових рядів, черги повідомлень і потокові обчислювальні модулі, що дозволяють працювати з даними в режимі близькому до реального часу.

Особливістю телеметрії *IoT*-систем є її висока варіативність і залежність від умов експлуатації. Частота вимірювань може змінюватися в залежності від режимів роботи, а самі дані можуть містити шум, пропуски або некоректні значення. Крім того, телеметрія різних пристроїв часто не синхронізована за часом, що ускладнює формування єдиного представлення стану системи у певний момент.

З точки зору діагностики технічних систем телеметрія виступає первинним джерелом ознак, на основі яких робиться висновок про стан об'єкта. Проте ці ознаки не завжди мають прямий фізичний сенс у відриві від контексту, а їх інформативність проявляється лише при аналізі спільної динаміки у часі. Це особливо характерно для деградаційних процесів, коли аномалія не проявляється у

вигляді різкого стрибка одного параметра, а формується як зміна взаємозв'язків між кількома показниками.

У традиційних підходах телеметричні дані часто піддаються спрощенню шляхом агрегації або усереднення, що призводить до втрати інформації про часову структуру процесів. Для задач інтелектуальної діагностики такий підхід є недостатнім, оскільки він не дозволяє виявляти ранні ознаки нестабільної поведінки системи. Натомість сучасні методи аналізу орієнтуються на збереження повної часової динаміки та багатовимірного характеру телеметрії.

1.2.2. Телеметричні параметри як діагностичні ознаки

Телеметричні параметри в *IoT*-системах виступають основними джерелами інформації про стан технічних об'єктів і систем. У задачах діагностики важливо правильно обрати параметри, які найбільш точно відображають зміни у функціональному стані об'єкта, та врахувати їх взаємозв'язок. Оскільки системи *IoT* зазвичай працюють у реальному часі та генерують великий обсяг даних, необхідно також враховувати, як ефективно обробляти й аналізувати ці дані для виявлення аномальних ситуацій.

Основними типами телеметрії є параметри, що описують стан об'єкта на рівні його функціонування (наприклад, температуру, тиск, напругу) та характеристики, що відображають зовнішні умови або середовище, в якому система працює (вологість, світлові умови, рівень шуму). Крім того, для *IoT*-систем важливим є моніторинг параметрів, які стосуються самих пристроїв, як-от рівень зарядки батареї, потужність сигналу, кількість оброблених даних тощо.

Для ефективного діагностування важливим є врахування динамічних властивостей цих параметрів. У багатьох випадках відмови системи не є миттєвими і не викликають різких змін у конкретних параметрах. Замість цього спостерігається поступове погіршення стану, що вимагає аналізу часових рядів та взаємозв'язків між параметрами. Наприклад, зростання затримки в мережі часто супроводжується підвищенням навантаження на процесор і зміною рівня сигналу, що необхідно враховувати в діагностичних моделях.

Найбільш поширеними параметрами для діагностики є затримка передачі даних, пакетні втрати, частота помилок та енергоспоживання. Затримка є важливим індикатором якості зв'язку і може вказувати на перевантаження мережі або на фізичні проблеми з каналами зв'язку. Втрати пакетів сигналізують про можливі перешкоди в мережі або перевантаження, що впливає на стабільність передачі даних.

Інші важливі параметри включають *CPU load*, *memory usage*, *disk I/O* — вони допомагають моніторити ефективність обробки даних на пристроях і можуть бути ознаками внутрішніх збоїв або перевантажень. Наприклад, різке підвищення навантаження на процесор може бути сигналом про некоректну роботу програмного забезпечення або про спроби несанкціонованого доступу до системи.

Для виявлення таких аномалій у реальному часі необхідно використовувати підходи, що здатні не тільки аналізувати окремі параметри, а й визначати їхнє взаємне відхилення від звичайного поведінкового шаблону системи. Виявлення таких аномалій за допомогою традиційних методів (наприклад, порогового аналізу) часто дає високий рівень помилкових спрацювань, що не підходить для складних систем з великою кількістю вимірюваних параметрів.

Для побудови надійної моделі діагностики важливо провести попередню обробку та нормалізацію телеметричних параметрів, що дозволить знизити вплив шуму та забезпечить більш точні результати моделювання. Врахування різних типів даних, їхнього масштабу і динаміки є основою для формування стабільної та адаптивної моделі для діагностики технічних систем.

1.2.3. Проблеми використання *IoT*-даних у задачах діагностики

Збір та обробка телеметрії є основними етапами у створенні системи діагностики технічних об'єктів, проте дані, що надходять з *IoT*-пристроїв, часто мають низку характеристик, які ускладнюють їх використання для точного діагностування. Особливо це стосується таких систем, де обсяг даних великий, а зміна станів відбувається повільно, або навіть непомітно на перших етапах. Існує кілька суттєвих проблем, які треба враховувати при розробці моделей для

виявлення аномалій у таких даних.

Таблиця 1.5

Проблеми використання *IoT*-даних у задачах діагностики

Проблема	Опис	Вплив на діагностику
Шум у даних	Фонові шуми, неточності вимірювань, спотворення сигналів	Знижує точність діагностики, потребує фільтрації
Неповнота даних	Відсутність частини даних через обмеження сенсорів або мережі	Ускладнює побудову моделі, веде до пропуску важливих ознак
Асинхронність	Різна частота вимірювань даних від різних пристроїв	Ускладнює синхронізацію даних і виявлення аномалій
Нестабільність	Коливання параметрів через зміни умов експлуатації	Погіршує інтерпретацію даних і веде до хибних спрацювань
Масштабованість	Велика кількість пристроїв та даних	Вимагає високопродуктивних алгоритмів для обробки великих обсягів даних

Перша проблема — це шум у даних, який є неминучим у більшості *IoT*-систем. Наявність фонових шумів, неточностей вимірювань, спотворень сигналів може суттєво спотворити показники телеметрії. Шум може виникати як внаслідок апаратних несправностей (наприклад, проблеми з сенсорами або зовнішніми впливами), так і через програмні збої, які не завжди можна відразу виявити. Це робить дані менш точними і може привести до хибних спрацювань класичних порогових методів діагностики.

Друга проблема пов'язана з неповнотою даних. *IoT*-пристрої можуть передавати лише частину даних через обмеження в пропускну здатності каналу зв'язку або через енергетичні обмеження самих пристроїв. Також можливі випадки, коли деякі параметри не вимірюються через відсутність відповідних сенсорів або

через збої в роботі пристроїв. Неповнота даних створює додаткові труднощі при побудові діагностичних моделей, оскільки інтерпретація неповних або часткових даних може бути суперечливою.

Асинхронність вимірювань є ще однією важливою проблемою. Оскільки різні сенсори і пристрої можуть вимірювати параметри з різною частотою, дані, що надходять до системи, не завжди синхронізуються. Це може призвести до ситуацій, коли дані для окремих параметрів збігаються лише частково, що ускладнює точне моделювання поведінки системи та виявлення відхилень.

Ще однією проблемою є нестабільність даних, коли параметри можуть змінюватися не тільки внаслідок деградації або відмов, але й через зміни зовнішніх умов.

Наприклад, в умовах змінного навантаження на пристрої або змін у зовнішньому середовищі (температура, вологість) можуть виникати коливання параметрів, які не є аномальними, але можуть бути інтерпретовані як відхилення. У таких випадках традиційні підходи до діагностики можуть не спрацювати, оскільки не враховують контекст зміни параметрів.

Задача масштабування є ще однією суттєвою проблемою. В *IoT*-системах можуть бути тисячі пристроїв, які передають телеметричні дані. Обробка такої великої кількості параметрів вимагає значних обчислювальних ресурсів та ефективних методів їх обробки. Це вимагає використання спеціалізованих алгоритмів і моделей, здатних ефективно працювати з великими обсягами даних, не втрачаючи при цьому точності діагностики.

Для вирішення цих проблем необхідно використовувати сучасні методи машинного навчання та глибокого навчання, які дозволяють автоматично коригувати вплив шумів, обробляти неповні та асинхронні дані, а також адаптуватися до змінюваних умов експлуатації (табл. 1.6). Зокрема, моделі *LSTM Autoencoder* здатні враховувати часову динаміку та виявляти відхилення навіть в умовах великої варіативності та неповноти даних.

Стратегії вирішення проблем з даними в *IoT*-системах

Проблема	Стратегія вирішення	Опис стратегії
Шум у даних	Фільтрація даних	Використання фільтрів для зменшення впливу шуму (наприклад, фільтри Калмана)
Неповнота даних	Інтерполяція та заповнення пропусків	Використання статистичних методів або методів машинного навчання для заповнення пропусків
Асинхронність	Синхронізація даних	Використання методів часової інтерполяції або агрегації даних для синхронізації вимірювань
Нестабільність	Адаптивне порогоування	Використання динамічних порогів, що адаптуються до зміни зовнішніх умов
Масштабованість	Розподілені обчислення	Використання розподілених обчислень та алгоритмів для обробки великих обсягів даних

1.3. Ключові параметри продуктивності в *IoT*-мережах та їх вплив на діагностування

Функціонування *IoT*-систем визначається сукупністю мережевих, енергетичних та експлуатаційних характеристик, які безпосередньо впливають на якість збору телеметричних даних і можливості їх подальшого аналізу. На відміну від традиційних інформаційних систем, *IoT*-інфраструктури працюють у розподіленому середовищі з великою кількістю автономних пристроїв, що формує специфічні вимоги до пропускної здатності мереж, затримок передачі даних, енергоспоживання та надійності.

Пропускна здатність мережі визначає потенціал *IoT*-системи щодо одночасної роботи значної кількості пристроїв і передачі великих обсягів телеметричної інформації. Для технічних систем, що генерують високочастотні або

багатовимірні дані, недостатня пропускну здатність може призводити до втрати частини вимірювань або до агрегації даних, що знижує точність діагностування. Тому під час проектування *IoT*-систем важливо враховувати баланс між обсягом переданих даних і можливостями мережевої інфраструктури [7].

Не менш важливим параметром є затримка передачі даних, яка визначає оперативність реакції системи на зміну технічного стану об'єкта. У задачах діагностування це впливає на швидкість виявлення аномалій і можливість своєчасного прийняття керувальних рішень. Надмірні затримки можуть знецінювати результати аналізу, особливо в системах, де критичним є виявлення швидкоплинних або аварійних процесів.

Особливістю більшості *IoT*-пристроїв є їх автономність і обмежені енергетичні ресурси. Енергоефективність стає визначальним фактором тривалості експлуатації сенсорних вузлів і стабільності збору даних. Частота вимірювань, інтенсивність передачі телеметрії та обчислювальна складність локальної обробки даних безпосередньо впливають на споживання енергії. У контексті діагностування це створює необхідність оптимізації алгоритмів аналізу та обґрунтованого вибору режимів роботи пристроїв [8].

Надійність *IoT*-систем характеризується здатністю підтримувати коректну роботу в умовах нестабільного зв'язку, апаратних збоїв і зовнішніх впливів. Для діагностичних застосувань важливо забезпечити цілісність і повноту телеметричних даних, оскільки втрата або спотворення інформації може призвести до хибних рішень. Тому сучасні *IoT*-архітектури передбачають використання механізмів буферизації, повторної передачі та автоматичного відновлення з'єднання. Тісно пов'язаною з надійністю є доступність системи, яка визначає частку часу, протягом якого *IoT*-інфраструктура залишається працездатною. Для технічних систем, що перебувають під постійним моніторингом, висока доступність є критичною умовою коректного діагностування. Досягнення цього показника забезпечується за рахунок резервування компонентів, використання хмарних і крайових обчислювальних підходів, а також автоматизованого управління ресурсами [10].

Окрему роль відіграє масштабованість *IoT*-систем, яка визначає здатність

інфраструктури розширюватися без втрати якості обслуговування. Зі збільшенням кількості сенсорів і обсягу даних зростають вимоги до обробки, зберігання та аналізу інформації. У таких умовах традиційні централізовані підходи стають малоефективними, що обґрунтовує застосування асинхронних протоколів, брокерів повідомлень і розподілених обчислювальних моделей [9].

1.3.1. Методи машинного навчання в задачах діагностування

Методи машинного навчання почали активно застосовуватися в задачах технічної діагностики у зв'язку зі зростанням обсягів експлуатаційних даних та ускладненням структури сучасних технічних систем. На відміну від класичних підходів, які базуються на заздалегідь визначених правилах або математичних моделях, методи машинного навчання здатні автоматично виявляти закономірності у даних та адаптуватися до змін умов роботи системи.

У задачах діагностування машинне навчання найчастіше використовується для класифікації станів системи, кластеризації режимів роботи або прогнозування параметрів, що характеризують її технічний стан. При цьому дані телеметрії розглядаються як вхідний простір ознак, у якому модель навчається розрізняти нормальні та аномальні режими функціонування. Такий підхід дозволяє зменшити залежність від експертних налаштувань і підвищити гнучкість діагностичних рішень.

Методи класифікації застосовуються у випадках, коли відомий набір можливих станів системи, наприклад «нормальний режим», «деградація» та «відмова». Для цього використовуються алгоритми, такі як наївний баєсівський класифікатор, метод опорних векторів або дерева рішень. Однак ефективність класифікаційних підходів значною мірою залежить від наявності маркованих даних, що в реальних умовах експлуатації технічних систем є суттєвим обмеженням (табл. 1.7). Кластеризаційні методи, зокрема *k-means* або ієрархічна кластеризація, дозволяють групувати телеметричні дані за схожістю без попереднього маркування.

Методи машинного навчання у задачах діагностики

Метод	Тип задачі	Переваги	Обмеження
Класифікація	Розпізнавання станів	Чітка інтерпретація	Потребує маркованих даних
Кластеризація	Виявлення режимів	Не потребує розмітки	Не враховує часову динаміку
Регресія	Прогноз параметрів	Простота реалізації	Низька точність при нестационарності

У задачах діагностики це може використовуватися для виявлення нетипових режимів роботи або сегментації даних за станами системи. Проте такі методи зазвичай працюють зі статичними векторами ознак і погано враховують часову динаміку процесів.

Регресійні методи застосовуються для прогнозування значень параметрів системи або інтегральних показників її стану. Наприклад, на основі історичних даних можна передбачати очікувану поведінку системи і порівнювати її з фактичними вимірюваннями. Відхилення між прогнозом і реальними даними можуть розглядатися як індикатор аномалії. Однак точність таких методів суттєво знижується у випадку нестационарних процесів і складних багатовимірних залежностей.

Загальним обмеженням класичних методів машинного навчання є необхідність ручного формування ознак та їх агрегації. У випадку *IoT*-систем телеметрія має вигляд багатовимірних часових рядів, і спрощення цих даних до статичних векторів часто призводить до втрати інформації про поведінкові патерни системи. Це ускладнює виявлення повільних деградацій та прихованих аномалій.

Більшість традиційних алгоритмів машинного навчання погано масштабується при збільшенні кількості параметрів і довжини часових рядів. Зі зростанням складності системи зростає і ризик перенавчання, що знижує узагальнювальну здатність моделей і їх практичну цінність для задач діагностики в реальному часі.

1.3.2. Методи глибинного навчання для аналізу часових рядів

Застосування методів глибинного навчання в задачах аналізу часових рядів стало відповіддю на обмеження традиційних підходів машинного навчання при роботі зі складними, нестаціонарними та багатовимірними даними. На відміну від класичних алгоритмів, моделі глибинного навчання здатні автоматично формувати ієрархічні представлення ознак без необхідності ручного проектування характеристик, що є особливо важливим у задачах технічної діагностики на основі *IoT*-телеметрії.

Одним із перших підходів глибинного навчання для аналізу часових рядів стали згорткові нейронні мережі, адаптовані для роботи з одновимірними послідовностями. У таких моделях згорткові фільтри дозволяють виявляти локальні часові шаблони, наприклад короткочасні піки, коливання або повторювані структури в телеметричних сигналах. Проте *CNN*-моделі обмежені у здатності враховувати довготривалі залежності, що є критичним для виявлення повільних деградаційних процесів.

Рекурентні нейронні мережі були розроблені для подолання цього обмеження шляхом введення внутрішнього стану, який зберігає інформацію про попередні елементи послідовності. Класичні *RNN* здатні моделювати часову залежність між вимірюваннями, однак на практиці вони страждають від проблеми зникання та вибуху градієнтів, що ускладнює навчання на довгих послідовностях даних.

Для вирішення цих проблем були запропоновані архітектури з керованою пам'яттю, зокрема *Long Short-Term Memory (LSTM)* та *Gated Recurrent Unit (GRU)*. Ці моделі використовують спеціальні механізми керування потоком інформації, що дозволяє зберігати значущі залежності протягом тривалих часових інтервалів. Завдяки цьому *LSTM*- і *GRU*-мережі стали стандартом де-факто для аналізу часових рядів у задачах прогнозування, виявлення аномалій і діагностики технічних систем.

У контексті *IoT*-діагностики важливою властивістю рекурентних моделей є їх здатність працювати з багатовимірними часовими рядами, де кожен крок часу описується вектором телеметричних параметрів. Це дозволяє враховувати

взаємозв'язки між параметрами, а також їх спільну динаміку, що є критичним для виявлення складних деградаційних сценаріїв, які не проявляються у зміні одного окремого показника (табл. 1.8).

Таблиця 1.8

Основні моделі глибинного навчання для часових рядів

Модель	Призначення	Переваги	Обмеження
<i>CNN (1D)</i>	Локальні шаблони	Швидкість, простота	Погано враховує довгі залежності
<i>RNN</i>	Часова послідовність	Аналіз динаміки	Проблема градієнтів
<i>LSTM</i>	Довготривалі залежності	Стійкість до деградацій	Вища обчислювальна складність
<i>GRU</i>	Спрощена <i>LSTM</i>	Менше параметрів	Менша гнучкість

Окремий клас моделей глибинного навчання для часових рядів становлять гібридні архітектури, що поєднують згорткові та рекурентні шари. У таких моделях *CNN*-компоненти відповідають за виділення локальних шаблонів, а рекурентні шари — за аналіз їх часової еволюції. Хоча ці підходи демонструють високу точність, їх складність та обчислювальні вимоги часто обмежують застосування у системах з обмеженими ресурсами або в режимі реального часу.

Ще однією важливою характеристикою моделей глибинного навчання є їх здатність працювати в умовах відсутності повної розмітки даних. Для технічних систем це має принципове значення, оскільки приклади відмов є рідкісними, а більшість даних відповідає нормальним режимам роботи. Саме ця особливість створює передумови для використання моделей, орієнтованих на навчання без учителя або з частковим учителем.

1.3.3. Автоенкодерні моделі для виявлення аномалій

Автоенкодерні нейронні мережі є класом моделей глибинного навчання, призначених для навчання компактного внутрішнього представлення даних шляхом відтворення вхідного сигналу на виході мережі. Основна ідея автоенкодера

полягає у стисканні вхідних даних у латентний простір меншої розмірності з подальшою реконструкцією початкового сигналу. Якість цієї реконструкції безпосередньо відображає здатність моделі описувати характерну структуру вхідних даних.

У задачах технічної діагностики автоенкодері набули широкого застосування завдяки можливості навчання без учителя. Модель навчається виключно на даних нормального режиму роботи системи, формуючи узагальнену модель її типової поведінки. У такому підході аномалія визначається не за наявністю конкретного класу відмови, а за ступенем відхилення поточних даних від відтвореної моделлю нормальної динаміки.

Ключовим діагностичним показником у автоенкодерних моделях є помилка реконструкції, яка визначається як різниця між вхідним сигналом та його реконструкцією на виході мережі. Для нормальних режимів роботи ця помилка є відносно малою, оскільки модель добре відтворює знайомі шаблони. У випадку деградації або аномального стану помилка реконструкції зростає, що дозволяє використовувати її як кількісний критерій для діагностики.

На відміну від класичних порогових методів, автоенкодер не потребує явного визначення допустимих меж для кожного параметра. Діагностичне рішення формується на основі комплексної поведінки багатовимірного вектора ознак, що є особливо важливим для *IoT*-систем, де аномалії часто проявляються через зміну взаємозв'язків між параметрами, а не через вихід одного показника за межі норми.

Існують різні архітектури автоенкодерів, серед яких класичні повнозв'язні, згорткові та рекурентні моделі. Для аналізу часових рядів найбільш придатними є рекурентні автоенкодері, у яких як кодер, так і декодер реалізовані на основі рекурентних нейронних мереж. Такі архітектури дозволяють враховувати часову структуру даних і відтворювати не лише окремі значення параметрів, а й їх динаміку у часі.

Особливе місце серед рекурентних автоенкодерів займають моделі на основі *LSTM*, які здатні ефективно працювати з довгими часовими послідовностями та зберігати інформацію про значущі події у віддаленому минулому. Це робить *LSTM Autoencoder* оптимальним вибором для діагностування повільних деградаційних

процесів, характерних для тривало експлуатованих технічних систем.

Для практичного застосування автоенкодерних моделей важливим є вибір порогу аномальності, який визначає межу між нормальним та аномальним станом. Поріг може встановлюватися на основі статистичних характеристик помилки реконструкції на навчальній вибірці або шляхом аналізу емпіричних розподілів.

1.3.4. Переваги автоенкодерного підходу у порівнянні з пороговими та статистичними методами

У задачах технічної діагностики традиційні порогові та статистичні методи тривалий час залишалися основним інструментом виявлення відмов і аномалій. Їх застосування ґрунтується на припущенні про наявність чітко визначених меж нормальної роботи системи або стабільних статистичних характеристик параметрів. Проте для сучасних технічних систем, зокрема *IoT*-орієнтованих, ці припущення часто не виконуються, що знижує ефективність класичних підходів.

Порогові методи передбачають встановлення фіксованих граничних значень для кожного параметра, перевищення яких інтерпретується як ознака несправності. Такий підхід є простим у реалізації, але не враховує багатовимірний характер системи та взаємозалежність параметрів. У випадку повільних деградацій або складних аномалій, які проявляються через зміну кореляцій між показниками, порогові методи часто не здатні виявити проблему на ранніх етапах.

Статистичні методи, зокрема аналіз середніх значень, дисперсій або контрольні карти, дозволяють частково враховувати змінність параметрів у часі. Проте вони зазвичай базуються на припущенні стаціонарності процесів і стабільності розподілів. У реальних умовах експлуатації *IoT*-систем ці припущення порушуються через змінні режими навантаження, вплив зовнішніх факторів і еволюцію самої системи з часом.

Автоенкодерний підхід принципово відрізняється від класичних методів тим, що він не потребує явного задання правил або порогів для кожного параметра. Натомість модель навчається відтворювати характерну структуру даних нормального режиму роботи системи, формуючи узагальнену поведінкову модель.

Діагностичне рішення базується на оцінці відхилення поточної поведінки від цієї моделі, що дозволяє враховувати комплексний вплив усіх параметрів.

Важливою перевагою автоенкодерів є їх здатність працювати з багатовимірними даними без необхідності зменшення розмірності шляхом агрегації або ручного відбору ознак. У задачах *IoT*-діагностики це дозволяє зберігати повну інформацію про часову динаміку параметрів і взаємозв'язки між ними. Таким чином, автоенкодер здатний виявляти аномалії, які не проявляються у вигляді різких змін окремих показників.

Ще однією суттєвою перевагою автоенкодерного підходу є можливість навчання без учителя, що є критично важливим для практичних систем діагностики. У більшості реальних випадків дані про відмови або аварійні стани є обмеженими або взагалі відсутні. Автоенкодери дозволяють формувати діагностичну модель на основі даних нормальної експлуатації, що значно спрощує впровадження системи.

З точки зору адаптивності автоенкодерні моделі перевершують класичні методи, оскільки можуть перенавчатися або донавчатися з урахуванням нових даних. Це дозволяє враховувати еволюцію системи, зміни умов експлуатації та появу нових режимів роботи без повної перебудови діагностичного алгоритму. Така властивість є особливо цінною для довготривалих систем моніторингу.

1.3.5. Особливості застосування *LSTM Autoencoder* до багатовимірних *IoT*-часових рядів

Застосування *LSTM Autoencoder* у задачах діагностування технічних систем обумовлене специфікою телеметричних даних, що формуються в *IoT*-середовищі. Такі дані мають вигляд багатовимірних часових рядів, у яких кожен момент часу описується вектором параметрів, що характеризують різні аспекти функціонування системи. Аналіз окремих параметрів у відриві від інших не дозволяє повною мірою відобразити реальний стан системи, тому використання моделей, здатних обробляти векторні послідовності, є принципово важливим.

LSTM Autoencoder поєднує переваги рекурентних нейронних мереж і

автоенкодерного підходу, забезпечуючи ефективне моделювання часової динаміки багатовимірних даних. Кодер *LSTM* поступово агрегує інформацію про послідовність телеметричних вимірювань, формуючи стислий латентний опис нормального режиму роботи системи. Декодер, у свою чергу, відтворює початкову послідовність на основі цього латентного представлення, що дозволяє оцінити якість реконструкції.

Особливістю *IoT*-часових рядів є їх неоднорідність за масштабом і характером змін параметрів. Наприклад, затримка мережевої передачі може змінюватися в межах десятків мілісекунд, тоді як енергоспоживання або навантаження процесора мають зовсім інший числовий діапазон. Для коректної роботи *LSTM Autoencoder* необхідним є попереднє масштабування та нормалізація даних, що дозволяє уникнути домінування окремих параметрів у процесі навчання моделі.

Ще однією важливою особливістю є вибір довжини часових вікон, на основі яких формується вхідна послідовність для автоенкодера. Надто короткі вікна не дозволяють зафіксувати довготривалі залежності та повільні деградаційні процеси, тоді як надто довгі вікна збільшують обчислювальну складність і можуть призводити до втрати локальних особливостей. Таким чином, вибір параметрів часової сегментації є важливою частиною проектування діагностичної системи.

У багатовимірних *IoT*-даних аномалії часто мають комплексний характер і проявляються у зміні взаємозв'язків між параметрами. *LSTM Autoencoder* здатний виявляти такі зміни за рахунок аналізу спільної динаміки всіх ознак у послідовності. Наприклад, незначне зростання затримки у поєднанні з підвищенням навантаження на процесор і зниженням рівня сигналу може бути ранньою ознакою деградації, яка не фіксується при аналізі кожного параметра окремо.

Практичне застосування *LSTM Autoencoder* передбачає розрахунок помилки реконструкції для кожного часового вікна або моменту часу. Отримане значення помилки використовується як інтегральний діагностичний показник, що відображає ступінь відхилення поточної поведінки системи від її нормального стану. Такий підхід дозволяє не лише фіксувати факт аномалії, а й оцінювати її

інтенсивність у динаміці.

Важливою перевагою *LSTM Autoencoder* є можливість роботи з даними, що містять пропуски або нерівномірні інтервали між вимірюваннями. За умови коректної попередньої обробки даних модель зберігає здатність формувати стабільне латентне представлення навіть у складних умовах збору телеметрії. Це робить підхід придатним для реальних *IoT*-систем, де ідеальна якість даних практично недосяжна.

1.4. Аналіз сучасних технологій та готових рішень дистанційного управління на прикладі систем освітлення

Системи дистанційного управління освітленням є однією з найбільш поширених і технологічно зрілих підсистем у складі сучасних автоматизованих і *IoT*-орієнтованих середовищ. Вони широко застосовуються як у побутових інсталяціях, так і в комерційних, промислових та інфраструктурних об'єктах, де освітлення виступає не лише засобом комфорту, а й елементом енергоефективності, безпеки та керування процесами. Наявність великої кількості готових програмно-апаратних рішень зумовлює необхідність їх системного аналізу з метою визначення сильних і слабких сторін, а також обґрунтування вибору архітектурних підходів для подальшої розробки власної системи.

Сучасні рішення для керування освітленням базуються на різних комунікаційних протоколах і апаратних платформах, що визначає їх функціональні можливості та сферу застосування. Бездротові технології дозволяють швидко розгортати системи без складних монтажних робіт, тоді як дротові стандарти орієнтовані на високу надійність і масштабованість. Програмна складова таких систем, як правило, реалізує функції централізованого керування, автоматизації сценаріїв, моніторингу стану пристроїв і взаємодії з користувачем через графічні або програмні інтерфейси.

Значну частку ринку займають платформи з відкритим програмним кодом, орієнтовані на локальне розгортання та глибоку інтеграцію різномірних пристроїв. Такі системи надають широкі можливості для налаштування логіки керування

освітленням, створення сценаріїв на основі подій і сенсорних даних, а також інтеграції з іншими підсистемами “розумного будинку”. Їхньою перевагою є незалежність від хмарної інфраструктури та підвищений рівень контролю над даними, проте ефективне використання часто потребує технічної підготовки та значних зусиль на етапі конфігурації.

Поряд із цим активно розвиваються комерційні хмарні рішення, орієнтовані на масового користувача. Такі системи забезпечують простоту підключення та інтуїтивно зрозумілий інтерфейс керування освітленням через мобільні застосунки. Вони підтримують типові сценарії автоматизації, інтеграцію з голосовими асистентами та віддалений доступ через Інтернет. Водночас залежність від зовнішніх серверів, обмежена можливість кастомізації та потенційні ризики з точки зору безпеки й доступності обмежують використання цих рішень у критично важливих або промислових системах.

Окрему категорію становлять професійні інсталяційні системи автоматизації будівель, які використовуються у великих комерційних і житлових об’єктах. Вони відзначаються високою надійністю, стандартизованими протоколами та підтримкою обладнання від численних виробників. Такі рішення забезпечують точне керування освітленням, інтеграцію з системами клімат-контролю, безпеки та енергоменеджменту, проте характеризуються високою вартістю впровадження, складністю проектування та необхідністю спеціалізованого програмного забезпечення для налаштування.

Порівняльний аналіз наявних рішень свідчить, що жоден із підходів не є універсальним. Локальні *open-source* платформи забезпечують гнучкість і автономність, але ускладнюють масштабування й супровід. Хмарні сервіси спрощують використання, проте обмежують контроль і надійність. Професійні стандартизовані системи гарантують стабільність і довговічність, однак є малодоступними з економічної точки зору та надмірними для задач середньої складності.

У контексті даної кваліфікаційної роботи доцільним є проектування власної комп’ютерної системи дистанційного управління освітленням, яка поєднує переваги існуючих підходів і усуває їх ключові обмеження. Така система повинна

базуватися на модульній архітектурі, підтримувати локальну обробку подій, асинхронну взаємодію з пристроями та масштабування до великої кількості *IoT*-вузлів. Особливу роль у цьому підході відіграє можливість подальшого розширення функціональності системи за рахунок інтелектуальних методів аналізу та діагностування, що є предметом подальшого дослідження у цій роботі.

Серед сучасних програмних платформ для дистанційного керування освітленням значне поширення отримали рішення з відкритим програмним кодом, орієнтовані на локальне розгортання та інтеграцію широкого спектра *IoT*-пристроїв. Одним із найвідоміших представників цього класу є *Home Assistant* (рис. 1.1). Дана система забезпечує централізоване управління освітленням і іншими підсистемами за допомогою вебінтерфейсу та сценарної логіки автоматизації. Конфігурація системи здійснюється через декларативні налаштування, що дозволяє гнучко описувати події та реакції без жорсткої прив'язки до конкретних виробників обладнання. Важливою особливістю є можливість повністю автономної роботи без постійного звернення до хмарних сервісів, що підвищує рівень контролю над даними та стабільність функціонування.

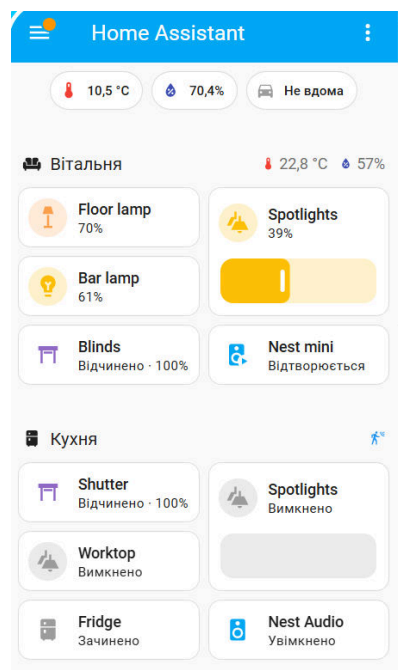


Рис. 1.1. Приклад управління розумним будинком в системі *Home Assistant*

Іншим прикладом відкритої платформи є *OpenHAB*, яка також орієнтована на локальне використання та масштабні інсталяції. Архітектура цієї системи

побудована на концепції модульних прив'язок, що дозволяє підключати різноманітні пристрої через уніфікований механізм інтеграції. *OpenHAB* добре підходить для складних об'єктів із великою кількістю компонентів, де необхідна детальна кастомізація логіки керування освітленням, безпекою та кліматичними системами.

Окрему нішу займають професійні стандартизовані системи автоматизації будівель, серед яких провідне місце посідає *KNX* (рис. 1.2). Для конфігурації таких систем використовується спеціалізоване інженерне програмне забезпечення *ETS*, яке дозволяє формалізовано описувати взаємодію між пристроями освітлення та іншими підсистемами. Перевагою цього підходу є висока надійність, довготривала підтримка стандарту та сумісність обладнання від багатьох виробників, що робить *KNX* поширеним вибором для комерційних і житлових об'єктів великого масштабу.

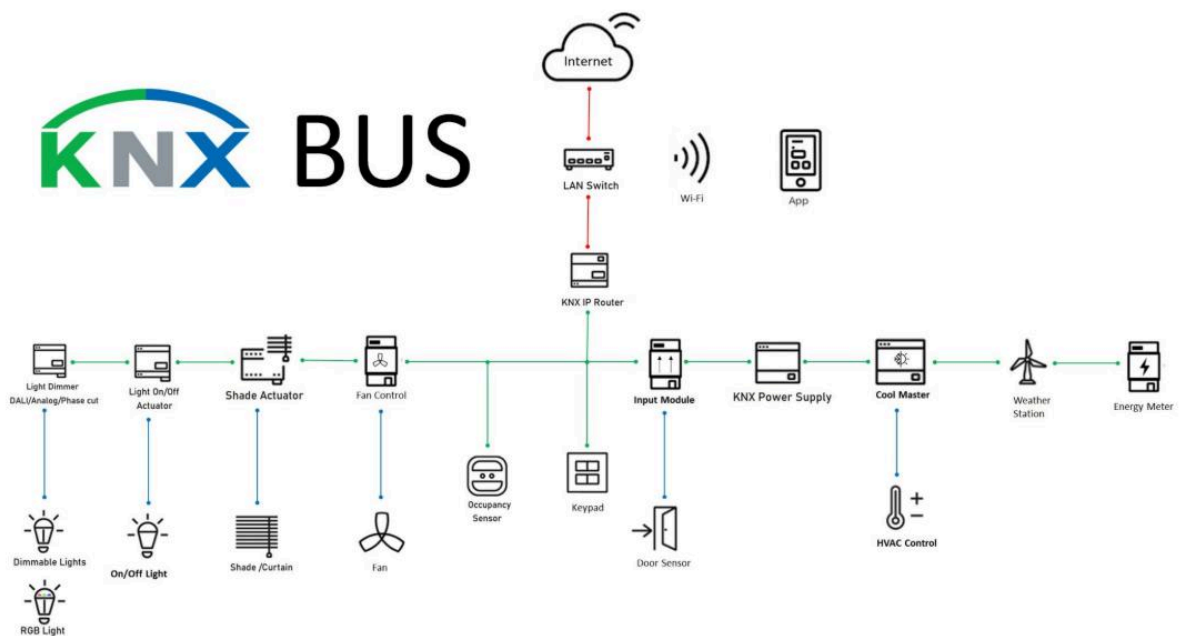


Рис. 1.2. Підхід до управління розумним будинком *KNX*

Серед комерційних рішень для керування освітленням значну популярність отримала екосистема *Philips Hue*, у якій ключову роль відіграє апаратно-програмний шлюз *Hue Bridge*. Він забезпечує зв'язок між світильниками, що працюють за бездротовим протоколом, і користувацькими застосунками або сервісами голосового керування (рис. 1.3). Такий підхід орієнтований на простоту використання та швидке розгортання, дозволяючи користувачам створювати сцени освітлення та автоматизовані сценарії без складного налаштування.

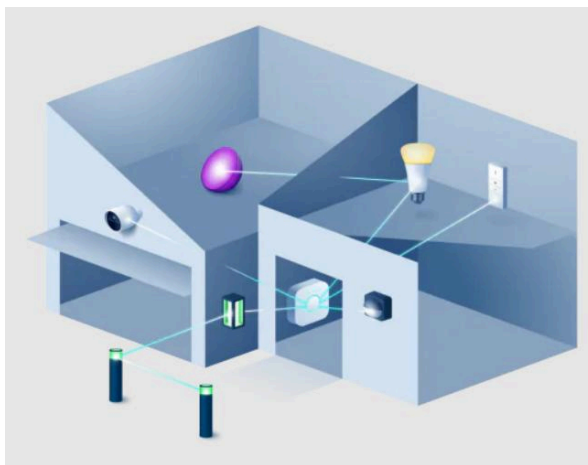


Рис. 1.3. Приклад управління розумним будинком *Philips Hue*

Ще одним прикладом хмарно-орієнтованих платформ є *Tuya*, яка надає інфраструктуру та програмні інструменти для масового підключення *IoT*-пристроїв різних виробників. Керування освітленням у цій екосистемі здійснюється через мобільні застосунки, що забезпечують уніфікований інтерфейс для широкого асортименту пристроїв. Основною перевагою такого підходу є мінімальні вимоги до налаштування, однак функціонування системи безпосередньо залежить від доступності хмарних сервісів.

Проміжне положення між локальними та хмарними рішеннями займає платформа *Shelly*, яка поєднує *Wi-Fi*-керовані реле й димери з можливістю інтеграції через відкриті програмні інтерфейси. Підтримка як локальних протоколів взаємодії, так і хмарних сервісів забезпечує гнучкість у виборі сценаріїв розгортання та дозволяє використовувати систему як у простих, так і в більш складних автоматизованих інсталяціях.

Наявні програмні рішення для керування освітленням можна умовно поділити на три класи. Локальні *open-source* платформи орієнтовані на автономність і глибоку інтеграцію, але потребують значних зусиль під час налаштування. Хмарні комерційні системи забезпечують простоту використання, проте обмежують контроль і залежать від зовнішньої інфраструктури. Професійні інсталяційні стандарти гарантують високу надійність і масштабованість, але характеризуються складністю впровадження та високою вартістю. Така різноманітність підходів підтверджує актуальність розробки власної комп'ютерної

системи, здатної поєднати гнучкість, автономність і можливість подальшого інтелектуального розширення.

1.5. Висновки до розділу

У першому розділі здійснено системний аналіз теоретичних основ діагностування технічних систем в умовах сучасних *IoT*-орієнтованих архітектур. Показано, що технічні системи в процесі експлуатації проходять низку функціональних станів, включаючи нормальний режим, деградацію, передвідмовний стан та відмову, причому межі між цими станами часто є розмитими та не можуть бути однозначно визначені за допомогою простих порогових критеріїв.

Розглянуто основні види відмов і деградацій, характерні для сучасних технічних систем, зокрема фізичні, інформаційні, мережеві та енергетичні. Встановлено, що значна частина деградаційних процесів має поступовий характер і проявляється у зміні часової динаміки та взаємозв'язків між параметрами, а не у різких відхиленнях окремих показників. Це ускладнює їх виявлення традиційними методами технічної діагностики.

Проаналізовано структуру *IoT*-систем та особливості формування телеметричних даних, які використовуються як первинні діагностичні ознаки. Визначено, що *IoT*-телеметрія має характер багатовимірних часових рядів з притаманними їй властивостями асинхронності, нестаціонарності, наявності шумів і пропусків. Такі властивості зумовлюють необхідність використання методів аналізу, здатних працювати з комплексною часовою динамікою даних.

У розділі показано обмеження класичних порогових і статистичних методів діагностики, а також традиційних підходів машинного навчання, які потребують маркованих даних або ручного формування ознак. Обґрунтовано, що ці методи є недостатньо ефективними для діагностування складних *IoT*-орієнтованих систем, особливо в умовах змінних режимів експлуатації та відсутності повної інформації про аварійні стани.

Особливу увагу приділено аналізу методів глибинного навчання для обробки

часових рядів, зокрема рекурентних нейронних мереж і їх модифікацій. Встановлено, що моделі класу *LSTM* є найбільш придатними для аналізу багатовимірної телеметрії завдяки здатності враховувати довготривалі часові залежності та складні нелінійні взаємозв'язки між параметрами.

На основі проведеного аналізу обґрунтовано доцільність використання автоенкодерного підходу для виявлення аномалій у задачах технічної діагностики. Показано, що автоенкодерні моделі дозволяють формувати узагальнену модель нормальної поведінки системи та здійснювати діагностику без необхідності попереднього маркування відмов. Особливо ефективним для аналізу *IoT*-часових рядів є використання *LSTM Autoencoder*, який поєднує аналіз часової динаміки та принцип реконструкції даних.

РОЗДІЛ 2

ІНТЕГРАЦІЯ ПРОГРАМНОГО МОДУЛЯ ДІГНОСТУВАННЯ РОБОТИ *IoT* ПРИБОРІВ В СИСТЕМУ ОСВІТЛЕННЯ

Розвиток *IoT*-орієнтованих систем освітлення супроводжується поступовим переходом від простих керованих пристроїв до складних розподілених технічних систем, у яких освітлювальні вузли виконують роль повноцінних учасників інформаційного обміну. Такі системи характеризуються постійною генерацією телеметричних даних, асинхронною взаємодією компонентів і динамічною зміною режимів роботи залежно від зовнішніх умов та користувацьких сценаріїв. За цих умов зростає значення не лише функціонального керування освітленням, а й забезпечення стабільності, надійності та прогнозованості роботи всієї інфраструктури.

Практика експлуатації *IoT*-систем освітлення показує, що значна частина відмов і збоїв має не миттєвий, а накопичувальний характер. Поступове погіршення якості зв'язку, деградація елементів живлення, нестабільна робота реле або контролерів, а також програмні помилки не завжди призводять до негайної втрати працездатності. Натомість система продовжує функціонувати у формально допустимих режимах, водночас демонструючи атипову або нехарактерну поведінку, яка залишається поза увагою класичних механізмів контролю.

Традиційні підходи до моніторингу роботи систем освітлення переважно зосереджені на контролі доступності пристроїв і перевірці окремих параметрів за наперед заданими порогоми. Подібні методи є простими в реалізації, проте не дозволяють аналізувати поведінку системи як цілісного динамічного об'єкта. У результаті діагностика зводиться до фіксації наслідків відмови, а не до виявлення її передумов, що суттєво обмежує можливості превентивного обслуговування та підвищення надійності.

<i>Кафедра ІКС</i>				<i>КАІ 25 02 90 000 ПЗ</i>			
<i>Виконав</i>	<i>Войцехівський Т.М.</i>			<i>Інтеграція програмного модуля діагностування роботи IoT пристроїв в систему освітлення</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	<i>41</i>	<i>83</i>
<i>Консульт.</i>					<i>M-126-24-1-IT</i>		
<i>Норм. контр.</i>	<i>Тупота С.В.</i>						
<i>Зав. Каф.</i>	<i>Нечипорук О.П.</i>						

Особливістю *IoT*-систем освітлення є тісний взаємозв'язок між великою кількістю параметрів, які формують складні багатовимірні часові ряди. Зміни навантаження, режимів освітлення, частоти комутацій, умов мережевої взаємодії та зовнішніх факторів відображаються у телеметричних даних у вигляді нелінійних залежностей, які важко формалізувати аналітичними методами. Це створює об'єктивну потребу у використанні підходів, здатних автоматично виявляти приховані закономірності та відхилення у поведінці системи.

Інтеграція програмного модуля діагностування в таку систему повинна здійснюватися з урахуванням її розподіленої архітектури та вимог до безперервної роботи. Діагностичний компонент не може втручатися у критичні контури керування освітленням або створювати додаткові точки відмови. Натомість він має функціонувати як незалежний аналітичний рівень, що отримує дані з існуючих каналів обміну та формує діагностичні оцінки на основі аналізу історії й поточної динаміки роботи пристроїв.

Застосування нейромережевих методів у задачах діагностування *IoT*-пристроїв освітлення дозволяє перейти від локального аналізу окремих сигналів до поведінкового аналізу системи в цілому. Моделі глибинного навчання, зокрема рекурентні автоенкодера, здатні формувати узагальнене уявлення про нормальні режими роботи освітлювальної інфраструктури на основі реальних експлуатаційних даних. Це створює можливість виявляти відхилення навіть у випадках, коли значення окремих параметрів залишаються в допустимих межах.

Важливою особливістю інтеграції діагностичного модуля є необхідність забезпечення його масштабованості та адаптивності. Системи освітлення можуть динамічно розширюватися за рахунок підключення нових пристроїв або зміни конфігурації існуючих, що потребує автоматичного пристосування механізмів діагностики до нових умов. У цьому контексті нейромережевий підхід дозволяє зменшити обсяг ручного налаштування та забезпечити узгодженість аналізу при зростанні системи.

Окрему увагу слід приділити питанням інтеграції діагностування з інформаційними потоками системи освітлення. Дані, що надходять від *IoT*-пристроїв, можуть мати різну частоту, точність і часову структуру, що ускладнює

їх пряме використання для аналітики. Тому інтеграція програмного модуля діагностування потребує попередньої синхронізації, нормалізації та агрегації телеметричних потоків з метою формування коректного вхідного простору для нейромережових моделей.

2.1. Загальна архітектура програмного комплексу діагностування

2.1.1. Загальна структура програмного комплексу діагностування

Програмний комплекс діагностування технічних систем призначений для забезпечення безперервного моніторингу стану об'єкта, аналізу телеметричних даних та автоматичного виявлення аномальних режимів роботи. Його структура формується з урахуванням розподіленого характеру *IoT*-систем, необхідності обробки поточкових даних у режимі, близькому до реального часу, а також інтеграції модуля глибокого навчання для інтелектуальної діагностики.

Архітектура програмного комплексу має багаторівневу структуру, яка включає рівень збору даних, рівень попередньої обробки, рівень інтелектуального аналізу та рівень представлення результатів. Такий підхід дозволяє розділити функціональну відповідальність між компонентами системи, спростити масштабування та забезпечити гнучкість при модернізації окремих модулів без повної перебудови комплексу.

На рівні збору даних здійснюється приймання телеметрії від *IoT*-пристроїв і сенсорних вузлів. Дані можуть надходити у вигляді потоків вимірювань з різною частотою та у різних форматах. Тому на цьому рівні реалізуються механізми агрегації та первинної валідації даних, що дозволяє відфільтрувати очевидно некоректні значення та забезпечити цілісність вхідної інформації.

Рівень попередньої обробки виконує підготовку телеметричних даних до подальшого аналізу. На цьому етапі здійснюється нормалізація параметрів, синхронізація часових рядів, заповнення пропусків і формування часових вікон фіксованої довжини. Коректна реалізація цього рівня є критично важливою для

стабільної роботи модуля глибинного навчання та забезпечення достовірності результатів діагностики (табл. 2.1).

Таблиця 2.1

Основні рівні програмного комплексу діагностування

Рівень	Функціональне призначення	Основні операції
Збір даних	Отримання телеметрії	Приймання, агрегація
Попередня обробка	Підготовка даних	Нормалізація, синхронізація
Інтелектуальний аналіз	Діагностика	<i>LSTM</i> <i>Autoencoder</i> , обчислення помилки
Представлення результатів	Взаємодія з користувачем	Візуалізація, сповіщення

Центральним елементом програмного комплексу є рівень інтелектуального аналізу, у якому реалізується модуль *LSTM Autoencoder*. Цей модуль здійснює аналіз багатовимірних часових рядів телеметрії та обчислює помилку реконструкції як інтегральний діагностичний показник. Результати аналізу передаються до модуля прийняття рішень, де формується висновок про поточний стан системи (табл. 2.2).

Таблиця 2.2

Основні компоненти програмного комплексу

Компонент	Опис	Роль у діагностиці
Модуль збору даних	Інтерфейс з <i>IoT</i> -пристроями	Формування телеметрії
Модуль обробки	Підготовка часових рядів	Забезпечення якості даних
Модуль <i>DL</i> -діагностики	<i>LSTM Autoencoder</i>	Виявлення аномалій
Інтерфейс користувача	Панелі моніторингу	Аналіз результатів

Рівень представлення результатів забезпечує взаємодію програмного комплексу з користувачем або зовнішніми інформаційними системами. На цьому рівні реалізуються інтерфейси візуалізації діагностичних показників, сповіщення про виявлені аномалії та засоби експорту результатів для подальшого аналізу.

Такий підхід дозволяє використовувати програмний комплекс як у автономному режимі, так і в складі більшої інформаційної системи.

Важливою особливістю структури програмного комплексу є можливість його масштабування та адаптації до різних типів технічних систем. Завдяки модульній архітектурі система може бути розширена шляхом додавання нових джерел телеметрії або модифікації модуля діагностики без порушення роботи інших компонентів. Це забезпечує універсальність та практичну придатність комплексу в різних сферах застосування.

2.1.2. Взаємодія компонентів програмного комплексу діагностування

Взаємодія компонентів програмного комплексу діагностування побудована за принципом послідовної обробки даних, починаючи з моменту їх надходження від *IoT*-пристроїв і завершуючи формуванням діагностичного висновку. Такий підхід забезпечує прозорість потоків інформації, спрощує контроль цілісності даних і дозволяє чітко визначити відповідальність кожного програмного модуля.

Початковим етапом взаємодії є передавання телеметричних даних від сенсорних вузлів до модуля збору даних. Залежно від архітектури *IoT*-системи це може здійснюватися через брокери повідомлень, *REST*-інтерфейси або спеціалізовані протоколи обміну. На цьому етапі дані розглядаються як потік вимірювань, що потребує мінімальної затримки та гарантії доставки для забезпечення актуальності діагностичної інформації.

Після приймання телеметрії модуль збору даних передає її до модуля попередньої обробки. Взаємодія між цими компонентами реалізується у вигляді асинхронної передачі структурованих даних, що дозволяє відокремити процес збору від обчислювально затратних операцій обробки. Такий підхід підвищує стійкість системи до пікових навантажень і втрати окремих пакетів даних.

Модуль попередньої обробки виконує перетворення сирих телеметричних даних у формат, придатний для подальшого аналізу. Результатом його роботи є сформовані багатовимірні часові вікна з нормалізованими параметрами та синхронізованими часовими мітками. Ці дані передаються до модуля

інтелектуального аналізу у вигляді послідовностей фіксованої довжини, що відповідають вимогам *LSTM Autoencoder*.

Взаємодія між модулем попередньої обробки та модулем глибинного навчання є ключовою для забезпечення коректної діагностики. Помилки на цьому етапі можуть призвести до спотворення латентного представлення або некоректної оцінки помилки реконструкції. Тому обмін даними між цими компонентами супроводжується контролем форматів, розмірностей і допустимих діапазонів значень.

Модуль інтелектуального аналізу на основі *LSTM Autoencoder* обробляє отримані часові вікна та обчислює значення помилки реконструкції. Результати аналізу передаються до модуля прийняття рішень, де відбувається порівняння отриманих значень із встановленими діагностичними порогами. На цьому етапі формується висновок про поточний стан системи та можливу наявність аномалії.

Модуль прийняття рішень взаємодіє з рівнем представлення результатів, передаючи інформацію про виявлені відхилення, їх інтенсивність та часову локалізацію. Це дозволяє відобразити діагностичну інформацію у вигляді графіків, індикаторів або повідомлень тривоги. Така взаємодія забезпечує своєчасне інформування оператора або зовнішніх систем управління.

Взаємодія компонентів програмного комплексу організована таким чином, щоб забезпечити мінімальні затримки, масштабованість і можливість незалежного розвитку окремих модулів. Чітко визначені інтерфейси обміну даними створюють основу для подальшого розширення системи та інтеграції додаткових діагностичних або аналітичних компонентів.

2.1.3. Вимоги до апаратного та програмного забезпечення програмного комплексу діагностування

Ефективне функціонування програмного комплексу діагностування технічних систем потребує відповідного апаратного та програмного забезпечення, здатного забезпечити стабільну обробку поточкових *IoT*-даних і виконання обчислювально складних алгоритмів глибинного навчання. Формування вимог

здійснюється з урахуванням необхідності роботи в режимі, близькому до реального часу, а також можливості масштабування системи при збільшенні кількості джерел телеметрії.

Апаратні вимоги визначаються, перш за все, обсягом і швидкістю надходження даних, а також складністю нейромережових моделей, що використовуються для діагностики. Для рівня збору та попередньої обробки телеметрії достатньо використання серверних або вбудованих обчислювальних платформ із помірними обчислювальними ресурсами. Водночас модуль глибокого навчання потребує значно вищої обчислювальної потужності, особливо на етапі навчання *LSTM Autoencoder*.

У випадку експлуатації програмного комплексу в розподіленому середовищі доцільним є використання окремих обчислювальних вузлів для нейромережового аналізу. Це дозволяє розвантажити модулі збору та обробки даних і підвищити загальну продуктивність системи. За наявності обмежень на апаратні ресурси допускається виконання лише етапу інференсу нейронної мережі на периферійних пристроях, тоді як навчання моделі виконується централізовано.

Програмні вимоги до комплексу визначаються необхідністю забезпечення сумісності між компонентами, стійкості до збоїв і простоти розгортання. Базове програмне середовище має підтримувати багатопоточну обробку даних, асинхронну взаємодію між модулями та інтеграцію з бібліотеками глибокого навчання. Операційна система повинна забезпечувати стабільну роботу мережових сервісів і ефективно управління ресурсами.

Для реалізації модуля глибокого навчання необхідна підтримка сучасних фреймворків, що забезпечують роботу з рекурентними нейронними мережами та автоматичне диференціювання. Програмне середовище повинно дозволити зберігати, завантажувати та оновлювати нейромережові моделі без зупинки всієї системи. Це є важливим для підтримки адаптивності діагностичного алгоритму в процесі експлуатації.

Важливою вимогою до програмного забезпечення є можливість журналювання даних і результатів діагностики. Збереження телеметрії, значень помилки реконструкції та діагностичних рішень дозволяє здійснювати подальший

аналіз ефективності системи, а також використовувати накопичені дані для донавчання нейромережевої моделі. Це сприяє підвищенню точності та надійності діагностики з часом.

З точки зору інформаційної безпеки програмний комплекс повинен забезпечувати захист каналів передавання даних, автентифікацію джерел телеметрії та контроль доступу до результатів діагностики. Оскільки система може працювати з критично важливими даними, дотримання базових вимог безпеки є обов'язковим для її практичного впровадження.

2.2. Розробка модуля попередньої обробки даних

2.2.1. Формування та підготовка *IoT*-даних для нейромережевого аналізу

Ефективність нейромережевої діагностики технічних систем значною мірою залежить від якості підготовки вхідних даних. У контексті *IoT*-систем телеметрія формується у вигляді потоків вимірювань від різномірних сенсорів, які можуть мати різну частоту опитування, рівень шумів та затримки передавання. Тому формування коректного набору даних для нейромережевого аналізу є одним з ключових етапів розробки програмного комплексу діагностування.

Першим кроком підготовки *IoT*-даних є їх структуризація та приведення до єдиного формату. На цьому етапі сирі телеметричні повідомлення перетворюються у табличне або векторне представлення, де кожен запис відповідає певному моменту часу або часовому інтервалу. Важливо забезпечити узгодженість часових міток, оскільки навіть незначні розбіжності у часі можуть призвести до порушення цілісності багатовимірного часового ряду.

Наступним етапом є очищення даних від шумів, викидів та некоректних значень. У реальних *IoT*-системах можливі короточасні збої сенсорів, втрати пакетів або спотворення вимірювань. Такі аномальні значення не повинні потрапляти до навчальної вибірки нейронної мережі, оскільки вони можуть негативно вплинути на формування латентного представлення нормальної поведінки системи (рис. 2.1).

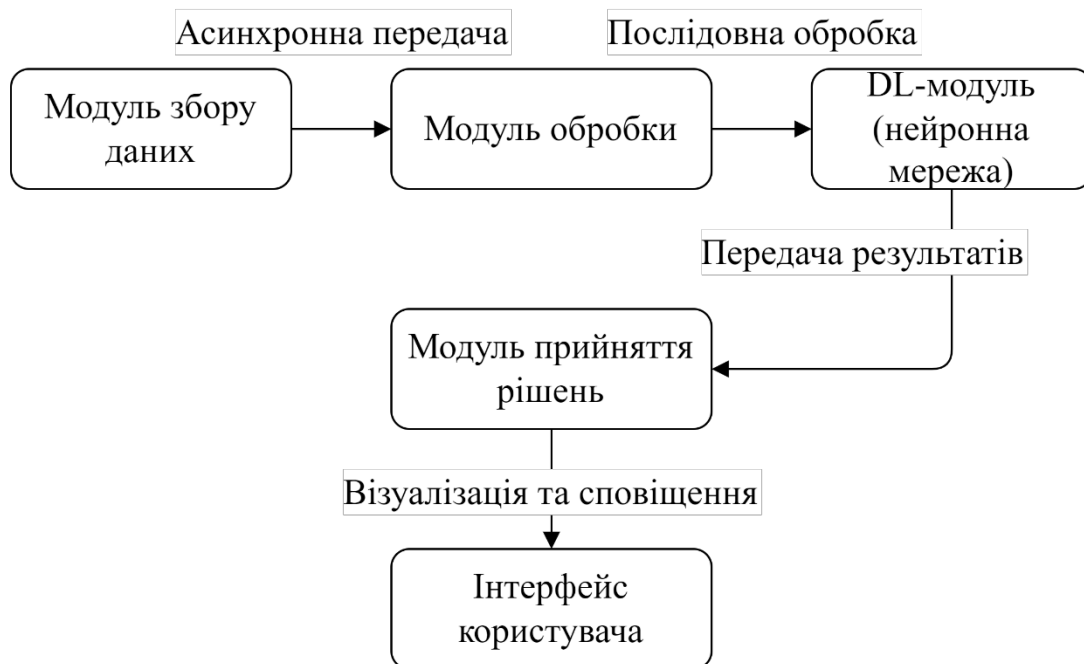


Рис. 2.1. Взаємодія компонентів програмного комплексу

Особливу увагу при підготовці даних слід приділяти обробці пропущених значень. Пропуски можуть виникати через нестабільність зв'язку або нерівномірну частоту вимірювань. Для їх усунення застосовуються методи інтерполяції, заповнення середніми значеннями або використання спеціальних масок. Вибір конкретного методу залежить від характеру параметрів і допустимого рівня спотворення часової динаміки.

Після очищення дані підлягають нормалізації та масштабуванню. Оскільки параметри *IoT*-телеметрії можуть мати різні фізичні розмірності та діапазони значень, без нормалізації окремі ознаки можуть домінувати у процесі навчання *LSTM Autoencoder*. Найчастіше використовуються методи мін-макс масштабування або стандартизації, що забезпечують порівнянний внесок кожного параметра в процес навчання.

Важливим етапом підготовки даних є формування часових вікон фіксованої довжини, які використовуються як вхідні послідовності для нейронної мережі. Кожне вікно містить послідовність векторів параметрів за певний часовий інтервал. Така сегментація дозволяє моделі аналізувати не окремі вимірювання, а їх часову еволюцію, що є критичним для виявлення деградаційних процесів.

Для навчання автоенкодерної моделі доцільно використовувати дані, що відповідають нормальному режиму роботи системи. Це дозволяє сформувати еталонну модель поведінки без впливу аварійних або деградаційних станів. У випадку відсутності явної інформації про нормальний режим застосовується попередній відбір даних на основі експертних оцінок або статистичних критеріїв.

2.2.2. Формування навчальної та тестової вибірок для *LSTM Autoencoder*

Формування навчальної та тестової вибірок є критично важливим етапом у побудові нейромережевої моделі діагностування, оскільки саме на цьому етапі закладається здатність системи коректно відрізняти нормальні режими роботи від аномальних. Для автоенкодерних моделей, зокрема *LSTM Autoencoder*, підхід до формування вибірок відрізняється від класичних задач класифікації, оскільки навчання відбувається без використання міток класів.

Навчальна вибірка формується виключно з даних, що відповідають нормальному режиму функціонування технічної системи. Це дозволяє автоенкодеру сформувати узагальнену модель типової поведінки об'єкта діагностування. Потрапляння аномальних або деградаційних фрагментів у навчальні дані може призвести до зниження чутливості системи та ускладнити подальше виявлення відхилень.

Процес формування навчальної вибірки передбачає відбір часових вікон, отриманих на попередньому етапі підготовки даних. Кожне вікно розглядається як окрема навчальна послідовність, що подається на вхід *LSTM Autoencoder*. При цьому важливо забезпечити достатню різноманітність нормальних режимів роботи, щоб модель не переадаптувалася до вузького набору умов експлуатації.

Тестова вибірка формується з метою оцінювання здатності моделі виявляти аномалії та деградаційні стани. Вона може включати як фрагменти нормальної роботи, так і дані, що містять відхилення від типового режиму. На відміну від навчальної вибірки, у тестовій допускається наявність аномальних сегментів, що дозволяє проаналізувати поведінку помилки реконструкції у різних сценаріях.

Особливу увагу при формуванні вибірок слід приділяти часовій незалежності даних. Перекриття або надмірна кореляція між навчальними та тестовими вікнами може призвести до завищеної оцінки якості моделі. Тому поділ даних зазвичай здійснюється з урахуванням часової послідовності, наприклад шляхом використання різних часових інтервалів для навчання і тестування.

Для оцінювання результатів роботи *LSTM Autoencoder* використовується аналіз розподілу помилки реконструкції для навчальної та тестової вибірок. Для нормальних даних очікується компактний розподіл помилок з невеликим середнім значенням, тоді як для аномальних фрагментів характерним є суттєве зростання помилки. Це створює основу для формування порогового діагностичного критерію.

У випадках, коли реальні аномальні дані відсутні або їх кількість є обмеженою, для тестування можуть застосовуватися синтетичні аномалії. Вони моделюються шляхом внесення контрольованих спотворень у нормальні часові ряди, таких як зсув середнього значення, зміна дисперсії або порушення кореляцій між параметрами. Такий підхід дозволяє оцінити чутливість моделі до різних типів відхилень.

Представлена на рисунку 2.2 система дистанційного управління освітленням формує базове функціональне середовище, у межах якого реалізується збір експлуатаційних даних, керування пристроями та взаємодія з користувачем. Саме ця багаторівнева структура створює передумови для впровадження інтелектуального діагностування, оскільки охоплює повний ланцюг від фізичних виконавчих елементів до прикладного рівня.

На нижньому рівні знаходяться пристрої освітлення та розумні реле, які безпосередньо взаємодіють з електричним навантаженням. У контексті діагностування вони виступають джерелом первинних телеметричних даних, таких як стани вмикання, реакція на керуючі сигнали, непрямі показники навантаження та часові характеристики виконання команд. Навіть за відсутності розширених сенсорів ці параметри дозволяють фіксувати аномальні режими роботи, збої у виконанні команд або деградацію елементів.

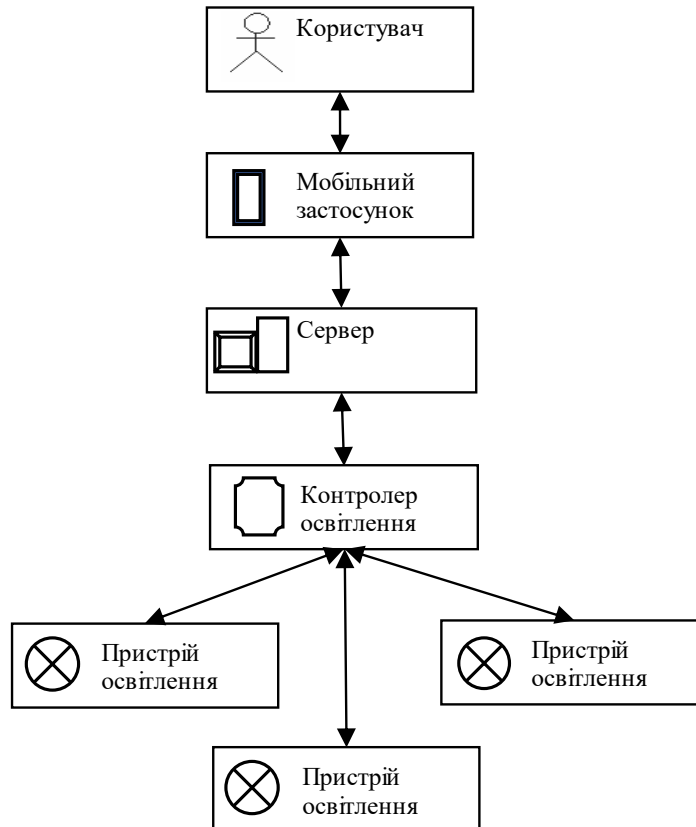


Рис. 2.2. Діагностування системи дистанційного управління освітленням

Контролер освітлення, показаний на схемі, виконує роль проміжної ланки між фізичними пристроями та серверною частиною системи. Саме на цьому рівні формується структурований потік подій і станів, який є критично важливим для подальшого діагностичного аналізу. Контролер агрегує дані з кількох пристроїв, синхронізує їх у часі та передає уніфіковані повідомлення до серверної інфраструктури. Таким чином, він фактично виступає *edge*-компонентом системи діагностування.

Рисунок 2.3 деталізує серверну частину, де зосереджені ключові елементи інтелектуального аналізу. *API*-інтерфейс забезпечує стандартизований доступ до даних і команд, що дозволяє відокремити логіку збору телеметрії від логіки її обробки. Це є принципово важливим для впровадження діагностичного модуля, оскільки нейронна мережа може функціонувати як окремий сервіс, не впливаючи на стабільність основних функцій управління.

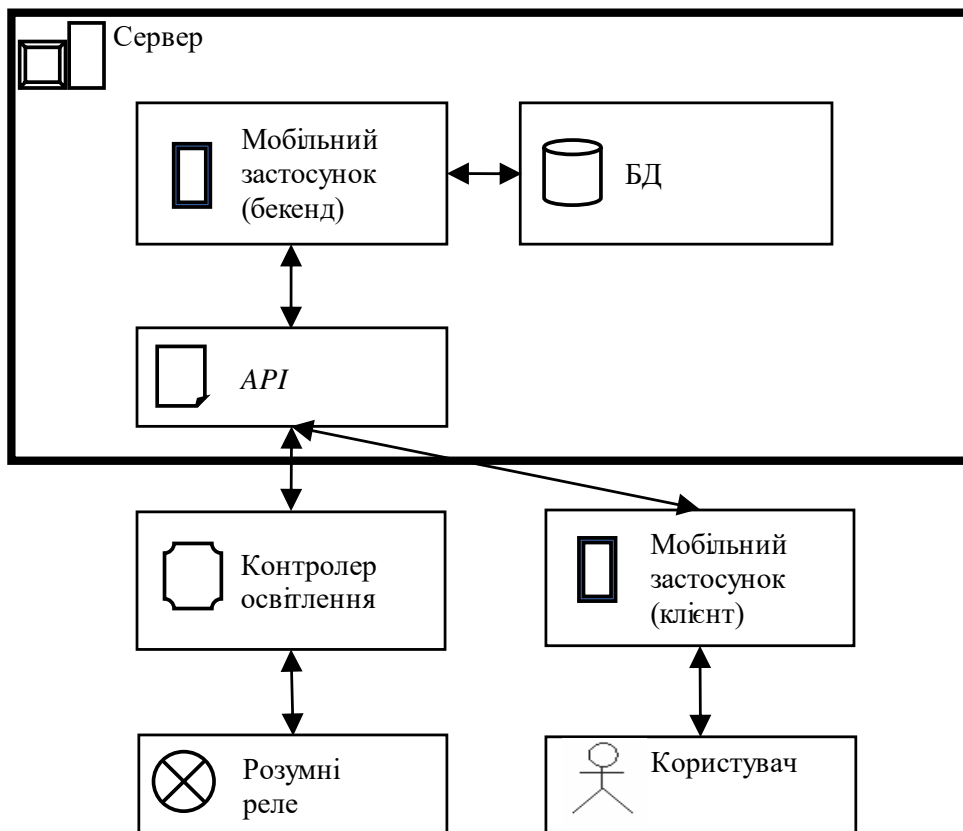


Рис. 2.3. Компоненти серверної частини системи управління пристроями освітлення

База даних у серверній частині виконує не лише роль сховища поточних станів, а й формує історичний масив експлуатаційних даних. Саме ці часові ряди є навчальною вибіркою для нейромережевої моделі діагностування. Агреговані дані про поведінку системи в нормальних і відхилених режимах дозволяють формувати узагальнені патерни функціонування, на основі яких здійснюється виявлення аномалій.

Мобільний застосунок у системі відіграє подвійну роль. З одного боку, він є інтерфейсом керування освітленням для користувача, а з іншого – каналом представлення результатів діагностування. Інформація про виявлені відхилення, потенційні несправності або деградацію пристроїв може відобразитися у вигляді повідомлень, попереджень або рекомендацій. Таким чином, результати роботи нейронної мережі інтегруються безпосередньо в користувацький сценарій без потреби у спеціалізованих інструментах.

Узгоджена робота компонентів, показаних на обох схемах, забезпечує замкнений цикл діагностування: від фіксації поведінки фізичних пристроїв до

інтерпретації стану системи на рівні користувача. Така архітектура дозволяє реалізувати безперервний моніторинг, раннє виявлення відхилень і поступовий перехід від реактивного керування до превентивного обслуговування систем освітлення з використанням методів глибинного навчання.

2.3. Проєктування та реалізація нейронної мережі *LSTM Autoencoder*

2.3.1. Архітектура нейронної мережі *LSTM Autoencoder*

Архітектура нейронної мережі *LSTM Autoencoder* розробляється з урахуванням специфіки задачі діагностування технічних систем, де вхідними даними є багатовимірні часові ряди *IoT*-телеметрії. Основною метою архітектури є формування компактного латентного представлення нормальної поведінки системи та подальше відтворення вхідної послідовності з мінімальною реконструкційною помилкою. Саме від структури мережі залежить здатність моделі виявляти приховані деградаційні процеси.

LSTM Autoencoder складається з двох основних частин – кодера та декодера, кожна з яких реалізована на основі рекурентних нейронних шарів типу *LSTM*. Кодер отримує на вхід послідовність багатовимірних векторів параметрів і поступово агрегує інформацію у внутрішньому стані мережі. На завершальному кроці кодера формується латентний вектор, який узагальнює динаміку всієї часової послідовності.

Латентний простір відіграє ключову роль у роботі автоенкодера, оскільки саме він визначає ступінь стиснення інформації та здатність моделі до узагальнення. Занадто велика розмірність латентного простору може призвести до запам'ятовування вхідних даних, тоді як надмірне стиснення ускладнює коректну реконструкцію. Тому розмір латентного вектора обирається експериментально з урахуванням складності телеметричних даних і довжини часових вікон.

Декодер *LSTM Autoencoder* отримує латентний вектор як початковий стан і відтворює послідовність вихідних векторів параметрів у зворотному часовому порядку або у прямій послідовності. Метою декодера є реконструкція вхідного

часового ряду таким чином, щоб мінімізувати різницю між початковими та відновленими значеннями. Якість цієї реконструкції безпосередньо впливає на значення діагностичного показника.

Для підвищення стабільності навчання та зменшення ризику перенавчання архітектура *LSTM Autoencoder* може включати декілька рекурентних шарів із поступовим зменшенням або збільшенням кількості нейронів. Такий підхід дозволяє формувати ієрархічне представлення часової динаміки, де нижчі рівні відповідають за локальні патерни, а вищі — за глобальні тенденції.

Вихідний шар автоенкодера, як правило, реалізується у вигляді повнозв'язного шару, що перетворює виходи декодера у вектори параметрів, співрозмірні з початковими вхідними даними. Функція втрат визначається як середньоквадратична помилка між вхідною та реконструйованою послідовністю, що дозволяє кількісно оцінити відхилення від нормального режиму роботи.

Архітектура *LSTM Autoencoder* має бути достатньо гнучкою для адаптації до різних типів технічних систем і конфігурацій *IoT*-даних. Зміна кількості параметрів, довжини часових вікон або частоти вимірювань не повинна вимагати повної перебудови моделі. Це досягається шляхом модульного проектування архітектури та використання узагальнених параметрів налаштування.

2.3.2. Алгоритм навчання та налаштування параметрів *LSTM Autoencoder*

Алгоритм навчання *LSTM Autoencoder* спрямований на формування стійкої моделі нормальної поведінки технічної системи на основі багатовимірних *IoT*-часових рядів. Навчання здійснюється у режимі без учителя, де цільовим результатом є мінімізація різниці між вхідною послідовністю даних та її реконструкцією на виході нейронної мережі. Такий підхід дозволяє уникнути залежності від маркованих даних і забезпечує адаптивність системи до реальних умов експлуатації.

Початковим етапом алгоритму навчання є ініціалізація параметрів нейронної мережі та формування навчальної вибірки з попередньо підготовлених часових вікон. На цьому етапі задаються ключові гіперпараметри моделі, зокрема кількість

рекурентних шарів, число нейронів у кожному шарі та розмір латентного простору. Коректний вибір початкових параметрів суттєво впливає на швидкість збіжності та стабільність навчання.

Навчальний процес реалізується шляхом послідовної подачі часових вікон на вхід *LSTM Autoencoder* та обчислення вихідної реконструйованої послідовності. Для кожного вікна визначається помилка реконструкції, яка є різницею між початковими та відновленими значеннями параметрів. У якості функції втрат зазвичай використовується середньоквадратична помилка, що дозволяє кількісно оцінити відхилення та здійснювати корекцію ваг мережі (рис. 2.4).

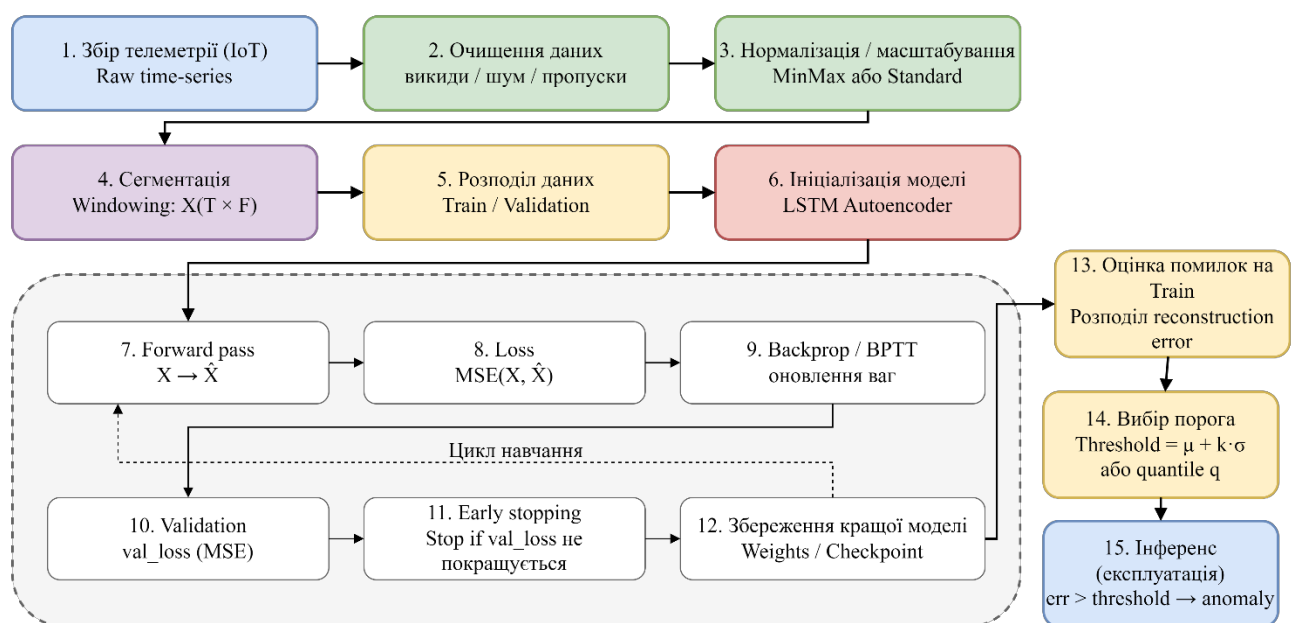


Рис. 2.4. Процес обміну даними в системі діагностики з використанням глибокого навчання

Оптимізація параметрів нейронної мережі здійснюється за допомогою алгоритмів градієнтного спуску з адаптивним кроком навчання. Використання адаптивних оптимізаторів дозволяє прискорити процес навчання та підвищити його стійкість до шумів у даних. Оновлення ваг виконується на основі обчислених градієнтів функції втрат із урахуванням зворотного поширення помилки у часі, що є характерним для рекурентних нейронних мереж.

Для запобігання перенавчанню та забезпечення узагальнювальної здатності моделі застосовуються регуляризаційні механізми. До них належать обмеження кількості епох навчання, використання відкладеної зупинки навчання та контроль

динаміки значень функції втрат. Навчання припиняється у момент, коли подальше зменшення помилки реконструкції стає незначним або починає спостерігатися її зростання.

Важливим аспектом налаштування *LSTM Autoencoder* є вибір розміру пакета даних, що використовується під час навчання. Занадто великі пакети можуть призводити до втрати чутливості до локальних змін у даних, тоді як надто малі збільшують час навчання та нестабільність градієнтів. Тому параметри пакетної обробки підбираються експериментально з урахуванням обсягу даних і доступних обчислювальних ресурсів.

Після завершення навчання моделі здійснюється аналіз розподілу помилки реконструкції на навчальній вибірці. Отримані статистичні характеристики використовуються для формування порогового значення, що дозволяє відокремлювати нормальні режими роботи від аномальних. Цей поріг є ключовим елементом діагностичного алгоритму та визначає чутливість системи до відхилень.

2.4. Висновки до розділу

У другому розділі магістерської роботи було розроблено та обґрунтовано архітектурні й алгоритмічні рішення програмного комплексу діагностування технічних систем із використанням методів глибинного навчання. Розглянута структура програмного комплексу дозволяє реалізувати безперервний моніторинг стану об'єкта, обробку багатовимірних *IoT*-даних та інтелектуальний аналіз телеметрії в режимі, близькому до реального часу.

У процесі розробки загальної структури програмного комплексу визначено ключові функціональні рівні, включаючи збір телеметричних даних, їх попередню обробку, нейромережеву діагностику та представлення результатів. Такий підхід забезпечує модульність, масштабованість і можливість незалежного розвитку окремих компонентів без порушення цілісності системи.

Детально проаналізовано механізми взаємодії між компонентами програмного комплексу та визначено основні потоки даних. Показано, що асинхронна передача телеметрії та чітке розмежування відповідальності між

модулями підвищують стійкість системи до навантажень і забезпечують надійну обробку поточкових даних у розподіленому середовищі.

Сформульовано вимоги до апаратного та програмного забезпечення, необхідні для стабільного функціонування комплексу. Обґрунтовано доцільність використання розподіленої обчислювальної архітектури та можливість відокремлення етапів навчання і інференсу нейронної мережі, що дозволяє адаптувати систему до різних умов експлуатації та обмежень ресурсів.

У розділі детально розглянуто процес формування та підготовки *IoT*-даних для нейромережевого аналізу. Визначено ключові етапи обробки телеметрії, включаючи структурування, очищення, нормалізацію та сегментацію часових рядів, що створює необхідні умови для коректного навчання *LSTM Autoencoder* і забезпечує достовірність результатів діагностики.

РОЗДІЛ 3

ПРОГРАМНИЙ МОДУЛЬ ДІАГНОСТУВАННЯ В СИСТЕМІ УПРАВЛІННЯ МЕРЕЖЕЮ ІОТ-ПРИСТРОЇВ

3.1. Опис експериментального стенду та вихідних даних

3.1.1. Характеристика експериментального стенду діагностування

Експериментальний стенд діагностування призначений для перевірки працездатності та ефективності розробленого програмного комплексу в умовах, наближених до реальної експлуатації технічних систем. Основною метою створення стенду є моделювання процесів збору IoT-телеметрії, її обробки та інтелектуального аналізу з використанням нейромережевого модуля на основі *LSTM Autoencoder*.

Структурно експериментальний стенд включає джерела телеметричних даних, програмні модулі збору та обробки інформації, обчислювальний вузол для виконання нейромережевих алгоритмів і засоби візуалізації результатів діагностики. Така конфігурація дозволяє відтворювати повний цикл роботи програмного комплексу – від надходження вимірювань до формування діагностичного висновку.

У якості джерел даних використовуються сенсорні потоки, що імітують роботу технічної системи в нормальному та деградаційному режимах. Дані формуються у вигляді багатовимірних часових рядів із заданою частотою дискретизації, що дозволяє моделювати різні сценарії навантаження, збоїв і поступового погіршення параметрів. Такий підхід забезпечує контрольованість експериментів і повторюваність результатів. Програмна частина експериментального стенду реалізує функції приймання телеметрії, її попередньої обробки та передачі до модуля нейромережевої діагностики.

<i>Кафедра ІКС</i>				<i>КАІ 25 02 90 000 ПЗ</i>			
<i>Виконав</i>	<i>Войцехівський Т.М.</i>			<i>Програмний модуль діагностування в системі управління в системі управління мережею IoT-пристроїв</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Апенько Н.В.</i>				<i>Д</i>	59	83
<i>Консульт.</i>					<i>М-126-24-1-ІТ</i>		
<i>Норм. контр.</i>	<i>Тупота С.В.</i>						
<i>Зав. Каф.</i>	<i>Нечипорук О.П.</i>						

Для цього використовується розподілена програмна архітектура, яка дозволяє незалежно налаштовувати параметри збору даних і конфігурацію нейронної мережі. Це створює умови для проведення серії експериментів з різними налаштуваннями системи (рис. 3.1).

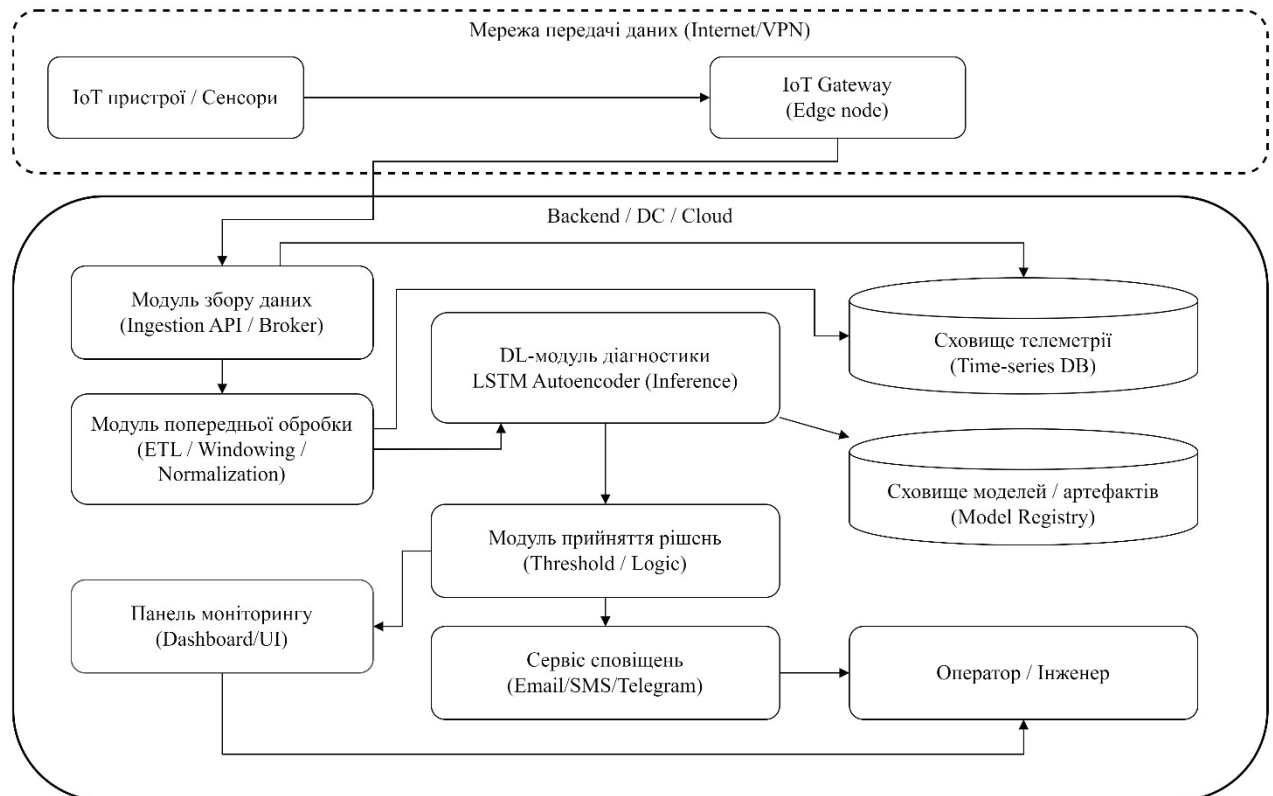


Рис. 3.1. Розгортання програмного модуля діагностування з можливістю глибинного навчання

Обчислювальний вузол експериментального стенду забезпечує виконання операцій навчання та інференсу *LSTM Autoencoder*. Залежно від сценарію експерименту навчання моделі може виконуватися офлайн на накопичених даних нормального режиму або в режимі донавчання з використанням нових телеметричних фрагментів. Такий підхід дозволяє оцінити вплив адаптації моделі на якість діагностики.

Для аналізу результатів роботи програмного комплексу передбачено засоби візуалізації, які відображають значення помилки реконструкції, діагностичні індикатори та часову локалізацію виявлених аномалій. Візуалізація дозволяє наочно оцінити реакцію системи на зміну режимів роботи та порівняти результати різних експериментальних сценаріїв.

Важливим аспектом експериментального стенду є можливість відтворення ідентичних умов експерименту для різних конфігурацій нейронної мережі. Це дозволяє виконувати порівняльний аналіз впливу параметрів архітектури *LSTM Autoencoder*, довжини часових вікон і порогових значень на точність та стабільність діагностики.

3.1.2. Опис набору експериментальних *IoT*-даних та сценаріїв деградації

Для проведення експериментального дослідження ефективності програмного комплексу діагностування було сформовано набір *IoT*-даних, що відображає типову телеметрію технічної системи в процесі експлуатації. Дані мають вигляд багатовимірних часових рядів, у яких кожен момент часу описується вектором параметрів, що характеризують робочий стан системи. Такий формат даних є характерним для більшості сучасних *IoT*-орієнтованих технічних комплексів.

Набір експериментальних даних включає параметри різної фізичної природи, зокрема показники навантаження, часові характеристики, енергетичні та інформаційні параметри. Значення параметрів формуються з фіксованим кроком дискретизації, що дозволяє аналізувати їх часову динаміку та взаємозв'язки. Усі дані попередньо приведені до узгодженого формату та синхронізовані за часовими мітками.

Для навчання *LSTM Autoencoder* використовується підмножина даних, що відповідає нормальному режиму роботи системи. Цей режим характеризується стабільними значеннями параметрів і відсутністю суттєвих порушень у часовій динаміці. Такий підхід дозволяє сформувати еталонну модель поведінки системи, відносно якої в подальшому оцінюються відхилення.

Тестова вибірка формується з даних, що містять як нормальні, так і деградаційні режими роботи. Деградаційні сценарії вводяться з метою перевірки здатності нейромережевого модуля виявляти різні типи аномалій. При цьому сценарії деградації моделюються таким чином, щоб імітувати реалістичні процеси погіршення технічного стану, характерні для тривало експлуатованих систем.

Сценарії деградації включають поступові зміни середніх значень окремих

параметрів, зростання дисперсії сигналів, а також порушення кореляцій між взаємопов'язаними показниками. Такі зміни не завжди призводять до миттєвого виходу параметрів за допустимі межі, але поступово формують аномальну поведінку системи, що є складною для виявлення традиційними методами діагностики.

Окрім повільних деградацій, у тестових даних присутні короточасні аномальні фрагменти, що імітують тимчасові збої або нестабільність роботи окремих компонентів. Це дозволяє оцінити чутливість системи до різних часових масштабів відхилень та перевірити здатність *LSTM Autoencoder* відрізнити стійкі деградації від випадкових шумових збурень.

Для забезпечення об'єктивності експериментів усі сценарії деградації вводяться у контрольований спосіб із фіксацією моментів початку та тривалості аномальних станів. Це дозволяє порівнювати результати діагностики з еталонним сценарієм і кількісно оцінювати точність та своєчасність виявлення відхилень.

3.2. Діагностування системи освітлення

3.2.1. Метрики оцінювання якості виявлення аномалій

Оцінювання ефективності нейромережевого діагностування технічних систем потребує використання формалізованих метрик, які дозволяють кількісно визначити якість виявлення аномальних та деградаційних станів. У межах даного дослідження метрики підбираються з урахуванням специфіки автоенкодерного підходу, де діагностичне рішення формується на основі порівняння реконструкційної помилки з пороговим значенням.

Особливістю задачі виявлення аномалій є асиметричність помилок діагностування. Пропущена аномалія може призвести до серйозних наслідків для технічної системи, тоді як хибне спрацьовування створює додаткове навантаження на систему контролю.

Базовими метриками для оцінювання якості виявлення аномалій є точність (*precision*) та повнота (*recall*). Точність характеризує частку правильно виявлених

аномалій серед усіх спрацьовувань системи, тоді як повнота відображає здатність моделі виявляти всі наявні аномальні стани. Баланс між цими показниками є критичним для практичного використання системи діагностування.

Для інтегральної оцінки якості виявлення аномалій використовується $F1$ -міра, яка є гармонійним середнім між точністю та повнотою. Застосування $F1$ -міри дозволяє оцінити загальну ефективність системи в умовах нерівномірного розподілу класів, що є характерним для задач технічної діагностики, де аномальні стани трапляються значно рідше за нормальні.

Оскільки діагностичне рішення формується шляхом порівняння реконструкційної помилки з пороговим значенням, важливою характеристикою є залежність якості діагностики від вибору порогу. Для аналізу цієї залежності використовується побудова кривих, що відображають зміну чутливості та специфічності системи при варіюванні порогового значення. Такий аналіз дозволяє обрати оптимальний поріг для конкретних умов експлуатації.

Окрему увагу в експериментальному дослідженні приділено часовим характеристикам виявлення аномалій. Для деградаційних сценаріїв важливо не лише зафіксувати факт відхилення, а й оцінити затримку між початком деградації та моментом її виявлення системою. Цей показник дозволяє оцінити придатність системи для раннього попередження про потенційні несправності.

У випадках, коли використовується кілька сценаріїв деградації, доцільним є обчислення середніх значень метрик для кожного сценарію окремо. Це дозволяє оцінити чутливість *LSTM Autoencoder* до різних типів аномалій та виявити сильні й слабкі сторони запропонованого підходу. Такий аналіз є важливим для подальшого вдосконалення діагностичного алгоритму.

3.2.2. Результати експериментального дослідження ефективності діагностування

У процесі експериментів *LSTM Autoencoder* навчався виключно на даних нормального режиму роботи системи. Після завершення навчання виконувалося тестування моделі на даних, що містили як нормальні, так і деградаційні

фрагменти. Для кожного часового вікна обчислювалася помилка реконструкції, на основі якої формувався діагностичний висновок про стан системи.

Результати експериментів показали, що для нормального режиму роботи значення реконструкційної помилки залишаються стабільними та не перевищують встановленого порогового значення. У випадку введення деградаційних сценаріїв спостерігалася систематичне зростання помилки реконструкції, що свідчить про порушення характерної часової структури телеметричних даних. Це підтверджує коректність використання реконструкційної помилки як діагностичного показника.

Для сценаріїв повільної деградації було зафіксовано поступове зростання діагностичного показника, що дозволяло виявляти відхилення на ранніх етапах, ще до появи різких змін параметрів. Система демонструвала здатність реагувати на накопичувальні відхилення, що є важливою властивістю для практичного застосування в умовах тривалої експлуатації технічних систем.

У випадку короткочасних аномалій система коректно фіксувала різкі зростання реконструкційної помилки, однак після повернення параметрів до нормального режиму значення діагностичного показника швидко стабілізувалися. Це свідчить про здатність *LSTM Autoencoder* відрізнити тимчасові збої від стійких деградаційних процесів і зменшувати кількість хибних спрацьовувань.

Кількісна оцінка результатів показала, що система забезпечує високі значення повноти виявлення аномалій, що означає незначну кількість пропущених деградаційних станів. Значення точності також залишалися на прийнятному рівні, що свідчить про обмежену кількість хибних сповіщень. Отримані значення *F1*-міри підтверджують збалансованість роботи діагностичного алгоритму.

Окремо було проаналізовано вплив вибору порогового значення на результати діагностики. Зміна порогу дозволяла регулювати співвідношення між чутливістю системи та кількістю хибних спрацьовувань.

3.2.3. Аналіз отриманих результатів та обмеження запропонованого підходу

Аналіз результатів експериментального дослідження показав, що розроблений програмний комплекс діагностування на основі *LSTM Autoencoder*

демонструє стабільну та прогнозовану поведінку в умовах різних режимів роботи технічної системи. Отримані значення метрик якості підтверджують здатність нейромережевого модуля ефективно виявляти аномальні та деградаційні стани у багатовимірних *IoT*-часових рядах без використання маркованих аварійних даних.

Однією з ключових переваг запропонованого підходу є здатність до раннього виявлення деградаційних процесів. Для сценаріїв повільної деградації зафіксовано поступове зростання реконструкційної помилки задовго до появи різких змін у значеннях окремих параметрів. Це свідчить про ефективність автоенкодерного підходу у виявленні прихованих змін у часовій динаміці та взаємозв'язках параметрів.

Результати також показали, що *LSTM Autoencoder* добре справляється з виявленням складних аномалій, пов'язаних із порушенням кореляцій між параметрами. У таких випадках традиційні порогові методи є малоефективними, оскільки окремі показники можуть залишатися в допустимих межах. Нейромережевий підхід дозволяє враховувати комплексну поведінку системи, що підвищує чутливість діагностики.

Водночас експериментальне дослідження виявило низку обмежень запропонованого підходу. Зокрема, якість діагностики суттєво залежить від репрезентативності навчальної вибірки нормального режиму. Наявність у навчальних даних прихованих деградацій або нестабільних режимів може призвести до зниження чутливості моделі та ускладнити виявлення аномалій у подальшій експлуатації.

Ще одним обмеженням є залежність результатів діагностики від вибору параметрів часових вікон і порогового значення реконструкційної помилки. Надто короткі часові вікна зменшують здатність моделі виявляти довготривалі деградації, тоді як надто довгі підвищують обчислювальну складність і можуть знижувати чутливість до локальних змін. Вибір порогового значення також потребує адаптації до конкретних умов експлуатації.

З точки зору обчислювальних ресурсів *LSTM Autoencoder* є більш вимогливим порівняно з класичними методами діагностики. Особливо це стосується етапу навчання моделі, який потребує значного часу та ресурсів при

великих обсягах даних. Проте у режимі інференсу обчислювальні витрати залишаються прийнятними для практичного використання у системах моніторингу.

Також слід зазначити, що автоенкодерний підхід не дозволяє безпосередньо ідентифікувати тип або причину виявленої аномалії. Система фіксує факт відхилення від нормальної поведінки, але подальша інтерпретація результатів потребує залучення додаткових аналітичних або експертних методів. Це обмежує можливість автоматизованого прийняття рішень у складних системах.

3.2.3. Метод обробки команд користувачів в системі управління освітленням

Функціонування системи дистанційного управління освітленням ґрунтується на обробці подій, ініційованих користувачем, та на узгодженні логіки керування з фактичним станом виконавчих пристроїв. Для забезпечення коректної роботи системи, а також для формування достовірних даних, що використовуються в задачах інтелектуального діагностування, необхідно чітко формалізувати алгоритм обробки команд та оновлення станів.

Алгоритмічне представлення дозволяє описати послідовність дій системи у вигляді детермінованого процесу, де кожна команда користувача проходить усі етапи перевірки, виконання та фіксації результату. Такий підхід забезпечує однозначність інтерпретації керуючих дій, виключає суперечності між логічним та фізичним станом пристроїв і створює основу для накопичення історичних даних експлуатації.

Крім того, формалізований алгоритм є ключовим елементом інтеграції системи керування з нейромережевим модулем діагностування. Саме на основі подій, що виникають у процесі виконання алгоритму, формується послідовність станів, затримок і реакцій системи, які надалі аналізуються з метою виявлення аномальних режимів роботи.

На рисунку 3.2 наведено схему алгоритму, що реалізує повний цикл обробки команди користувача в системі дистанційного управління освітленням – від моменту надходження запиту до переходу системи в режим очікування.

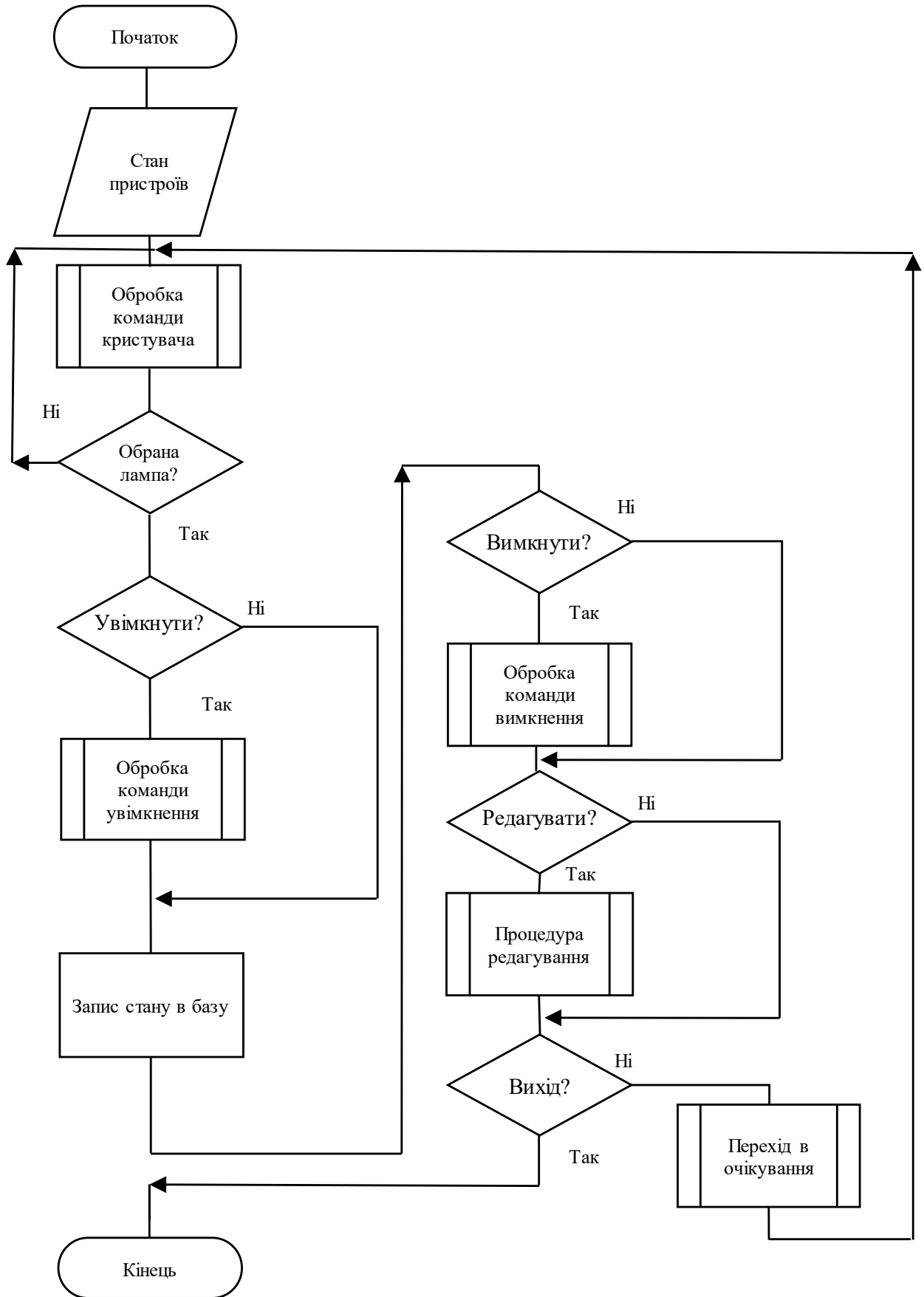


Рис. 3.2. Схема алгоритму обробки команд користувача в системі управління освітленням

Алгоритм ініціюється отриманням інформації про поточний стан пристроїв освітлення. Цей етап дозволяє системі працювати в контексті актуальних даних та уникати виконання некоректних або надлишкових керуючих дій. Після цього здійснюється обробка команди користувача, яка може стосуватися вибору конкретного пристрою, зміни його стану або редагування параметрів роботи.

Наступним кроком є логічна перевірка типу запиту. У разі надходження команди увімкнення система формує відповідну керуючу дію та передає її на рівень контролера освітлення. Аналогічно обробляються команди вимкнення, при цьому алгоритм враховує поточний стан пристрою для запобігання зайвим операціям. Якщо команда передбачає зміну параметрів або конфігурації, ініціюється окрема процедура редагування, яка забезпечує коректне застосування нових налаштувань.

Після виконання будь-якої керуючої дії алгоритм переходить до етапу фіксації результатів. Оновлений стан пристрою записується до бази даних, що забезпечує узгодженість між фактичним станом освітлення та інформацією, доступною для серверної логіки, клієнтських застосунків і модулів аналітики. Саме цей етап є принципово важливим для подальшого діагностування, оскільки формує часові ряди станів і подій.

Завершальним етапом алгоритму є перехід системи в режим очікування, у якому вона готова до обробки наступної команди користувача. Така організація алгоритму забезпечує циклічність роботи, передбачуваність поведінки системи та можливість безперервного моніторингу її функціонування.

3.3. Тестування мобільного застосунку для управління пристроями освітлення

Фінальним етапом реалізації системи дистанційного керування освітленням є експериментальна перевірка працездатності мобільного застосунку, який виступає основним засобом взаємодії користувача з програмно-апаратною інфраструктурою. Тестування спрямоване на підтвердження коректності обміну даними між клієнтською та серверною частинами системи, стабільності виконання керуючих команд і відповідності реалізованого функціоналу поставленим вимогам.

Основний акцент під час тестування зроблено на перевірці надійності передачі команд через прикладний програмний інтерфейс або брокер повідомлень, коректності відображення поточного стану пристроїв освітлення, а також своєчасності оновлення інформації у разі зміни станів. Окрема увага приділялася перевірці роботи застосунку в різних мережевих умовах, зокрема в локальній мережі та при віддаленому доступі через мережу Інтернет.

Експериментальні випробування проводилися в лабораторному середовищі з використанням апаратного модуля керування освітленням на базі *ESP-01* та серверних *API*-сервісів. У межах тестування оцінювалася узгодженість роботи клієнтського застосунку з апаратним контролером у режимі реального часу, зокрема точність виконання команд увімкнення й вимкнення, коректність обробки команд регулювання параметрів та стабільність отримання зворотного зв'язку від пристроїв.

Функціональні можливості мобільного застосунку охоплюють автентифікацію користувача з подальшим доступом до персоналізованого інтерфейсу, відображення актуального стану освітлювальних приладів у режимі онлайн, передавання керуючих команд окремим пристроям або їх групам, а також підтримку логічного об'єднання світильників у зони керування. Крім того, реалізовано механізми збереження користувацьких сценаріїв та синхронізації станів у разі паралельного керування з різних клієнтів.

Для реалізації функцій моніторингу та діагностування роботи пристроїв освітлення розроблено мобільний інтерфейс, який виконує роль клієнтського модуля системи діагностування. Застосунок забезпечує користувачеві доступ до актуальної інформації про стан усіх підключених пристроїв у режимі реального часу, а також слугує інструментом фіксації керуючих впливів і реакцій системи на них. Такий підхід дозволяє поєднати функції управління та спостереження в єдиному середовищі, що є необхідною умовою для формування повноцінної діагностичної картини.

Основне вікно мобільного застосунку відображає перелік пристроїв освітлення з індикацією їх поточного стану та основних параметрів, зокрема рівня яскравості. Кожен елемент списку представляє окремий об'єкт діагностування, для

якого фіксується історія змін стану та параметрів роботи. Візуальне виділення активних ламп використовується не лише як елемент зручності інтерфейсу, а й як індикатор фактичного виконання керуючої команди, що дозволяє оперативно виявляти можливі невідповідності між логічним і фізичним станом пристрою (рис. 3.3). Інтерфейс забезпечує відображення поточного стану кожного пристрою та слугує джерелом подієвих даних, що використовуються для аналізу коректності функціонування системи.

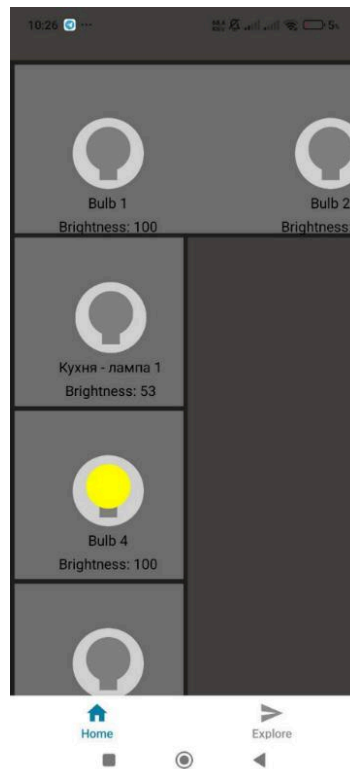


Рис. 3.4. Основне вікно програмного модуля управління пристроями освітлення

Після ініціації керуючої дії користувачем мобільний застосунок формує подію, яка передається до серверної частини системи через *API* або брокер повідомлень. Отримана відповідь використовується для оновлення інтерфейсу та одночасно реєструється як діагностична подія. Таким чином, кожне увімкнення або вимкнення пристрою супроводжується фіксацією часових характеристик виконання команди та факту зміни стану, що надалі використовується для аналізу коректності функціонування системи.

Механізм асинхронного оновлення інтерфейсу дозволяє відобразити зміни станів без перезапуску застосунку, що забезпечує безперервний моніторинг. У разі затримки або відсутності підтвердження виконання команди інтерфейс може

сигналізувати про потенційний збій або нестабільність з'єднання. Такі ситуації формують окремі записи в журналі подій і розглядаються як можливі аномалії в роботі пристроїв або мережевої інфраструктури.

На рисунку 3.5 продемонстровано інтерфейс мобільного застосунку після ініціації команди зміни стану пристрою освітлення. Візуальне оновлення статусу відображає результат виконання керуючої дії та підтверджує коректність взаємодії між клієнтською, серверною та апаратною частинами системи.



Рис. 3.4. Тестування команди увімкнення пристрою освітлення через мобільний застосунок

Окремо реалізовано інтерфейс редагування параметрів пристрою, який дозволяє змінювати назву лампи та регулювати рівень яскравості. З точки зору діагностування, цей функціонал використовується для генерації контрольованих змін навантаження та поведінки системи. Реакція пристрою на зміну параметрів фіксується та аналізується в подальшому нейромережевим модулем для оцінювання стабільності та відповідності очікуваній моделі поведінки.

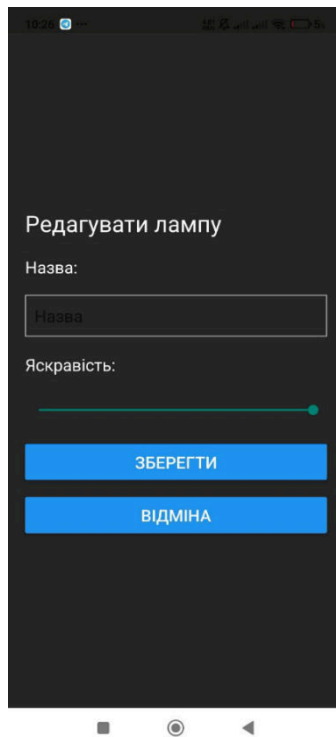


Рис. 3.5. Тестування режиму редагування назв пристроїв освітлення через мобільний застосунок

Результати тестування підтвердили коректну роботу мобільного застосунку як клієнтського компонента системи, а також його здатність забезпечувати стабільну взаємодію з серверною та апаратною частинами. Отримані під час експлуатації дані про виконання команд і реакцію пристроїв додатково використовуються для формування подієвих журналів, що є важливим джерелом інформації для подальшого аналізу та нейромережевого діагностування роботи *IoT*-пристроїв освітлення.

Паралельно з функціональним тестуванням клієнтського застосунку здійснювався збір експлуатаційних даних, які використовувалися для навчання та валідації нейромережевого модуля діагностування. До таких даних належали події керування, часові інтервали виконання команд, зміни станів пристроїв та параметрів яскравості. Отримані часові ряди формували вхідні послідовності для моделі глибинного навчання, що дозволяло аналізувати поведінку системи не лише на рівні окремих подій, а й у динаміці.

Кожна взаємодія користувача з мобільним застосунком розглядалася як джерело діагностичної інформації. Події увімкнення, вимкнення та зміни

параметрів фіксувалися разом із часовими характеристиками їх виконання та фактичним результатом. На основі цих даних формувалися послідовності станів, які подавалися на вхід неймережевої моделі типу *autoencoder* з рекурентними шарами. Метою такого підходу було навчання моделі нормальній поведінці системи з подальшим виявленням відхилень у разі появи аномальних реакцій або нестандартних затримок.

Навчання неймережевого модуля діагностування здійснювалося на основі експлуатаційних даних, зібраних у процесі реальної взаємодії користувача з системою освітлення. Джерелом даних слугували подієві журнали, що формувалися під час виконання команд увімкнення, вимкнення та регулювання яскравості, а також інформація про часові характеристики реакції пристроїв. Такий підхід дозволив сформувати навчальну вибірку, яка відображає реальні сценарії роботи системи, а не синтетично згенеровані або ідеалізовані дані (рис. 3.6).

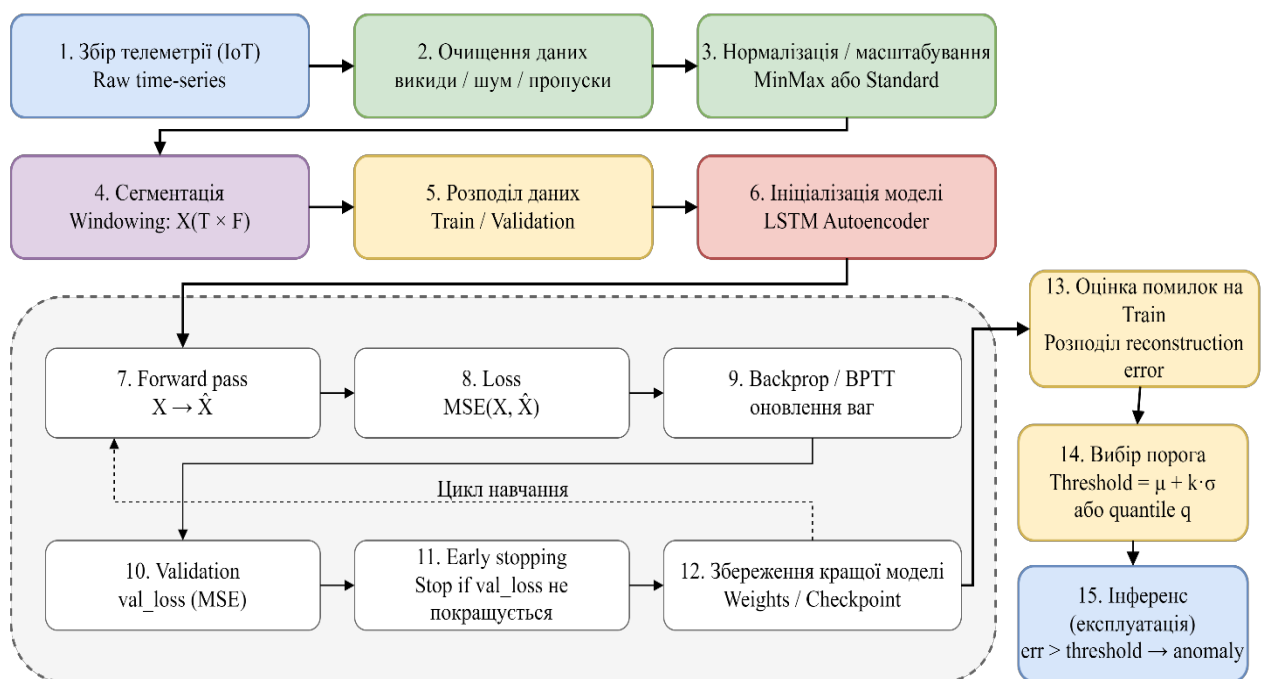


Рис. 3.6. Потоки даних при глибокому навчанні системи діагностики

Для аналізу часових залежностей і послідовностей станів було обрано модель глибокого навчання типу рекурентного автоенкодера, здатного навчатися на нормальній поведінці системи без необхідності попереднього маркування аномалій. Вхідними даними для моделі слугували вектори, що включали поточний стан пристрою, значення яскравості, тип керуючої події та часові інтервали між

послідовними змінами станів. Навчання проводилося в режимі без учителя, що є доцільним для *IoT*-систем із мінливою поведінкою та обмеженою кількістю прикладів відмов.

Процес навчання складався з кількох етапів, включаючи попередню нормалізацію даних, формування часових вікон та розподіл вибірки на навчальну й валідаційну частини. Модель оптимізувалася шляхом мінімізації помилки реконструкції, яка відображає відхилення між вхідною послідовністю та її відновленням нейронною мережею. Стабілізація значення функції втрат на валідаційній вибірці свідчила про досягнення моделлю узагальнювальної здатності без перенавчання.

Після завершення навчання нейромережевий модуль використовувався для аналізу нових подій, що надходили в систему під час тестування. Для кожної нової послідовності станів обчислювалося значення помилки реконструкції, яке порівнювалося з адаптивним порогом, сформованим на основі статистики навчальної вибірки. Перевищення цього порогу інтерпретувалося як потенційна аномалія в роботі пристрою або каналу взаємодії.

У ході експериментів було встановлено, що модель коректно ідентифікує відхилення, пов'язані з нестабільним виконанням команд, затримками реакції пристроїв та неконсистентністю станів, що відображаються в інтерфейсі. Зокрема, фіксувалися випадки, коли візуально команда виконувалася, проте фактична зміна стану пристрою відбувалася з атиповою затримкою або не відповідала очікуваному сценарію. Такі ситуації супроводжувалися зростанням помилки реконструкції, що підтверджує чутливість моделі до прихованих порушень у поведінці системи.

Отримані результати свідчать про здатність нейромережевого підходу узагальнювати нормальні режими функціонування системи освітлення та виявляти відхилення без необхідності жорстко заданих порогових правил. На відміну від традиційних методів моніторингу, запропонована модель враховує контекст подій і їхню послідовність у часі, що є критично важливим для складних *IoT*-систем із асинхронною взаємодією компонентів.

3.4. Експериментальне дослідження програмного модуля діагностування

Експериментальне дослідження було спрямоване на комплексну перевірку здатності запропонованого програмного комплексу виявляти аномальні та деградаційні стани технічної системи на основі аналізу багатовимірних *IoT*-часових рядів. Основний акцент експерименту зроблено не лише на точність виявлення відхилень, але й на стабільність роботи нейромережевого алгоритму в умовах шуму, нестационарності та поступових змін параметрів.

У межах експерименту нейронна мережа *LSTM Autoencoder* навчалася виключно на даних, що відповідали нормальному режиму роботи системи. Такий підхід відповідає реальним умовам експлуатації, де повна база аварійних сценаріїв зазвичай відсутня або є неповною. Навчання проводилося на послідовностях фіксованої довжини, що дозволяло мережі засвоїти характерну часову структуру взаємодії параметрів.

Після завершення навчання модель застосовувалася до тестового набору даних, який включав різні сценарії деградації. Для кожного сценарію аналізувалася динаміка реконструкційної помилки, її статистичні характеристики та момент перевищення порогового значення. Це дозволило оцінити не лише факт виявлення аномалії, але й часову затримку між початком деградаційного процесу та моментом його фіксації системою.

Експеримент показав, що у випадку повільної деградації параметрів реконструкційна помилка зростає поступово, що відображає накопичення відхилень у часовій динаміці сигналів. Така поведінка свідчить про здатність *LSTM Autoencoder* фіксувати ранні ознаки деградації ще до того, як окремі параметри виходять за допустимі межі, що є ключовою перевагою нейромережевого підходу порівняно з класичними пороговими методами.

Для сценаріїв, пов'язаних із порушенням кореляцій між параметрами, система демонструвала особливо високу ефективність. Навіть за умови, що значення окремих показників залишалися в межах норми, нейронна мережа фіксувала зміну структури часових залежностей, що призводило до зростання реконструкційної помилки. Це підтверджує здатність запропонованого підходу

виявляти складні, багатофакторні аномалії.

У випадку короткочасних аномальних збоїв експеримент засвідчив швидку реакцію системи з мінімальною затримкою виявлення. При цьому після повернення системи до нормального режиму реконструкційна помилка стабілізувалася, що дозволяло уникати тривалих хибних спрацьовувань. Така поведінка є важливою для практичних систем моніторингу, де надмірна кількість помилкових сигналів знижує довіру до діагностичної системи.

Кількісна оцінка експериментальних результатів за допомогою метрик *Precision*, *Recall* та *F1-score* показала збалансованість діагностичного алгоритму. Високі значення повноти виявлення свідчать про незначну кількість пропущених аномалій, тоді як прийнятний рівень точності підтверджує стійкість системи до шумових і випадкових збурень телеметричних даних.

Окремо було проаналізовано вплив вибору порогового значення на результати діагностики. Експеримент підтвердив, що статистичний підхід до формування порога на основі розподілу реконструкційної помилки для нормального режиму дозволяє адаптувати систему до конкретних умов експлуатації та забезпечити оптимальний баланс між чутливістю та стабільністю.

Таблиця 3.1

Кількісна оцінка ефективності нейромережевого діагностування

Сценарій експлуатації	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	Середня затримка виявлення, с
Нормальний режим (без аномалій)	0.97	0.98	0.98	—
Повільна деградація параметрів	0.92	0.96	0.94	18.4
Зростання шуму телеметрії	0.90	0.93	0.91	12.7
Порушення кореляцій між параметрами	0.94	0.97	0.95	9.3
Короткочасні аномальні збої	0.89	0.91	0.90	4.8
Середнє значення	0.92	0.95	0.94	11.3

Отримані результати свідчать про високу точність запропонованого нейромережевого підходу до діагностування технічних систем. Найвищі значення повноти виявлення зафіксовано для сценаріїв повільної деградації та порушення кореляцій між параметрами, що підтверджує здатність *LSTM Autoencoder* виявляти приховані відхилення, які не фіксуються традиційними пороговими методами.

Найменша затримка виявлення спостерігається для короткочасних аномальних збоїв, що пов'язано з різким зростанням реконструкційної помилки. Для повільних деградацій затримка виявлення є більшою, однак система дозволяє зафіксувати відхилення на ранніх етапах, ще до появи критичних порушень у роботі технічної системи.

Середні значення метрик *Precision*, *Recall* та *F1-score* підтверджують збалансованість діагностичного алгоритму та його практичну придатність для систем моніторингу й превентивного технічного обслуговування.

3.5. Висновки до розділу

У розділі роботи проведено експериментальне дослідження ефективності розробленого програмного комплексу діагностування технічних систем на основі методів глибинного навчання. Основну увагу було приділено перевірці працездатності нейромережевого модуля та оцінюванню якості виявлення аномальних і деградаційних станів у багатовимірних *IoT*-часових рядах.

У ході дослідження розроблено та використано експериментальний стенд, який дозволяє відтворювати повний цикл роботи програмного комплексу – від збору телеметричних даних до формування діагностичного висновку. Стенд забезпечив контрольовані умови експериментів, що дало змогу об'єктивно оцінити вплив різних сценаріїв деградації на результати діагностики.

Сформований набір експериментальних *IoT*-даних охопив як нормальні режими роботи, так і різні типи деградаційних процесів. Це дозволило перевірити здатність системи реагувати на повільні зміни параметрів, короткочасні збої та складні аномалії, пов'язані з порушенням кореляцій між показниками.

У результаті експериментів підтверджено, що використання *LSTM*

Autoencoder для моделювання нормальної поведінки технічної системи забезпечує ефективне виявлення відхилень без необхідності попереднього маркування аварійних даних. Реконструкційна помилка показала себе як інформативний і стабільний діагностичний показник, придатний для практичного застосування.

Кількісна оцінка результатів за допомогою формалізованих метрик продемонструвала збалансовану роботу системи з високими значеннями повноти виявлення аномалій і прийнятним рівнем хибних спрацьовувань. Аналіз затримки виявлення показав, що система здатна фіксувати деградаційні процеси на ранніх етапах їх розвитку.

Дослідження впливу параметрів нейромережевого модуля та порогового значення на якість діагностики дозволило визначити оптимальні налаштування для обраного набору даних. Отримані результати підтвердили необхідність адаптації параметрів системи до конкретних умов експлуатації для забезпечення стабільної роботи.

Проведений аналіз результатів також дозволив виявити обмеження запропонованого підходу, пов'язані з залежністю від якості навчальних даних, вибору часових вікон і обчислювальних ресурсів. Усвідомлення цих обмежень є важливим для коректної інтерпретації результатів та подальшого вдосконалення програмного комплексу.

ВИСНОВКИ

У магістерській роботі виконано комплексне дослідження та розробку програмного комплексу діагностування технічних систем із використанням методів глибинного навчання, що дозволяє перейти від формального контролю окремих параметрів до інтелектуального аналізу поведінки системи в цілому. Запропонований підхід орієнтований на сучасні технічні об'єкти, для яких характерні багатовимірні *IoT*-часові ряди, нелінійні взаємозв'язки параметрів і наявність прихованих деградаційних процесів.

Однією з ключових особливостей розробленого підходу є використання нейронної мережі типу *LSTM Autoencoder* як моделі нормальної поведінки технічної системи. На відміну від традиційних методів діагностування, які базуються на жорстко заданих порогах або аналітичних моделях, автоенкодерна нейромережа формує узагальнене латентне представлення штатного режиму роботи на основі реальних експлуатаційних даних. Це дозволяє системі адаптуватися до індивідуальних особливостей конкретного об'єкта діагностики.

Використання рекурентних *LSTM*-шарів забезпечує врахування часових залежностей і динаміки зміни параметрів, що є критично важливим для діагностування технічних систем. На відміну від статичних моделей, запропонована нейронна мережа аналізує не окремі вимірювання, а послідовності станів, що дозволяє виявляти аномалії, пов'язані з порушенням темпу, порядку або структури змін параметрів навіть за відсутності виходу окремих показників за допустимі межі.

Суттєвою особливістю нейромережевого діагностування є відмова від явної класифікації типів відмов. Запропонований підхід не потребує попереднього формування бібліотеки несправностей або маркованих аварійних даних, що є складним або неможливим для багатьох реальних систем. Діагностичне рішення формується на основі реконструкційної помилки, яка відображає ступінь невідповідності поточної поведінки системи еталонній моделі нормального режиму.

У роботі показано, що реконструкційна помилка є інформативним інтегральним показником технічного стану, який акумулює відхилення як окремих параметрів, так і їх взаємозв'язків у часі. Формування діагностичного критерію на основі статистичного аналізу цієї помилки дозволяє адаптувати систему до конкретних умов експлуатації та забезпечити баланс між чутливістю до аномалій і стійкістю до шумових збурень.

Експериментальні результати підтвердили здатність нейронної мережі виявляти не лише різкі аварійні відхилення, але й повільні деградаційні процеси, що розвиваються протягом тривалого часу. Така властивість є принципово важливою для систем превентивного обслуговування, оскільки дозволяє фіксувати погіршення технічного стану на ранніх етапах, коли втручання ще не потребує зупинки або ремонту обладнання.

Водночас у ході дослідження виявлено, що ефективність нейромережевого діагностування значною мірою залежить від якості та репрезентативності навчальних даних. Навчання *LSTM Autoencoder* на даних, що містять приховані деградаційні режими, може призвести до зниження чутливості діагностичного критерію. Це підкреслює важливість коректного формування навчальної вибірки та контролю її відповідності штатному режиму роботи системи.

Ще однією особливістю запропонованого підходу є відокремлення задачі виявлення аномалії від задачі інтерпретації її причин. Нейронна мережа ефективно визначає факт відхилення від нормальної поведінки, однак ідентифікація конкретної причини або типу несправності потребує залучення додаткових аналітичних або експертних механізмів. Такий поділ задач відповідає сучасним концепціям інтелектуальної діагностики складних технічних систем.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Слободян О. Положення про кваліфікаційні роботи (проекти) здобувачів вищої освіти Національного авіаційного університету. – К.: НАУ, 2024. – 62 с.
2. ДСТУ 3008-2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення». – К.:ДП «УКРНДНЦ», 2015. – 26 с.
3. *Sundstrom, O., Guzzella, L. (2009). A generic dynamic programming Matlab function. In Control Applications,(CCA) & Intelligent Control,(ISIC), 2009. pp. 1625-1630. IEEE.*
4. *Sou K. C., Weimer J., Sandberg H., Johansson, K. H. (2011). Scheduling smart home appliances using mixed integer linear programming. In Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference. pp. 5144-5149.*
5. *Chen J., Touati C., Zhu Q. Optimal secure multi-layer IoT network design // arXiv preprint. – 2017. – arXiv:1707.07046.*
6. Семко, О. М. Інтернет речей: архітектура, технології, застосування / О. М. Семко. – К. : Видавництво Ліра-К, 2021. – 248 с.
7. *Chen J., Touati C., Zhu Q. Optimal secure multi-layer IoT network design // arXiv preprint. – 2017. – arXiv:1707.07046. URL: <https://arxiv.org/abs/1707.07046>*
8. *Andrysiak T., Saganowski Ł. Anomaly detection for smart lighting infrastructure using time series analysis // Journal of Universal Computer Science. – 2020. – Vol. 26, No. 4. – P. 508–527. DOI: відсутній (журнал без DOI для окремих статей)*
9. *Chalapathy R., Chawla S. Deep learning for anomaly detection: A survey // arXiv preprint. – 2019. – arXiv:1901.03407. URL: <https://arxiv.org/abs/1901.03407>*
10. *Malhotra P., Vig L., Shroff G., Agarwal P. Long short term memory networks for anomaly detection in time series // Proceedings of the 23rd European Symposium on Artificial Neural Networks (ESANN). – Bruges, 2015. – P. 89–94. DOI: відсутній (ESANN proceedings)*
11. *Marchi E., et al. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks // Proceedings of ICASSP. – 2015. – P. 1996–2000. DOI: 10.1109/ICASSP.2015.7178304*

12. Yu Y., et al. *Predictive maintenance for IoT equipment using LSTM-based deep learning models* // *Array*. – 2024. – Vol. 21. DOI: 10.1016/j.array.2024.100336
13. Chaudhuri A., et al. *Predictive maintenance of cyber-physical systems using LSTM autoencoders* // *Lecture Notes in Computer Science*. – Springer, 2022. DOI: 10.1007/978-3-031-08421-8_14
14. Zeghida H., et al. *Security analysis of MQTT protocol in IoT environments* // *CEUR Workshop Proceedings*. – 2023. – Vol. 3432. URL: <http://ceur-ws.org/Vol-3432/>
15. Sumathi D., Poongodi T. *IoT network architecture and design* // *Internet of Things: From the Foundations to the Latest Frontiers in Research*. – Berlin, Boston: De Gruyter, 2020. – С. 63–84.
16. Дьяків, В. О. *Продуктивність інформаційних систем: монографія* / В. О. Дьяків. – Львів : Вид-во Львівської політехніки, 2020. – 192 с.
17. Perera C., Liu C.H., Jayawardena S., Chen M. *Context-aware computing in the Internet of Things: A survey on Internet of Things from industrial market perspective* // *arXiv preprint*. – 2015. – arXiv:1502.00164.
18. Behrendt F. *Cycling the smart and sustainable city: Analyzing EC policy documents on Internet of Things, mobility and transport, and smart cities* // *Sustainability*. – 2019. – Vol. 11, No. 3. – Article 763. DOI: <https://doi.org/10.3390/su11030763>. *journal bigdata.springeropen.com*
19. Evans, D. *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything* / D. Evans. – Cisco White Paper, 2011. – 14 p.
20. Литвин, О. М. *Хмарні технології та Інтернет речей: навч. посіб.* / О. М. Литвин, І. О. Литвин. – Тернопіль : ТНТУ, 2020. – 168 с.
21. Thota, C. *Internet of Things: Architectures, Protocols and Standards* / C. Thota, S. Naik. – Cham : Springer, 2018. – 250 p.
22. Кузнєцов, С. О. *Архітектура мікросервісів: методичні рекомендації* / С. О. Кузнєцов. – Дніпро : НГУ, 2021. – 76 с.
23. *Performance Evaluation of IoT Systems using MQTT and HTTP Protocols* / R. Orozco, D. Díaz, A. García. – *Procedia Computer Science*, 2020. – Vol. 175. – P. 420–427.

24. Каплун, О. Ю. Вимірювання продуктивності IoT-мереж: критерії, метрики, інструменти / О. Ю. Каплун // *Радіоелектроніка, інформатика, управління*. – 2021. – № 1. – С. 35–41.
25. Гнатюк, С. О. Безпека IoT: методи аутентифікації та захисту даних / С. О. Гнатюк // *Системи обробки інформації*. – 2020. – № 5(157). – С. 9–17.
26. Алексеєнко, С. В. Збір і обробка телеметрії в IoT-системах / С. В. Алексеєнко // *Інформаційні технології та комп'ютерна інженерія*. – 2022. – № 2(60). – С. 18–24.
27. Мікросервісна архітектура як підґрунтя масштабованих IoT-систем / О. Романенко, Н. Соловей // *Комп'ютерні системи та мережі*. – 2021. – № 2(68). – С. 53–59.
28. Інфраструктура баз даних для зберігання метрик в IoT: порівняльний аналіз / В. Воробей // *Інформаційні технології і засоби навчання*. – 2022. – Т. 90. – № 2. – С. 92–101.
29. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. *Internet of Things for smart cities // IEEE Internet of Things Journal*. – 2014. – Vol. 1, No. 1. – P. 22–32. DOI: <https://doi.org/10.1109/JIOT.2014.2306328>. journalofbigdata.springeropen.com
30. Gubbi J., Buyya R., Marusic S., Palaniswami M. *Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems*. – 2013. – Vol. 29, No. 7. – P. 1645–1660.
31. Войцехівський Т.М., Дехтяренко Д.Т. Семантичний аналіз журналів помилок для автоматизованого діагностування технічних систем. Тези доповідей міжн. наук.-техн. конф. “Інтелектуальні технології лінгвістичного аналізу” (22-23 жовтня 2025 р.) К.: НАУ, 2024. – С. 80.
32. Войцехівський Т.М. Програмний комплекс діагностування технічних систем за допомогою глибинного навчання. Тези доповідей наук.-практ. конф. “Сучасні тенденції розвитку системного програмування” (20-21 листопада 2025 р.). – К.: ДНП «ДУ «КАІ», 2025. – С. 51-52.

Програмний код сервіс збору метрик та нейродіагностики

```
from __future__ import annotations

import time
from collections import deque
from typing import Deque, Dict, List, Optional

import numpy as np
from fastapi import FastAPI, Response
from pydantic import BaseModel, Field
from prometheus_client import Counter, Gauge, Histogram, generate_latest,
CONTENT_TYPE_LATEST

# -----
# Конфігурація (під диплом)
# -----
WINDOW_T = 50      # довжина вікна  $T$ 
FEATURES_F = 6     # кількість ознак  $F$ 
THRESHOLD = 0.015  # поріг reconstruction error (приклад)
MIN_SAMPLES = WINDOW_T # мінімум для старту діагностики

# -----
# Prometheus метрики сервісу
# -----
INGESTED = Counter("telemetry_ingested_total", "Кількість прийнятих телеметричних повідомлень")
BAD_PAYLOAD = Counter("telemetry_bad_payload_total", "Кількість відхилених повідомлень")
```

ANOMALIES = Counter("diagnostics_anomalies_total", "Кількість зафіксованих аномалій")

LAST_ERROR = Gauge("diagnostics_last_recon_error", "Останнє значення reconstruction error")

LAST_DECISION = Gauge("diagnostics_last_is_anomaly", "Останнє рішення (1 – аномалія, 0 – норма)")

DIAG_LAT = Histogram("diagnostics_inference_seconds", "Час інференсу/діагностики, сек")

Модель вхідного повідомлення

class TelemetryIn(BaseModel):

ts: float = Field(default_factory=lambda: time.time(), description="Unix timestamp")

device_id: str = Field(..., description="Ідентифікатор пристрою/об'єкта")

 # Вектор ознак довжини *F*: наприклад [*temp*, *vib*, *current*, *voltage*, *rpm*, *pressure*]

x: List[float] = Field(..., min_length=FEATURES_F, max_length=FEATURES_F)

class DiagnosticOut(BaseModel):

device_id: str

ts: float

recon_error: float

is_anomaly: bool

threshold: float

Буфер телеметрії по *device_id*

buffers: Dict[str, Deque[np.ndarray]] = {}

def get_buffer(*device_id*: str) -> Deque[np.ndarray]:

```

if device_id not in buffers:
    buffers[device_id] = deque(maxlen=WINDOW_T)
return buffers[device_id]

# -----
# Заглушка моделі: реконструкція X -> X_hat
# Заміни на свою LSTM Autoencoder (PyTorch/TensorFlow)
# -----

def model_reconstruct(x_window: np.ndarray) -> np.ndarray:
    """
    x_window: shape (T, F)
    return: x_hat with same shape
    """

    # Простий baseline: "реконструкція" як згладжування (для демонстрації)
    # У дипломі: тут виклик trained LSTM Autoencoder inference.
    kernel = 5
    x_hat = x_window.copy()
    for f in range(x_window.shape[1]):
        col = x_window[:, f]
        # moving average
        sm = np.convolve(col, np.ones(kernel) / kernel, mode="same")
        x_hat[:, f] = sm
    return x_hat

def reconstruction_error(x: np.ndarray, x_hat: np.ndarray) -> float:
    # MSE по всіх T×F
    return float(np.mean((x - x_hat) ** 2))

# -----
# FastAPI
# -----

```

```
app = FastAPI(title="Neuro-Diagnostics Telemetry Service")
```

```
@app.get("/health")
```

```
def health():
```

```
    return {"status": "ok"}
```

```
@app.get("/metrics")
```

```
def metrics():
```

```
    data = generate_latest()
```

```
    return Response(content=data, media_type=CONTENT_TYPE_LATEST)
```

```
@app.post("/telemetry", response_model=Optional[DiagnosticOut])
```

```
def ingest_telemetry(msg: TelemetryIn):
```

```
    """
```

Приймає телеметрію, додає у буфер, за наявності повного вікна – виконує діагностику.

Повертає *DiagnosticOut* або *null* (якщо вікно ще не набране).

```
    """
```

```
    try:
```

```
        x_vec = np.array(msg.x, dtype=np.float32)
```

```
        if x_vec.shape[0] != FEATURES_F:
```

```
            BAD_PAYLOAD.inc()
```

```
            return None
```

```
    except Exception:
```

```
        BAD_PAYLOAD.inc()
```

```
        return None
```

```
    INGESTED.inc()
```

```
    buf = get_buffer(msg.device_id)
```

```
    buf.append(x_vec)
```

```

if len(buf) < MIN_SAMPLES:
    # ще не накопичили вікно – діагностику не робимо
    return None

x_window = np.stack(list(buf), axis=0) # (T, F)

# Інференс/діагностика
with DIAG_LAT.time():
    x_hat = model_reconstruct(x_window)
    err = reconstruction_error(x_window, x_hat)

is_anom = err > THRESHOLD

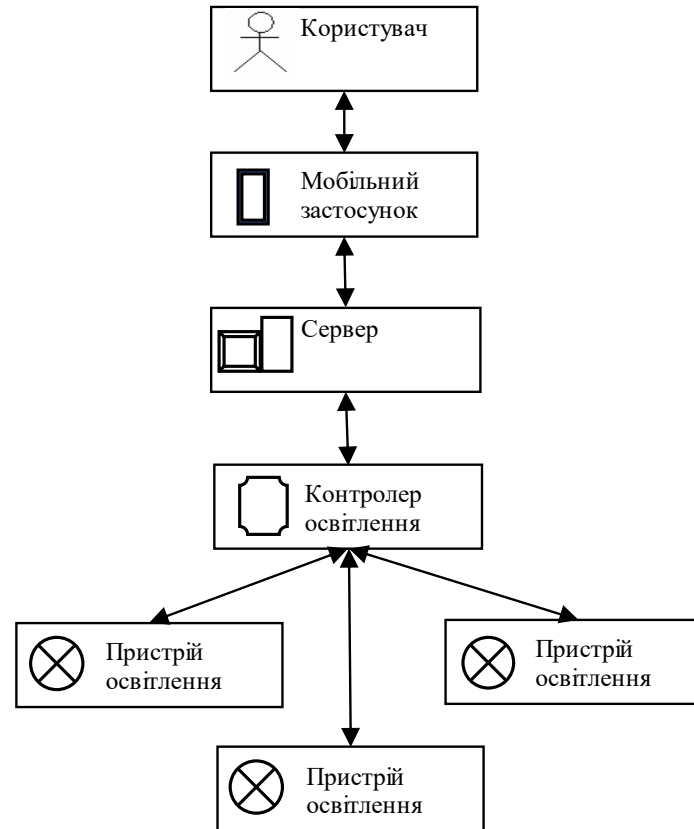
LAST_ERROR.set(err)
LAST_DECISION.set(1 if is_anom else 0)

if is_anom:
    ANOMALIES.inc()
    # Тут можна додати: запис у БД, webhook, email/telegram, Kafka/MQTT тощо.

return DiagnosticOut(
    device_id=msg.device_id,
    ts=msg.ts,
    recon_error=err,
    is_anomaly=is_anom,
    threshold=THRESHOLD
)

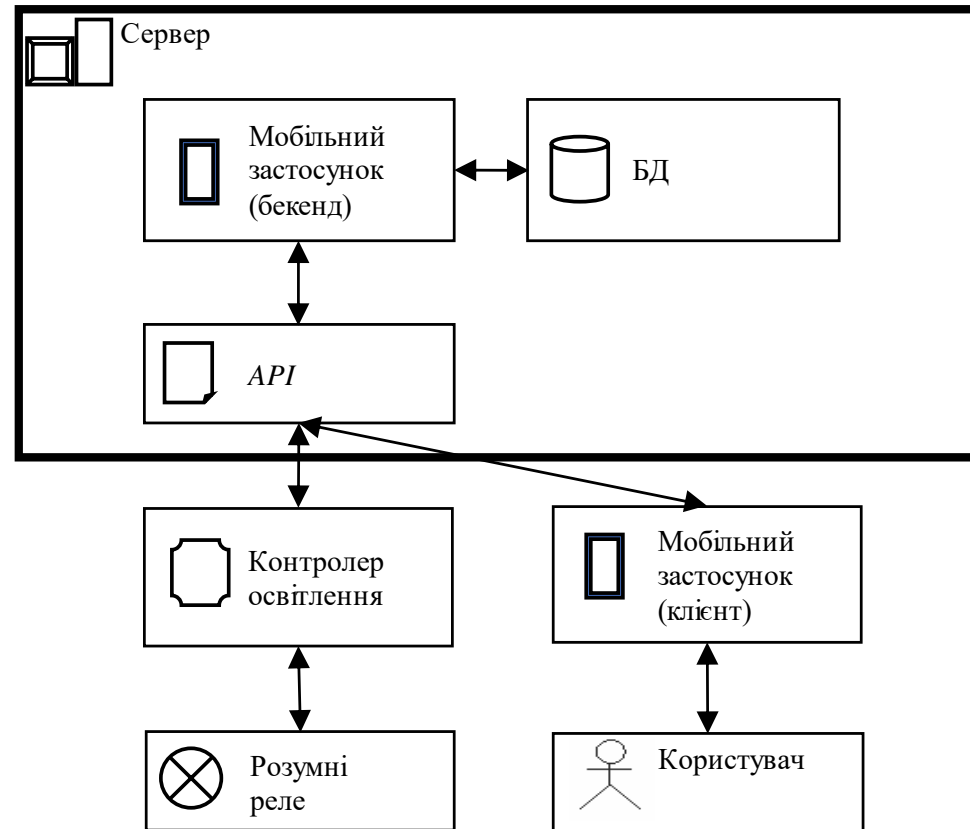
```

Компоненти системи дистанційного управління освітленням



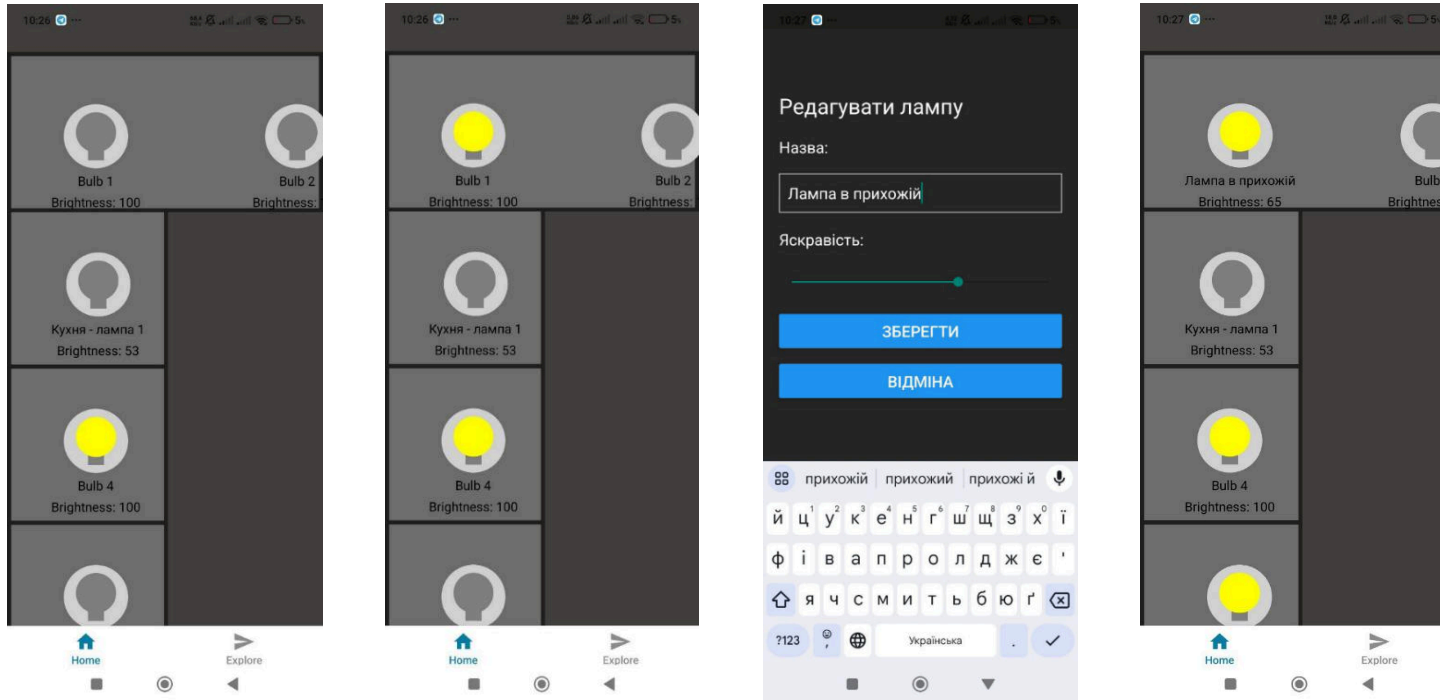
					<i>КАІ 25 02 90 001 ПЛ</i>				
					<i>Компоненти системи дистанційного управління освітленням</i>	<i>Літ.</i>		<i>Маса</i>	<i>Масштаб</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		<i>Д</i>			
<i>Виконав</i>		<i>Войцехівський Т.М.</i>							
<i>Керівник</i>		<i>Аленько Н.В.</i>							
<i>Рецензент</i>						<i>Лист</i>	<i>1</i>	<i>Листів</i>	<i>1</i>
<i>Н. Контр.</i>		<i>Тупота Е.В.</i>				<i>М-126-24-1-ІТ</i>			
<i>Зав. каф.</i>		<i>Нечипорук О.П.</i>							

Компоненти серверної частини системи управління пристроями освітлення

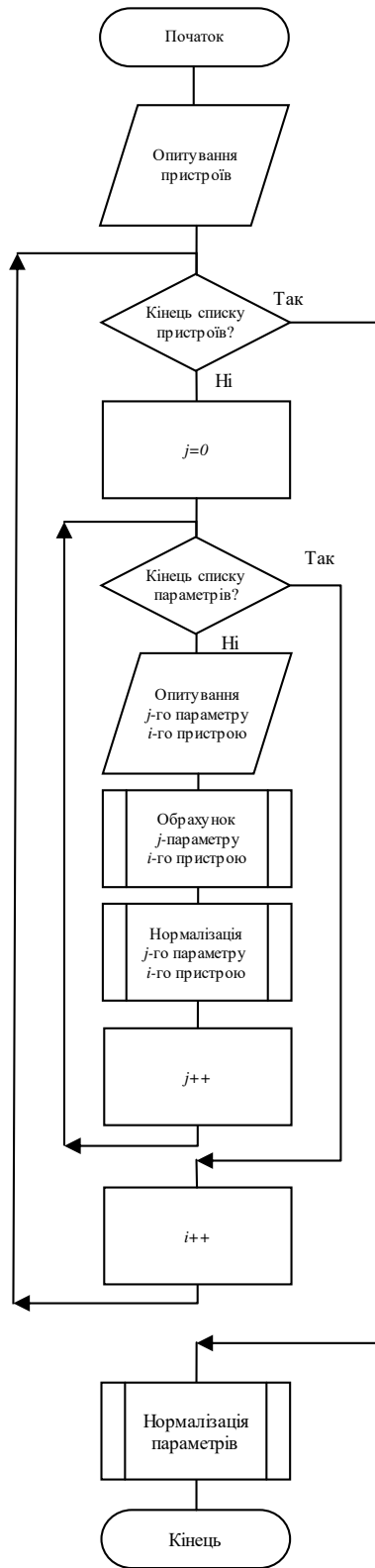


					<i>КАІ 25 02 90 002 ПЛ</i>				
					<i>Компоненти серверної частини системи управління пристроями освітлення</i>	<i>Літ.</i>		<i>Маса</i>	<i>Масштаб</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		<i>Д</i>			
<i>Виконав</i>		<i>Войцехівський Т.М.</i>							
<i>Керівник</i>		<i>Аленько Н.В.</i>							
						<i>Лист 1</i>		<i>Листів 1</i>	
<i>Рецензент</i>						<i>М-126-24-1-ІТ</i>			
<i>Н. Контр.</i>		<i>Тупота Е.В.</i>							
<i>Зав. каф.</i>		<i>Нечипорук О.П.</i>							

Вікна програмного модуля управління мережею IoT-пристроїв



					<i>КАІ 25 02 90 003 ПЛ</i>				
					<i>Вікна програмного модуля управління мережею IoT-пристроїв</i>	<i>Лім.</i>		<i>Маса</i>	<i>Масштаб</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		<i>Д</i>			
<i>Виконав</i>		<i>Войцехівський Т.М.</i>							
<i>Керівник</i>		<i>Аленько Н.В.</i>							
						<i>Лист</i>	<i>1</i>	<i>Листів</i>	<i>1</i>
<i>Рецензент</i>						<i>М-126-24-1-ІТ</i>			
<i>Н. Контр.</i>		<i>Тупота Е.В.</i>							
<i>Зав. каф.</i>		<i>Нечипорук О.П.</i>							



Зм.	Лист	№ документа	Підпис	Дата
Виконав		Войцехівський Т.М.		
Керівник		Апенько Н.В.		
Консульт.				
Н. контроль		Тупота Е.В.		
Зав. каф.		Нечипорук О.П.		

КАІ 25 02 90 005 ПМ

Діагностування роботи мережі
IoT-пристроїв
(схема алгоритму)

Літера	Маса	Масштаб
Лист 1	Листів 1	

M-126-24-1-IT