ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методики прогнозування збоїв в розумному будинку на основі методів машинного навчання»

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

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

_____ Юрій БАЗАК
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Юрій БАЗАК
група ІСДМ-62

Керівник: Вікторія ЖЕБКА
науковий ступінь, вчене звання д.т.н., професор

Рецензент:
науковий ступінь, вчене звання _____
Ім'я, ПРІЗВИЩЕ

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІПЗАС

_____ Каміла СТОРЧАК
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Базакові Юрієві Костянтиновичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка методики прогнозування збоїв в розумному будинку на основі методів машинного навчання

керівник кваліфікаційної роботи Вікторія ЖЕБКА д.т.н., професор,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, параметри розумного будинку, метрики, методи машинного навчання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження систем управління розумним будинком
Аналіз технологій машинного навчання та можливості застосування в розумному будинку
Вибір методу машинного навчання

Розробка платформи прогнозування збоїв та відповідного програмного забезпечення

5. Перелік графічного матеріалу: *презентація*
    1. Технології та системи управління розумним будинком
    2. Огляд збоїв в розумному будинку
    3. Аналіз методів машинного навчання
    4. Платформа прогнозування збоїв
    5. Програмне забезпечення для прогнозування збоїв

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення матеріалів для аналіз систем управління розумним будинком	05.11-12.11.23	
3	Дослідження збоїв в розумному будинку	13.11-19.11.23	
4	Дослідження технологій машинного навчання. Застосування машинного навчання до прогнозування збоїв в розумному будинку	20.11-25.11.23	
5	Побудова платформи прогнозування збоїв та відповідного програмного забезпечення.	27.11-03.12.23	
6	Дослідження ефективності побудованої системи	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти _____ Юрій БАЗАК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

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

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 75 стор., 14 табл., 14 рис., 30 джерел.

Мета магістерської роботи – розробка ефективної системи прогнозування можливих несправностей та збоїв в інтелектуальних системах розумного будинку.

Об'єкт дослідження – процес прогнозування збоїв в розумному будинку.

Предметом дослідження – методи машинного навчання, які застосовуються для аналізу даних, виявлення патернів та взаємозв'язків між різними системами розумного будинку з метою передбачення можливих збоїв та несправностей.

Короткий зміст роботи: Робота включає в себе комплексний аналіз сучасних технологій та компонентів, що застосовуються в розумних будинках. Починаючи з огляду технологій, датчиків та систем управління цією інноваційною сферою, робота прогнозує можливі збої, які можуть виникнути в таких системах. Проведено аналіз методів машинного навчання, які використовуються для прогнозування цих збоїв. Це включає збір, обробку та аналіз статистичних даних, вибір та навчання моделей машинного навчання. Ключовий момент у роботі – розробка методики прогнозування збоїв в розумному будинку на основі методів машинного навчання. Це включає розробку програмного забезпечення, огляд платформи для прогнозування збоїв, інтеграцію цієї платформи в систему розумного будинку та дослідження ефективності впровадження блоку машинного навчання в систему.

КЛЮЧОВІ СЛОВА: РОЗУМНИЙ БУДИНОК, ЗБОЇ, МАШИННЕ НАВЧАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ABSTRACT

The text part of the qualification work for obtaining a master's degree: 75 pages, 14 table, 14 figures, 30 sources.

The goal of the master's thesis is to develop an effective system for predicting possible malfunctions and failures in the intelligent systems of a smart home.

The object of research is the process of predicting failures in a smart home.

The subject of the research is machine learning methods that are used to analyze data, identify patterns and relationships between various smart home systems in order to predict possible failures and malfunctions.

Summary of the work: The work includes a comprehensive analysis of modern technologies and components used in smart homes. Starting with an overview of the technologies, sensors and control systems of this innovative field, the paper predicts possible failures that may occur in such systems. The machine learning methods used to predict these failures are analyzed. This includes collecting, processing and analyzing statistical data, selecting and training machine learning models. The key point in the work is the development of a methodology for predicting failures in a smart home based on machine learning methods. This includes software development, review of a platform for failure prediction, integration of this platform into a smart home system, and research into the effectiveness of implementing a machine learning unit into the system.

KEY WORDS: SMART HOME, FAILURES, MACHINE LEARNING, SOFTWARE.

ЗМІСТ

ВСТУП

Актуальність роботи. Розумні будинки стають все більш поширеними завдяки розвитку Інтернету речей (IoT) і розумних технологій. Вони забезпечують автоматизацію та зручність управління різними системами, такими як освітлення, опалення, безпека, енергоефективність та багато інших.

Проте із зростанням комплексності цих систем збільшується ймовірність виникнення збоїв або проблем. Нестабільність мережі, програмні помилки, несправні пристрої – усе це може призвести до непередбачуваних ситуацій, які вплинуть на зручність користування та безпеку розумного будинку.

Тому розробка методики прогнозування збоїв у розумному будинку за допомогою методів машинного навчання є надзвичайно важливою. Використання алгоритмів машинного навчання дозволяє аналізувати великі обсяги даних, виявляти відхилення та патерни, що передують збоям. Такий підхід дозволяє передбачати можливі проблеми та вживати заходів на їх запобігання ще до того, як вони виникнуть.

Мета магістерської роботи полягає в розробці ефективної системи прогнозування можливих несправностей та збоїв в інтелектуальних системах розумного будинку.

Завдання дослідження:

1. Огляд існуючих рішень.
2. Аналіз можливих збоїв в розумному будинку.
3. Дослідження методів машинного навчання.
4. Створення методики прогнозування збоїв.
5. Оцінка ефективності використання машинного навчання.

Об'єктом дослідження є процес прогнозування збоїв в розумному будинку.

Предметом дослідження є методи машинного навчання, які застосовуються для аналізу даних, виявлення патернів та взаємозв'язків між різними системами розумного будинку з метою передбачення можливих збоїв та несправностей.

Методи дослідження: методи машинного навчання, оптимізаційні методи, методи математичної статистики.

Наукова новизна магістерської роботи полягає в розробка методики прогнозування збоїв у розумному будинку за допомогою методів машинного навчання, що відрізняється від існуючих введенням блоку прогнозування збоїв та попередження виникнення надзвичайних ситуацій а розумному будинку.

Практична значущість магістерської роботи полягає у розробці програмного забезпечення на основі запропонованої методики прогнозування збоїв і впровадження його в процес управління розумним будинком.

Апробація: магістерська робота пройшла апробацію на Науково-практичній конференція «Проблеми комп'ютерної інженерії». Результати дослідження опубліковані в наступних працях:

1. Базак Ю.К. «Порівняння методів машинного навчання для прогнозування збоїв в розумному будинку». Тези доповіді на Науково-практична конференція «Проблеми комп'ютерної інженерії». Збірник тез. – К.: ДУІКТ, 2023. с. 125-127
2. Жебка В.В., Базак Ю.К., Сторчак К.П. Особливості прогнозування збоїв в розумному будинку на основі методів машинного навчання // Телекомунікаційні та інформаційні технології. 2023. № 4 (81), с. 4-12.

Теоретична, методична та практична значущість отриманих результатів: результати магістерської роботи можуть бути використані в процесі управління розумним будинком.

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатку.

1 АНАЛІЗ ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ В РОЗУМНОМУ БУДИНКУ

1.1 Огляд технологій, датчиків та систем управління розумним будинком

Розумний будинок як цілісна система – включає в себе різноманітні пристрої, сенсори, мережі зв'язку та програмне забезпечення для автоматизації та управління побутовими функціями.

Рис. 1.1. Схема розумного будинку

Розумний будинок – це екосистема, де використовуються різноманітні технології для автоматизації та оптимізації побутових процесів. Технології, що застосовуються в такому будинку, можна розділити на кілька основних категорій (табл. 1.1):

1. Системи автоматизації – забезпечують управління освітленням, опаленням, кондиціюванням повітря, водопостачанням та іншими побутовими системами. Такі системи можуть бути інтегровані з голосовими помічниками

(наприклад, Amazon Alexa, Google Assistant) для управління за допомогою голосових команд.

2. Смарт-пристрої – це пристрої, які можуть бути підключені до мережі Інтернет речей (IoT). Сюди входять смарт-термостати, смарт-замки, відеодомофони, водяні лічильники, розумні кухонні прилади тощо.

3. Системи безпеки – включають в себе відеоспостереження, системи виявлення витоків води або газу, сигналізацію та інші пристрої для захисту будинку та його мешканців.

4. Енергоефективність – використання сонячних батарей, систем управління енергоспоживанням, розумні лічильники електроенергії, які дозволяють ефективно використовувати енергію та зменшувати витрати.

Таблиця 1.1

Призначення технологій в розумному будинку

Категорія технології	Приклади технологій	Призначення
Системи автоматизації	Смарт-термостати, системи освітлення, голосові помічники	Забезпечення комфорту, ефективного управління побутовими системами
Смарт-пристрої	Смарт-замки, відеодомофони, смарт-прилади для кухні	Підвищення безпеки та зручності використання
Системи безпеки	Відеоспостереження, сигналізація, системи виявлення витоків	Захист будинку та мешканців від небезпеки
Енергоефективність	Сонячні батареї, системи управління енергоспоживанням, розумні лічильники	Оптимізація споживання енергії та зменшення витрат

Ці технології спільно створюють інтелектуальне середовище у будинку, що дозволяє оптимізувати різні аспекти життя мешканців, збільшуючи комфорт, безпеку та енергоефективність.

Датчики грають ключову роль у розумних будинках, забезпечуючи збір даних про оточуюче середовище та стан певних систем. Вони дозволяють моніторити різні параметри та управляти різними аспектами життя в будинку.

Таблиця 1.2

Призначення датчиків в розумному будинку

Тип датчика	**Приклади**	**Призначення**
Температурний	Термодатчики	Контроль та регулювання температури у приміщенні
Вологості	Гігродатчики	Вимірювання вологості повітря для комфортних умов
Руху	Інфрачервоні датчики	Активація систем безпеки та автоматизація освітлення
Відкриття/Закриття	Магнітні датчики	Контроль доступу та управління системами безпеки
Витоків	Датчики витоків	Виявлення потенційно небезпечних ситуацій
Освітлення	Фоторезистори	Автоматичне регулювання освітлення для енергоефективності

Типи датчиків та їх функціональність можна представити наступним чином (табл. 1.2):

1. Датчики температури та вологості – вимірюють температуру та вологість в приміщенні, що дозволяє автоматично регулювати системи опалення та кондиціонування повітря для створення комфортних умов.

2. Датчики руху – виявляють рух в певних зонах будинку, що використовується для активації систем безпеки, освітлення або автоматичного управління.

3. Датчики відкриття/закриття – виявляють відкриття або закриття дверей, вікон та інших об'єктів, що дозволяє контролювати доступ та вмикачі у розумному будинку.

4. Датчики витоків – виявляють витоки води або газу, сприяючи вчасному виявленню потенційно небезпечних ситуацій.

5. Датчики освітлення – вимірюють рівень освітленості, дозволяючи автоматично регулювати освітлення для енергоефективності та комфорту.

Ці різні типи датчиків разом створюють систему моніторингу та управління, яка дозволяє розумному будинку адаптуватися до потреб мешканців та оптимізувати споживання ресурсів.

Системи управління в розумному будинку грають ключову роль у забезпеченні автоматизації та контролю за різними аспектами життя в будинку. Ці системи координують роботу пристроїв та забезпечують їх взаємодію для створення зручного та ефективного середовища для мешканців. Огляд деяких типів систем управління представлений в таблиці 1.3:

1. Централізовані системи управління – об'єднують різні пристрої та підсистеми (освітлення, опалення, безпека тощо) у єдину платформу для централізованого керування. Вони можуть бути представлені в різних форматах, від програмного забезпечення до спеціальних пультів керування.

2. Мобільні додатки – додатки для смартфонів або планшетів, які дозволяють користувачам віддалено керувати різними системами у будинку, навіть коли вони поза його межами. Це може включати управління освітленням, температурою, відеоспостереженням тощо.

3. Голосові помічники – інтеграція з голосовими помічниками, такими як Amazon Alexa, Google Assistant або Apple HomeKit, що дозволяє користувачам керувати пристроями та системами в будинку за допомогою голосових команд.

4. Інтелектуальні сценарії та автоматизація – створення програмованих сценаріїв для виконання певних дій або автоматизації процесів у будинку. Наприклад, автоматичне вмикання освітлення під час заходу сонця або зміна температури при певних умовах.

Таблиця 1.3

Системи управління розумним будинком та їх функціональність

Тип системи управління	Приклади	Функції
Централізовані системи	Control4, Crestron	Централізоване керування всіма системами будинку
Мобільні додатки	Apple Home, Samsung SmartThings	Віддалене керування через смартфон або планшет
Голосові помічники	Amazon Alexa, Google Assistant	Голосове керування системами за допомогою команд
Автоматизація	SmartThings, IFTTT	Програмовані сценарії та автоматичні дії для оптимізації роботи

Ці різні системи управління працюють разом, щоб надати користувачам широкі можливості контролю та управління різними аспектами їх розумного будинку.

Технології, датчики та системи управління взаємодіють у розумному будинку, створюючи систему, яка реагує на оточуюче середовище та виконує певні дії відповідно до заданих параметрів та умов. Способи їх взаємодії можна описати наступним чином:

1. Датчики та збирання даних – розміщені по всьому будинку, вони збирають дані про температуру, вологість, рух, відкриття/закриття, освітлення та інші параметри. Ці дані потім передаються до центральної системи збору даних.

2. Аналіз даних та прийняття рішень – центральна система аналізує отримані дані в реальному часі або за певними програмованими алгоритмами. Вона визначає відповідь на оточуюче середовище на підставі цих даних.

3. Прийняття рішень та виконання дій – на основі аналізу даних система управління приймає рішення про потрібні зміни в будинку. Наприклад, вона може вмикати/вимикати світло, регулювати температуру, відкривати/закривати двері або виконувати інші дії.

4. Взаємодія між системами – різні системи управління, такі як мобільні додатки або централізовані системи, взаємодіють з центральною системою збору даних для відображення інформації користувачам та передачі команд до підключених пристроїв у будинку.

5. Зміна параметрів залежно від умов – на основі отриманих даних та встановлених алгоритмів системи управління автоматично або за командою користувача змінюють параметри систем у будинку для оптимізації комфорту, безпеки та енергоефективності.

Отже, взаємодія між датчиками, технологіями та системами управління створює інтелектуальне середовище, яке реагує на зміни та вимоги користувачів для оптимального функціонування розумного будинку.

Таблиця 1.4

Алгоритм виконання дій в системі розумного будинку

Етап	**Дії**	**Приклади**
1.	Збір даних датчиками	Температура, вологість, рух
2.	Передача даних до системи збору	Центральна система збору даних
3.	Аналіз даних та виявлення патернів	Виявлення збоїв, аналіз параметрів
4.	Прийняття рішень системою управління	Регулювання температури, вмикання світла
5.	Виконання дій	Зміни параметрів, керування пристроями
6.	Взаємодія з користувачем	Мобільні додатки, голосові команди

В таблиці 1.4 представлена послідовність дій у взаємодії між датчиками, системою збору даних, аналізом та управлінням, що веде до прийняття рішень і виконання певних дій у розумному будинку.

В розумному будинку основні параметри мають відповідати значенням представлених в таблиці 1.5.

Таблиця 1.5

Числові значення основних параметрів розумного будинку

Показник	Межі/Числове вираження
Електропостачання	Напруга: 220V або 110V, Частота: 50Hz або 60Hz
Системи безпеки	Рівень сигналізації: 0 до 100. Активність відеоспостереження: вкл./викл. Статус датчиків: відкрито/закрито. Статус системи: в роботі/помилка.
Водопостачання	Тиск: від 0 до макс. значення. Витрата води: у літрах за хвилину. Рівень води у баках: від 0 до макс. обсягу.
Опалення та кондиціонування	Температура: від 15°C до 30°C. Вологість: від 0% до 100%. Режим роботи систем: увімкнено/вимкнено.
Мережа Wi-Fi	Швидкість передачі даних: в Мбіт/с,. Рівень сигналу: від 0 до 100. Кількість підключених пристроїв.
Моніторинг стану систем	Статус пристроїв: в роботі/непрацюючі. Кількість датчиків, які передають дані: кількість.

Взаємодія технологій, датчиків та систем управління у розумному будинку – це як система, де кожен елемент виконує свою частину для створення оптимального середовища. Датчики збирають дані про навколишнє середовище, від температури до руху, а системи збору і аналізу даних перетворюють цю інформацію на корисні знання. Системи управління реагують на ці дані,

приймаючи рішення щодо оптимізації енергоспоживання, забезпечення безпеки та комфорту для мешканців. Цей складний механізм дозволяє розумному будинку адаптуватися до змінних потреб і пріоритетів, забезпечуючи максимальну ефективність і зручність. Ця взаємодія створює зручний доступ до управління будинком з будь-якого місця, роблячи життя більш гнучким та насиченим. Покращена безпека, енергоефективність та підвищений комфорт стають не просто перевагами, а фундаментальними складовими нового стандарту сучасного життя.

Зростаюча популярність IoT та розумних технологій призводить до збільшення складності систем, що вимагає ефективних та передбачуваних засобів управління та контролю. Прогнозування можливих збоїв на підставі аналізу великих обсягів даних стає критично важливим для забезпечення безпеки, ефективності та стабільності роботи систем у розумних будинках. Особливо це важливо у зв'язку з підвищеною складністю інтегрованих систем, які можуть потребувати швидкого та точного реагування на можливі неполадки для забезпечення комфорту та безпеки користувачів.

1.2 Аналіз можливих збоїв в розумному будинку

Збій в розумному будинку – це ситуація, коли один або кілька пристроїв чи систем, які управляють певними функціями або аспектами будинку (такі як освітлення, опалення, системи безпеки, тощо), перестають працювати або працюють неправильно.

Це може виникнути через технічні несправності, проблеми з мережею, програмні збої, вплив зовнішніх факторів або неправильні налаштування систем. Збій може призвести до втрати контролю над певними аспектами будинку, зниження безпеки, втрати зручності управління системами або навіть до загрози для функціонування будинку.

Рівень збою може бути різним: від дрібних проблем, які можна швидко виправити, до серйозних ситуацій, які потребують професійної інтервенції або ремонту. Однак, важливо вчасно виявляти та виправляти збої, щоб забезпечити безпеку та нормальну роботу всіх систем у розумному будинку.

Збої в розумному будинку можуть відбуватися з різних причин. Ось деякі з можливих сценаріїв:

1. Технічні помилки або несправності. Подібно до будь-яких електронних систем, у розумному будинку можуть виникати технічні помилки через несправності пристроїв, датчиків чи систем управління. Це може призвести до неправильної роботи системи опалення, освітлення або безпеки.

2. Проблеми з мережею чи зв'язком. Якщо будь-який з пристроїв або систем втрачає з'єднання з мережею Wi-Fi або між собою, це може спричинити втрату функціональності відповідного пристрою або навіть весь розумний будинок.

3. Програмні збої. Помилки в програмному забезпеченні системи управління або додатків для мобільних пристроїв також можуть викликати неправильну роботу системи чи навіть призвести до втрати зв'язку з деякими пристроями.

4. Вплив зовнішніх чинників. Екстернальні фактори, такі як кібератаки або перешкоди в мережі, також можуть призвести до збоїв в роботі розумного будинку.

5. Неправильні налаштування або людські помилки. Неправильні налаштування системи, невірне використання функцій або помилки користувача також можуть призвести до проблем в роботі систем.

Хоча розумні будинки досить надійні, збої все ж можуть виникати. Однак, зазвичай більшість проблем можна вирішити шляхом технічної підтримки, перезавантаження системи чи перевірки налаштувань пристроїв. Чим складніша система, тим більше потрібно уваги при виборі, налаштуванні та підтримці, але з правильними заходами можна значно знизити ризик виникнення збоїв.

Типи збоїв, які можуть бути в розумному будинку з можливими причинами їх виникнення представлено в таблиці 1.5.

Таблиця 1.5 допомагає ідентифікувати та аналізувати можливі причини збоїв у розумному будинку, що можуть виникнути з різних джерел: технічних несправностей, проблем з мережею, програмних помилок, впливу зовнішніх факторів та неправильних налаштувань.

Таблиця 1.5

Типи збоїв у розумному будинку

Тип збою	Опис	Причини
Технічні несправності	Неправильна робота пристроїв, систем	Несправність пристроїв, датчиків, систем управління
Проблеми з мережею	Втрата зв'язку з мережею, Wi-Fi	Втрата зв'язку, перевантаження мережі, несправність обладнання
Програмні збої	Помилки в програмному забезпеченні	Баги, неправильні оновлення, програмні помилки
Зовнішні впливи	Кібератаки, втручання в роботу системи	Кіберзагрози, перешкоди в мережі, вплив зовнішніх факторів
Неправильні налаштування	Невірне використання або налаштування	Неправильне програмування, людські помилки

Таблиця 1.6 показує можливі моменти виникнення збоїв у розумному будинку та шляхи їх можливого вирішення: від технічної підтримки до перевірки налаштувань чи застосування заходів безпеки для захисту від зовнішніх впливів.

Таблиця 1.6

Виникнення збоїв та можливі шляхи їх усунення

Тип збою	Момент виникнення	Можливі шляхи ліквідації
Технічні несправності	Під час роботи пристроїв або систем	Технічна підтримка, заміна несправних пристроїв
Проблеми з мережею	При втраті зв'язку з мережею	Перевірка мережевого з'єднання, перезавантаження маршрутизатора
Програмні збої	Під час використання програмного забезпечення	Оновлення програмного забезпечення, технічна підтримка
Зовнішні впливи	При злому або кібератаках	Захист від кібератак, мережеві заходи безпеки
Неправильні налаштування	Під час конфігурації або використання	Перевірка налаштувань, навчання користувачів правильному використанню

Збої в розумному будинку можуть мати різний рівень ушкодження системи в залежності від їхньої природи та впливу на функціонування системи. Оцінка ризику може бути такою:

1. Технічні несправності – можуть призвести до часткового або повного відмову пристроїв або систем, але за умови належного обслуговування швидко можуть бути виправлені без серйозних наслідків.

2. Проблеми з мережею – втрата зв'язку може призвести до втрати контролю над певними пристроями чи системами управління. Однак, це може бути виправлено перевіркою мережі та перезавантаженням маршрутизатора без значного ушкодження системи.

3. Програмні збої – помилки в програмному забезпеченні можуть призвести до неправильної роботи систем, але зазвичай вони можуть бути виправлені оновленням програмного забезпечення без серйозних наслідків.

4. Зовнішні впливи – кібератаки чи вплив зовнішніх факторів можуть мати серйозний вплив на безпеку та функціонування систем. Їх наслідки можуть бути значними, включаючи порушення безпеки, втрату даних чи навіть збій систем.

5. Неправильні налаштування – неправильні налаштування можуть призвести до неналежної роботи системи, але, якщо вчасно виявлені, вони можуть бути виправлені без серйозних наслідків.

Кожен з цих ризиків може викликати проблеми з роботою розумного будинку. Однак, важливо вчасно виявляти та виправляти збої, щоб уникнути серйозних ушкоджень системи.

Відповідно до проведеного аналізу рівня пошкодження мережі при настанні тієї чи іншої ризикової ситуації було встановлено наступний рейтинг типів збою (табл.1.7). Чим вище значення, тим більш можливий вплив збою на роботу мережі та функціонування систем. Як видно з таблиці найвищий рівень ушкодження мережі настає в результаті зовнішніх впливів.

Таблиця 1.7

Рівень ушкодження мережі в результаті збою

Тип збою	**Числове значення рівня ушкодження мережі**
Технічні несправності	3
Проблеми з мережею	4
Програмні збої	2
Зовнішні впливи	5
Неправильні налаштування	1

Збої в розумному будинку – це перешкоди у функціонуванні пристроїв та систем, що можуть виникнути з різних причин, включаючи технічні несправності, проблеми з мережею, програмні помилки або зовнішні впливи. Ці збої можуть призвести до втрати зручності, порушення безпеки чи навіть недоступності певних функцій будинку. Важливо активно виявляти та усувати збої, забезпечуючи стабільну та надійну роботу розумного будинку.

Моніторинг, регулярне обслуговування та швидка реакція на виявлені проблеми є ключовими для забезпечення ефективного функціонування розумного будинку. Шляхи ліквідації збоїв можуть варіюватися від перевірки налаштувань до технічної підтримки або заміни пристроїв. Ретельний нагляд і вчасні заходи можуть покращити надійність та продуктивність розумного будинку, забезпечуючи комфорт та безпеку його мешканців.

Використання методів машинного навчання у розумному будинку може бути дуже корисним для зменшення збоїв та покращення його ефективності. Машинне навчання може допомогти в розпізнаванні патернів, прогнозуванні відмов, аналізі поведінки користувачів та оптимізації систем управління.

Ці методи дозволяють системам розумного будинку навчитися відповідати на зміни в середовищі, передбачати можливі проблеми та адаптуватися до потреб користувачів. Так, блок на основі машинного навчання може попереджати про будь-які відхилення від визначених патернів, що може свідчити про проблеми.

Проте важливо враховувати, що жодна система не є повністю беззбіжною. Машинне навчання також може мати свої обмеження і не завжди точно передбачати всі можливі ситуації. Тому використання цих методів повинно супроводжуватися правильним плануванням, тестуванням та резервними планами у разі виникнення непередбачених обставин.

Все ж, відповідно налаштовані системи машинного навчання можуть великою мірою знизити ризик збоїв та покращити загальну надійність розумного будинку, забезпечуючи більше зручності та безпеки для його мешканців.

1.3 Дослідження методів машинного навчання та їх використання в розумному будинку

На сьогоднішній день прогнозування збоїв в розумному будинку в основному базується на реактивному аналізі даних. Це означає, що системи виявляють

аномальні ситуації або збої вже після їх виникнення, що може ускладнювати уникнення можливих проблем.

Однак, з використанням методів машинного навчання, зокрема алгоритмів класифікації, кластеризації та передбачення, можна розвивати системи, які будуть здатні прогнозувати збої в розумному будинку заздалегідь.

Такі системи використовують аналіз даних з сенсорів, пристроїв IoT, систем управління, споживання енергії та інші дані, щоб виявляти патерни та аномалії, які можуть передувати збоям. На основі цих відомостей системи машинного навчання можуть побудувати моделі передбачення, які реагують на певні сигнали чи зміни у звичайному функціонуванні, попереджуючи про можливі проблеми або вживаючи заходів для їх запобігання.

Цей напрямок досліджень ще розвивається, але він обіцяє покращення систем управління розумним будинком шляхом надання їм здатності передбачати та запобігати можливим збоям заздалегідь, забезпечуючи більш високу надійність та безпеку для користувачів.

Математичні моделі, що використовуються в методах машинного навчання для прогнозування збоїв, можуть варіюватися залежно від вибраного алгоритму. Загальний вигляд формули для задачі прогнозування на основі, наприклад, лінійної регресії має вигляд:

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n+\varepsilon, \quad (1.1)$$

де y – цільова змінна, яку необхідно передбачити (наприклад, можливий збій у розумному будинку);

$x_1,x_2,\ldots,x_n$ – ознаки або фактори, які використовуються для прогнозування (це можуть бути дані з сенсорів, історічні дані про роботу систем тощо);

$\beta_0,\beta_1,\beta_2,\ldots,\beta_n$ – коефіцієнти моделі, які визначають вагу кожної ознаки у прогнозуванні;

ε – помилка моделі, що представляє випадкову складову, яка не була врахована в моделі.

Формула (1.1) представляє лінійну регресійну модель, яка може бути однією з математичних моделей, застосовуваних у методах машинного навчання для прогнозування збоїв.

Для прогнозування збоїв в розумному будинку можна використовувати різні алгоритми машинного навчання, залежно від характеристик даних та специфіки системи:

1. Нейронні мережі (Neural Networks) – глибокі нейронні мережі, зокрема рекурентні (RNN) або здатні адаптуватися до динаміки системи та виявлення складних залежностей між даними.
2. Випадковий ліс (Random Forest) – цей метод добре працює з великою кількістю вхідних характеристик і може бути ефективним для виявлення аномалій та патернів у даних розумного будинку.
3. Метод опорних векторів (Support Vector Machines, SVM) – використовується для виявлення взаємозв'язків між даними та може бути ефективним у випадках, коли потрібно розділити різні класи даних.
4. Глибоке навчання зі згортковими нейронними мережами (Convolutional Neural Networks, CNN) – використовується для обробки зображень або сигналів, що може бути корисним для аналізу даних з сенсорів у розумному будинку.
5. Методи кластеризації, наприклад, k-середніх (k-means) – можуть допомогти виявити групи або патерни у даних, що можуть свідчити про можливі аномалії.

Визначення "найвищої точності" у контексті прогнозування збоїв в розумному будинку залежить від кількох чинників, таких як характеристики даних, обсяг навчального набору, рівень розуміння системи, зміни у середовищі тощо. Отже, конкретний алгоритм, який надає найвищу точність, може варіюватися в кожному випадку.

Зазвичай, нейронні мережі, зокрема глибокі нейронні мережі, мають потенціал надавати високу точність через їхню здатність виявляти складні залежності в даних. Однак, вони можуть вимагати значних обчислювальних ресурсів для навчання та оптимізації.

Таблиця 1.8

Порівняльна характеристика методів машинного навчання

Хактеристики алгоритмів	**Нейронні мережі**	**Випадковий ліс**	**SVM**	**CNN**	**k-середніх**
Ефективність передбачення збоїв	Висока	Висока	Висока	Висока	Середня
Складність моделі та інтерпретованість	Висока складність, низька інтерпретованість	Середня складність, середня інтерпретованість	Середня складність, висока інтерпретованість	Висока складність, низька інтерпретованість	Низька складність, висока інтерпретованість
Обчислювальна складність	Висока	Середня	Середня	Висока	Низька
Стійкість до перенавчання	Залежить від параметрів, може бути проблемою	Висока	Висока	Висока	Низька
Здатність до роботи з різноманітними типами даних	Добре працюють з різними типами даних	Добре пристосовані для різноманітних даних	Працюють добре з числовими даними	Переважно зображення або сигнали	Використовуються для кластеризації числових даних
Масштабованість	Залежить від обсягу даних та архітектури мережі	Добре масштабуються на великі обсяги даних	Підходять для невеликих обсягів даних	Залежить від розмірів датасету та архітектури	Залежить від кількості кластерів та об'єму даних

Випадковий ліс також може бути дуже ефективним завдяки своїй здатності працювати з великими обсягами даних і виявляти аномалії без великої потреби в налаштуваннях.

Також важливо враховувати, що не завжди найбільш складний чи популярний алгоритм забезпечить найкращі результати. Іноді простіші алгоритми можуть працювати краще, особливо якщо дані обмежені або якщо модель потребує швидкої реакції на зміни в середовищі розумного будинку.

Порівняння методів машинного навчання наведено в таблиці 1.8 та представлено на рисунку 1.2.

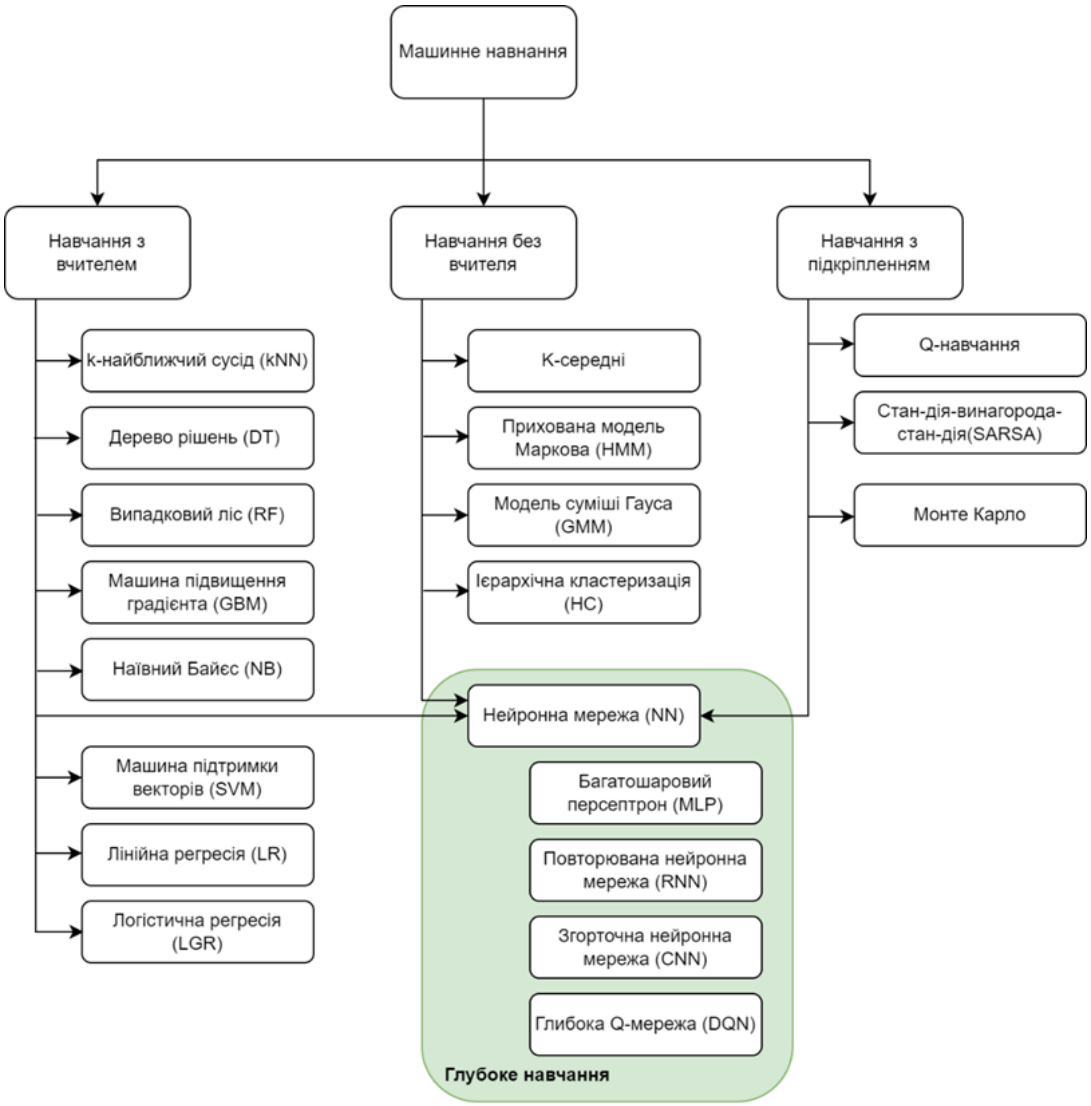

Рис. 1.2 Методи машинного навчання

Застосування методів машинного навчання може допомогти у зменшенні впливу деяких збоїв в розумному будинку. Можливі способи їх застосування представлено в таблиці 1.9.

Таблиця 1.9

Способи застосування методів машинного навчання до типів збоїв

Тип збою	**Методи машинного навчання**	**Вплив**
Технічні несправності	Прогнозування відмов	Попередження перед відмовою, можливість планування обслуговування та ремонту
Проблеми з мережею	Виявлення аномалій в мережі	Попередження перед втратою зв'язку, швидше виявлення та вирішення проблеми
Програмні збої	Детектування аномалій у програмному забезпеченні	Швидше виявлення та виправлення програмних помилок
Зовнішні впливи	Виявлення атак та незвичайної активності в мережі	Вчасне реагування на кіберзагрози, попередження від зловмисних атак
Неправильні налаштування	Автоматизоване коригування налаштувань	Зменшення ймовірності людських помилок, оптимізація налаштувань

Ці методи можуть допомогти виявити та уникнути деяких збоїв перед їх виникненням або швидше відновити роботу системи після збою. Вони можуть забезпечити раннє виявлення аномалій у роботі, що дозволить попередити виникнення проблем або швидко реагувати на них, що в свою чергу допоможе у зменшенні впливу цих збоїв на роботу розумного будинку.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЗБОЇВ ТА АНАЛІЗ НЕОБХІДНИХ ДАНИХ

2.1 Збір та обробка статистичних даних

Проблемою реальних даних IoT є відносно рідкісні збої пристроїв, які можуть бути використані для навчання алгоритмів машинного навчання. У поєднанні з коротким інтервалом часу вимірювання даних, це призведе до величезної кількості даних з дуже невеликою кількістю помилок. Наприклад, враховуючи 1 секунду інтервалу протягом року, загальна кількість записів даних становитиме близько 31,5 мільйон. Протягом цього часу пристрій IoT може вийти з ладу або взагалі не вийти з ладу, якщо його не поставити під значний вплив. Ця величезна кількість даних навряд чи поміститься в пам’яті, і рішення, як наприклад онлайн-навчання, як правило, вимагає дуже багато часу для виконання. Тому було зроблено вибір для створення штучних даних, які мають характеристики реальних даних IoT.

Ці характеристики включають наступні складові:

1. Відносно велика кількість записів, що генеруються з інтервалом в 1 секунду. Однак, тривалість обмежена двома тижнями, що призводить до приблизно 1,2 мільйона записів. Вибірка дозволяє тестувати алгоритми у відповідний проміжок часу.
2. Невелика кількість помилок порівняно із загальними даними, що призводить до дисбалансу даних. Проте збої, було збільшено до сотень або кількох тисяч, щоб мати можливість тренувати алгоритми.
3. Перекриваються збої та невідмови. Інколи може передувати збою контекст даних, який не може привести до збою. Зазвичай це причина, чому використовуються алгоритми машинного навчання, інакше фіксовані пороги можна легко оцінити за допомогою простих методів.
4. Включено позначку часу, дані про навколишнє середовище, дані моніторингу та збої. Це включає міри з дискретними, але також міри з неперервними значеннями. На практиці датчики надають дані про навколишнє

середовище, а пристрої обмінюються даними про їх внутрішній стан. Автоматизована система або супроводжувач може реєструвати збої.

Однак під час процесу генерації також були включені деякі спрощення, які, загалом, не повинно впливати на порівняння алгоритмів:

1. Передбачається, що дані надходять з одного пристрою. Хоча середовища IoT складається з багатьох пристроїв, їх можна обробляти окремо.

2. Після виходу з ладу пристрою час, протягом якого пристрій не надсилає жодних даних, не включено.

3. У даних немає аномалій. Основна увага полягає в тому, щоб порівняти алгоритми звичайними даних. Крім того, аномалії можуть бути оброблені в компоненті попередньої обробки.

4. Хоча короткі інтервали вимірювань мають сенс для центрального процесору або пам'яті, для температури або вологості воно є перебільшеним і може тривати до кількох секунд або хвилин. Це призводить до даних з різними інтервалами та, як наслідок, відсутніх значень після інтеграція. Ця проблема вирішується шляхом генерації всіх значень з інтервалом в 1 секунду.

За винятком позначок часу і збоїв, існує 11 характеристик, розділених на 3 категорії: час, моніторинг і особливості середовища. Функції часу включають годину, день тижня та робочі дні, що є числом послідовних днів без збоїв. Зазвичай ці функції можна отримати з позначку часу на етапі попередньої обробки. Окрім години та дня тижня, також можна отримати додаткову інформацію з позначки часу, таку як хвилини, місяці або свята. Однак, цих даних достатньо, щоб ввести значущі помилки в конкретному випадку. Моніторинг функції надає інформацію про пристрій. Це використання центрального процесору, використання пам’яті, зайнятість сховища та температура центрального процесору. Особливості середовища представляють дані, отримані від датчиків. Усі значення генеруються як числові значення, щоб пізніше уникнути кодування. Сюди також входять будні, які починатися з нуля (понеділок) і закінчуватися шістьма (неділя). Заповненість сховища, а також процесор і використання пам'яті, відображаються у відсотках.

Температура центрального процесору, температура середовища та вологість є десятковими числами з цифрою після коми. Температури представлені в градусах Цельсія, а вологість – у відсотках. Кондиціонер і обігрівач можуть мати значення 0 означає вимкнено або 1 означає увімкнено. У разі збоїв 1 означає невдачу, а 0 — відсутність збою. Значення помилки в записі даних слід інтерпретувати як стан пристрою після надсилання всіх інших дій. Хоча дані є штучними, їх намагалися зробити максимально реальними. Це можна зробити на підставі статистичних даних, таких як середнє значення, стандартне відхилення, мінімум і максимум. Дані аналізуються за допомогою Python.

Дані були згенеровані за допомогою реалізації на мові Java, яка виводить файл csv. Генератор використовує набір попередньо визначених правил для кожної функції та застосовує певний ступінь випадковості. Кожна функція має початкове значення, яке відповідає моменту початку процесу генерації.

Потім значення змінюються відповідно до індивідуальних ймовірностей та інтервалів. Використання пам’яті при цьому змінюється рідше і залишається в діапазоні 80%. Використання сховища з часом збільшується повільними темпами та випадковими підйомами та падіннями. Температура процесора залишається в нормальному діапазоні з низькою швидкістю зміни. Відсутній особливий зв'язок між функціями моніторингу. Навпаки, особливості навколишнього середовища залежать одна від одної, здебільшого від значення температури. Температура залежить від часу доби, вранці вона починає підвищуватися, а вдень знижуватися. Крім того, є обмеження кондиціонером і обігрівачем. При 25 °C включається кондиціонер і залишається активним, поки температура не знизиться до 23 °C. Так само обігрівач включається при 15 °C і залишається активним, поки температура не підніметься до 17 °C. Вологість залежить тільки від значення температури і не має випадкових змін.

В результаті навчання було змодельовано збій в робочі дні з 8 до 16 години. Крім того, використання процесору і пам’яті має бути вище 92%, температура– понад 50,1 °C, а температура навколишнього середовища від 19 °C до 23,5 °C. Зауважте, що завдяки кондиціонеру температура ніколи не перевищує 25 °C.

Встановлення обмеження на 23,5 °C охоплює обидва випадки: несправності, при яких кондиціонер вимкнено, і несправності, при яких він увімкнений. Обігрівач завжди вимкнений. Вологість і зберігання не мають прямого впливу на збої. Немає причин для вибору цієї конкретної умови даних для збою, алгоритми повинні мати можливість виявляти шаблони незалежно від типу збою. Однак діапазони були обрані для отримання відповідної кількості невдач. Для порівняння вибраних алгоритмів машинного навчання було створено два різні набори даних. У першому випадку збої виникають у 70% випадків, коли виконується зазначена умова даних. У другому – 30% часу. Перший має показати продуктивність алгоритму, коли є достатня кількість відмов для навчання (приблизно 4000), другий повинен показати свою ефективність при невеликій кількості відмов (приблизно 500).

Перейдемо до вибору методу машинного навчання.

2.2 Вибір моделей машинного навчання, їх навчання та тестування

Порівняння алгоритмів машинного навчання виконується на чотирьох різних представленнях даних: оригінальному, збалансованому, нормалізованому та стандартизованому. Вихідні дані не змінюються порівняно з обраними за винятком видалення значень часових позначок. Балансування виконується шляхом недостатньої вибірки безвідмовних даних у навчальних даних. Введення даних без збоїв обираються випадковим чином і видаляються з набору даних, що призводить до однакової кількості збоїв і без збоїв. Недостатня вибірка може стерти важливу інформацію з даних, що призведе до погіршення роботи алгоритмів. Основна перевага балансування даних полягає в тому, що воно зменшує ресурси та час, необхідні для навчання алгоритмів. Нормалізація даних у цій магістерській роботі стосується масштабування значень ознак у діапазоні від нуля до одиниці. Масштабування виконується за допомогою (2.1) окремо для кожної ознаки шляхом знаходження її мінімального та максимального значень. Дані також стандартизовані шляхом розгляду кожної функції окремо. Значення ознаки віднімаються від середнього, а потім діляться на стандартне відхилення, як показано в (2.2). Це

призводить до значень, зосереджених навколо нуля з одиничною дисперсією. Додатковим етапом попередньої обробки, який виконується перед нормалізацією та стандартизацією даних, є перетворення часових ознак дня тижня та години. Щоб вказати, що різниця між годинами 23 і 0 є такою ж, як різниця між 22 і 23, значення перетворюються в циклічні представлення за допомогою перетворення Фур'є [6]. У процесі перетворення обчислюються значення синусів і косинусів для кожної функції, як показано в рівнянні (2.3). Отже, кожна ознака замінюється відповідною ознакою синуса та косинуса. Обчислення залежить від загальної кількості різних значень ознак *N*, яка становить 24 для годин і 7 для днів тижня.

$$x_n = \frac{x - \min}{\max - \min} \tag{2.1}$$

$$x_n = \frac{x - \mu}{\sigma} \tag{2.2}$$

$$x_{\sin} = \sin\left(\frac{2\pi x}{N}\right), \qquad x_{\cos} = \cos\left(\frac{2\pi x}{N}\right) \tag{2.3}$$

де x – значення ознаки, min – мінімальне значення ознаки, max – максимальне значення ознаки, μ – середнє значення ознаки, σ – стандартне відхилення ознаки, N – загальної кількості різних значень ознак.

Деякі з алгоритмів можуть мати проблеми з розмірністю та працювати значно повільніше порівняно з іншими алгоритмами. Проблема вирішується шляхом зменшення кількості вимірів за допомогою аналізу головних. Метод застосовується лише для конкретних алгоритмів і конкретних представлень даних, залежно від швидкості навчання та прогнозування. Крім того, деякі з алгоритмів працюють лише з певним форматом введення.

Існує багато алгоритмів машинного навчання, які можна використовувати для прогнозування збоїв пристрою. Вони можуть відрізнятися за багатьма властивостями та ознаками. Вони можуть бути контрольованими чи неконтрольованими, вони можуть вирішувати завдання класифікації, регресії чи кластеризації, або вони можуть належати до різних сімейств, таких як глибоке

навчання, деревоподібне, імовірнісне чи лінійне. Загальна кількість розглянутих для порівняння алгоритмів була обмежена через тривале налаштування та навчання. Вибір алгоритмів здійснювався за кількома критеріями:

1. Контрольоване навчання: виходячи з припущення, що дані, які будуть використовуватися для навчання алгоритмів, також містять записи про помилки, було вибрано контрольовані алгоритми навчання.

2. Практичне використання: деякі з алгоритмів використовуються більше, ніж інші, для прогнозного обслуговування.

3. Різноманітність: алгоритми були вибрані для представлення різних сімейств, завдань і функцій, таких як онлайн-навчання або прогнозування з часом.

На основі цих критеріїв вибір було зроблено з десяти алгоритмів, дев’ять з яких реалізовані як алгоритми класифікації, а один – як алгоритм регресії часових рядів (табл.2.1, табл. 2.2). Є представники різних родин і типів. Усі алгоритми підтримують онлайн-навчання або неявно, або через певні варіації.

Таблиця 2.1

Перелік методів машинного навчання та їх короткий опис

Англійська назва	Український переклад	Опис
k-Nearest Neighbor	k-Найближчі сусіди	Класифікаційний метод, що визначає клас об'єкта шляхом аналізу його найближчих сусідів у просторі ознак.
Decision Tree	Дерево рішень	Класифікаційний або регресійний алгоритм, що використовує структуру дерева для прийняття рішень на основі розділень за ознаками.
Random Forest	Випадковий ліс	Ансамбль дерев прийняття рішень, де кілька дерев об'єднуються для уникнення перенавчання та покращення точності.

Extreme Gradient Boosting	Екстремальне градієнтне підсилення	Ансамбль моделей, що використовують градієнтний підйом для покращення точності кожної наступної моделі.
Naive Bayes	Наївний Баєс	Ймовірнісний метод, що використовує теорему Байєса для класифікації на основі ймовірностей входження ознак у клас.
Support Vector Machine	Машина опорних векторів	Алгоритм, що визначає оптимальну границю розділення між класами за допомогою векторів підтримки.
Logistic Regression	Логістична регресія	Метод класифікації, що використовує логістичну функцію для визначення ймовірності належності об'єкта до певного класу.
Stochastic Gradient Descent	Стохастичний градієнтний спуск	Оптимізаційний алгоритм, що використовує градієнт для знаходження мінімуму функції втрат з випадково обраною підмножиною даних.
Multi-Layer Perceptron	Багатошаровий персептрон	Нейронна мережа з однією чи більше прихованими шарами, яка використовується для класифікації та регресії на основі вагових коефіцієнтів.
Long Short-Term Memory	Довга короткочасна пам'ять	Вид рекурентної нейронної мережі, призначений для збереження та використання інформації протягом тривалого періоду часу для передбачення збоїв або подій.

Таблиця 2.2

Порівняльна характеристика методів машинного навчання

Метод	Принцип роботи	Сфера застосування	Переваги	Недоліки
k-Nearest Neighbor	Визначення класу об'єкта через його найближчих сусідів	Виявлення аномалій, прогнозування збоїв	Простота реалізації, відсутність навчання	Чутливість до викидів, високі обчислювальні витрати
Decision Tree	Прийняття рішень на основі послідовних розділень за ознаками	Виявлення аномалій, класифікація збоїв	Легкість інтерпретації, вмістимість умов	Схильність до перенавчання, нестабільність
Random Forest	Ансамбль дерев для уникнення перенавчання	Виявлення аномалій, прогнозування збоїв	Висока точність, узгодженість рішень	Велика кількість гіперпараметрів, час навчання
Extreme Gradient Boosting	Використання градієнтного підйому для покращення точності	Прогнозування збоїв, виявлення аномалій	Висока точність, менше схильність до перенавчання	Велика кількість гіперпараметрів, складність інтерпретації
Naive Bayes	Використання теореми Байєса для ймовірнісної класифікації	Фільтрація аномалій, виявлення патернів збоїв	Ефективність для невеликих даних, простота моделі	Передбачення недостатньо гнучкі

Support Vector Machine	Визначення оптимальної границі розділення між класами	Класифікація аномалій, прогнозування збоїв	Ефективність у високо-розмірних просторах, гнучкість	Вимоги до підготовки даних, велика складність настройки
Logistic Regression	Визначення ймовірності належності об'єкта до певного класу	Класифікація збоїв, виявлення аномалій	Інтерпретабельність, простота реалізації	Потребує лінійної роздільної поверхні
Stochastic Gradient Descent	Використання градієнта для оптимізації функції втрат	Навчання моделей, аналіз збоїв	Швидкість навчання, ефективність для великих даних	Вимоги до гіперпараметрів, схильність до заїкання
Multi-Layer Perceptron	Нейронна мережа з однією чи більше прихованими шарами	Розпізнавання образів, прогнозування часових рядів	Здатність до вирішення складних задач	Потребує багато даних для навчання, час навчання
Long Short-Term Memory	Рекурентна нейронна мережа для довгострокового запам'ятовування	Аналіз часових послідовностей, прогнозування збоїв	Здатність розпізнавати залежності в часі	Високі обчислювальні витрати, складність настройки

Основні метрики для оцінки різних алгоритмів машинного навчання у завданнях передбачення або класифікації дозволяють здійснити об'єктивне порівняння ефективності цих алгоритмів.

Точність (Accuracy) представляє відсоток правильно класифікованих випадків у загальній кількості випадків, що дає загальне уявлення про точність алгоритму.

$$T=(TP+TN)/(FP+FN+TP+TN), \tag{2.4}$$

де TP – True Positive (правильно класифіковані позитивні), TN – True Negative (правильно класифіковані негативні), FP – False Positive (неправильно класифіковані позитивні), FN – False Negative (неправильно класифіковані негативні).

Точність класифікації (Precision) визначає відсоток правильно визначених позитивних класів серед усіх визначених позитивних класів, що є корисним для роботи з нерівномірними класами.

$$P = \frac{TP}{FP + TP} \ . \tag{2.5}$$

Повнота (Recall) відображає відсоток правильно визначених позитивних класів серед усіх фактичних позитивних класів, що є важливим для виявлення пропущених важливих випадків.

$$R = \frac{TP}{(TP + FN)} . \tag{2.6}$$

F1-середнє (F1-Score) використовує гармонічне середнє між точністю та повнотою, дозволяючи зрозуміти, наскільки добре модель вирішує завдання класифікації.

$$F1 = 2 \bullet \frac{P \bullet R}{P + R} \ . \tag{2.7}$$

ROC-AUC вимірює площу під кривою ROC та оцінює ефективність моделі в залежності від різних порогів класифікації, допомагаючи визначити її здатність до правильних передбачень.

$$ROC - AUC = \int_0^1 R(S)\,\mathrm{d(S)}\,. \tag{2.8}$$

Для апроксимації цієї площі, зазвичай використовують чисельні методи, такі як метод трапецій чи метод Сімпсона, оскільки сама формула ROC-AUC є інтегралом.

Повнота (R) та частота (S) визначаються за допомогою матриці плутанини для бінарної класифікації, яка включає чотири основних елементи: True Positive (TP), False Positive (FP), True Negative (TN) та False Negative (FN).

Показник *S* знаходиться за формулою (2.6), а частота за формулою (2.9).

$$S = \frac{FP}{FP + TN}\ . \tag{2.9}$$

Матриця плутанини (Confusion Matrix) дає детальну інформацію про реальні та передбачені класи, що допомагає оцінити рівень правильності та помилок для кожного класу, що є важливим при аналізі моделі.

Матриця плутанини допомагає оцінити продуктивність моделі класифікації, візуалізуючи реальні та прогнозовані значення. Вона є основою для розрахунку різних метрик, таких як точність, чутливість, специфічність, F1-оцінка та інші.

Ці метрики є ключовими для оцінки ефективності алгоритмів та вибору того, який найбільш підходить для конкретної задачі в залежності від вимог і потреб.

В таблиці 2.3 представлено ефективність різних методів машинного навчання за основними розглянутими метриками.

Таблиця 2.3 та рисунок 2.1 відображає ефективність різних методів машинного навчання за основними метриками, такими як точність, точність класифікації, повнота, F1-середнє та ROC-AUC. Оцінка "Висока", "Середня" або

"Дуже висока" в колонці "Ефективність" є узагальненою характеристикою ефективності методів, зробленою на підставі цих метрик.

Таблиця 2.3

Ефективність різних методів машинного навчання за основними метриками

Метод	Точність	Точність класифікації	Повнота	F1-середнє	ROC-AUC	Ефективність
k-Найближчі сусіди	0,85	0,81	0,89	0,85	0,92	Висока
Дерево рішень	0,78	0,82	0,75	0,76	0,85	Висока
Випадковий ліс	0,81	0,85	0,79	0,80	0,88	Висока
Екстремальне градієнтне підсилення	0,87	0,88	0,86	0,87	0,94	Висока
Наївний Баєс	0,75	0,79	0,72	0,73	0,82	Середня
Машина опорних векторів	0,82	0,84	0,80	0,81	0,89	Висока
Логістична регресія	0,79	0,83	0,77	0,78	0,86	Середня
Стохастичний градієнтний спуск	0,80	0,82	0,79	0,80	0,87	Середня
Багатошаровий персептрон	0,84	0,86	0,82	0,83	0,91	Висока
Довга короткочасна пам'ять	0,88	0,90	0,87	0,88	0,95	Дуже висока

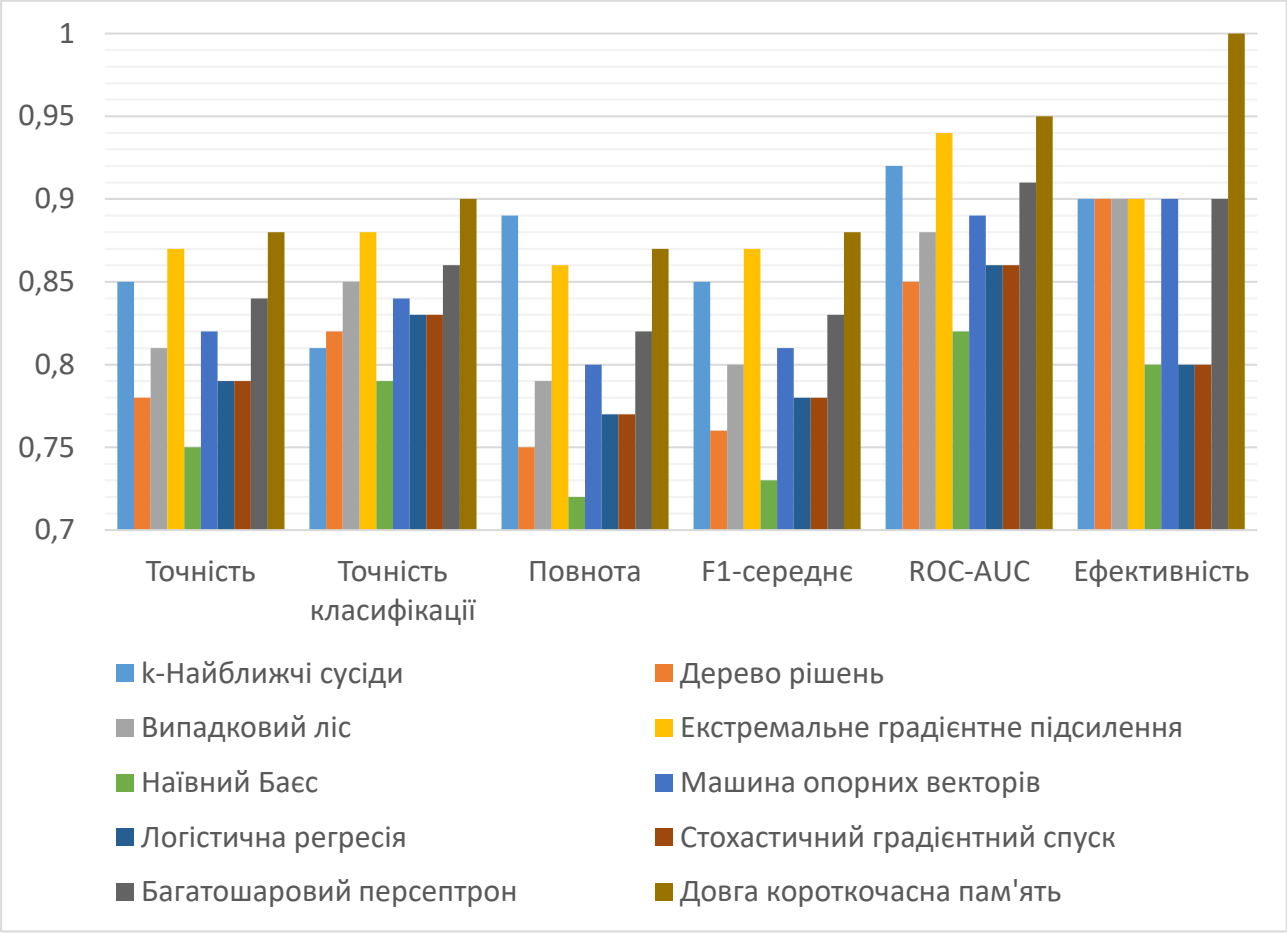

Рис. 2.1 Порівняння методів машинного навчання

За результатами дослідження було встановлено, що модель Довга короткочасна пам'ять (LSTM) вирізняється своєю здатністю працювати з послідовностями даних та зберігати інформацію на тривалий час. Це робить LSTM ефективним для аналізу часових рядів, таких як дані з сенсорів у розумному будинку, де інформація зазвичай є послідовною за часом.

Модель LSTM здатна зберігати інформацію протягом тривалого часу, дозволяючи їй ефективно розуміти та аналізувати послідовність даних з сенсорів у реальному часі. Використовуючи механізми, що забезпечують забування частини інформації та збереження інших, LSTM може врахувати довгострокові залежності та важливість окремих подій в часових рядах. LSTM може адаптуватися до різних обсягів даних, включаючи великі об'єми даних з сенсорів розумного будинку, і навчатися з них, що дозволяє створювати точніші прогнози збоїв. Модель LSTM

може адаптуватися до змінних умов та виявляти зміни в часових рядах, що дозволяє прогнозувати збої та аномалії в реальному часі. LSTM може обробляти різноманітні типи даних (текст, числа, послідовності тощо), що робить його універсальним для застосування в різних сценаріях прогнозування та аналізу.

Саме тому не дивно що даний алгоритм показав найкращі результати для прогнозування збоїв в розумному будинку.

У моделі LSTM на кожному кроці часу t маємо такі елементи:

1. Вхідні дані x_t.
2. Попередній вихід h_{t-1}.
3. Попередній стан пам'яті C_{t-1}.
4. Ворота:
   - Ворота забування f_t.
   - Ворота входу i_t.
   - Вихідні ворота o_t.
5. Новий стан пам'яті C_t.
6. Новий вихід h_t.

Сама модель складається з наступних формул:

1. Забування попереднього стану пам'яті:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) . \tag{2.10}$$

2. Визначення того, що буде оновлюватися в пам’яті:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) , \tag{2.11}$$

$$\tilde{C}_t = \mathrm{tg}\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) . \tag{2.12}$$

3. Оновлення стану пам'яті:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t . \tag{2.13}$$

4. Оновлення вихідного значення:

$$o_t = \sigma\, W_O \cdot [h_{t-1}, x_t] + b_O\,, \tag{2.14}$$

$$h_t = o_t * \mathrm{tg}\; C_t\,, \tag{2.15}$$

де f_t – ворота забування (forget gate), які вирішують, що потрібно забути про минулий стан, i_t – ворота входу (input gate), визначає, яка частина нового входу буде додана до стану пам'яті, $\tilde{C}_t$ – новий кандидат для стану пам'яті. C_t – стан пам'яті, o_t – вихідний ворота (output gate), визначає, який вихід буде наступним, x_t – вхід на кроці часу t, h_{t-1} – попередній вихід на кроці часу $t-1$,W та b – ваги та зсуви, які потрібно навчити під час процесу навчання.

Ці рівняння дозволяють LSTM визначати, яку інформацію забути, яку зберегти, а також як використовувати її для генерації вихідних значень.

Покроковий алгоритм для навчання моделі Довга короткочасна пам'ять (LSTM) містить наступні етапи:

1. Підготовка даних:
   - Вхідні дані: отримання набору даних, які містять часові ряди або послідовності.
   - Підготовка даних: нормалізація, перетворення формату даних у відповідність з вимогами моделі.
2. Побудова архітектури LSTM:
   - Створення моделі: використання бібліотек машинного навчання для побудови мережі LSTM.
   - Визначення параметрів: кількість шарів, кількість нейронів у кожному шарі, функції активації тощо.
3. Розділення даних:
   - Тренувальний та тестувальний набори: розділення даних на тренувальний та тестувальний набори для оцінки ефективності моделі.
4. Навчання моделі:

- Навчання моделі: підгонка LSTM до тренувальних даних з використанням зворотного поширення помилки (backpropagation).
- Оцінка моделі: оцінка відповідності прогнозів моделі фактичним даним на тестовому наборі.

5. Оцінка та покращення моделі:
   - Аналіз результатів: перегляд результатів прогнозу та оцінка їх точності.
   - Покращення моделі: використання технік оптимізації, зміна гіперпараметрів чи модифікація архітектури для покращення результатів.
6. Тестування та прогнозування:
   - Прогнозування збоїв: застосування навченої моделі до нових даних для прогнозування збоїв у розумному будинку.
7. Оцінка результатів:
   - Оцінка ефективності: порівняння прогнозів з реальними даними для визначення точності та ефективності моделі.
8. Підтримка та покращення моделі:
   - Постійне навчання: збір нових даних та покращення моделі на їх основі для підтримки актуальності прогнозів.

Це загальний опис алгоритму навчання LSTM для прогнозування збоїв в розумному будинку.

Підхід, представлений у цій роботі, використовує переваги LSTM для передбачення часових рядів. LSTM реалізовано за допомогою Keras (це високорівневий інтерфейс для роботи з нейронними мережами, який спрощує процес створення та навчання штучних нейронних мереж; це бібліотека машинного навчання, яка працює на базі фреймворків Tensorflow, Theano та Microsoft Cognitive Toolkit) у вигляді послідовної моделі з двома шарами LSTM та щільним вихідним шаром. Він отримує послідовність внесків даних (нормалізовані значення ознак без відмов) і видає послідовність значень відмов.

Було вирішено використовувати однократне введення даних, оскільки це значно скорочує час навчання і є достатнім для того, щоб алгоритм розпізнавав

патерни відмов. Довжина вихідної послідовності визначає тривалість часу виконання. Ефективність прогнозування досліджується на трьох різних вихідних послідовностях: 1 (1 секунда), 300 (5 хвилин) та 1800 (30 хвилин). В результаті було побудовано три різні моделі LSTM. Для навчання використовувався набір даних із 70% відмов, створивши розбиття на тренувальний і тестовий набори в пропорції 75%–25% без перемішування. Крім того, дані готувалися шляхом створення вихідних послідовностей для кожного запису даних, які потім використовувалися для тренування. Після успішного процесу тренування три моделі LSTM були оцінені на тестових даних. У вихідному шарі використовується сигмоїдна активаційна функція, значення знаходяться в межах від 0 до 1.

Сигмоїдна активаційна функція використовується в нейронних мережах для трансформації вхідних даних у вихідні значення в діапазоні від 0 до 1. Ця функція відображає будь-яке число на значення між 0 та 1, що робить її корисною для прогнозування ймовірностей. Формула сигмоїдної функції зазвичай виглядає так:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.16}$$

де e – це число Ейлера, а x – вхідні дані. Ця функція застосовується для вирішення завдань бінарної класифікації, де необхідно передбачити ймовірність належності даних до певного класу (наприклад, 0 або 1).

Тестові дані включають дні з п'ятниці по понеділок. Моделі змогли розпізнати патерни відмов, за винятком помилкових позитивних передбачень у суботу. Це може бути пояснено обмеженим розміром навчальних даних. Дані містять значення протягом двох тижнів, і тільки одна субота була для навчання. Оскільки прогнозування робляться кожну секунду для кожного введення даних, значення відмов у наступних вихідних послідовностях перекриваються. В результаті для однієї точки в майбутньому з різних вихідних послідовностей існують кілька передбачень відмов. Щоб мати змогу інтерпретувати результати, послідовності перетворюються в одновимірний масив. Для часу 5 та 30 хвилин це призводить до значно більшої кількості передбачених відмовних значень, ніж розмір тестових

даних. Однак патерн відмов все ще розпізнаваний і таке перетворення дозволяє обчислити показник F1 та точність.

Показник F1 – це міра точності моделі класифікації, яка об'єднує узгодженість між точністю (precision) і повнотою (recall) моделі. Він є гармонічним середнім між цими двома метриками та використовується для оцінки ефективності алгоритмів класифікації. Загальною метою є досягнення балансу між точністю (відсутність помилкових позитивних результатів) і повнотою (відсутність пропущених позитивних результатів). Чим вище показник F1, тим краща узгодженість між цими двома метриками і, отже, краща загальна ефективність класифікаційної моделі.

Додатковою вимогою для розрахунку показника F1 та точності є відображення значень передбачення на 1 або 0. Ці значення відображаються за допомогою знаходження оптимального порогу. Показник F1 та точність показані на Рис. 2.2 та 2.3. Очікувано, найвищий показник F1 у прогнозуванні зі збором даних кожну 1 секунду. Хоча передбачення з часом 5 хвилин має вищий показник F1, ніж той з часу 30 хвилин, різниця дуже мала і мало помітна.

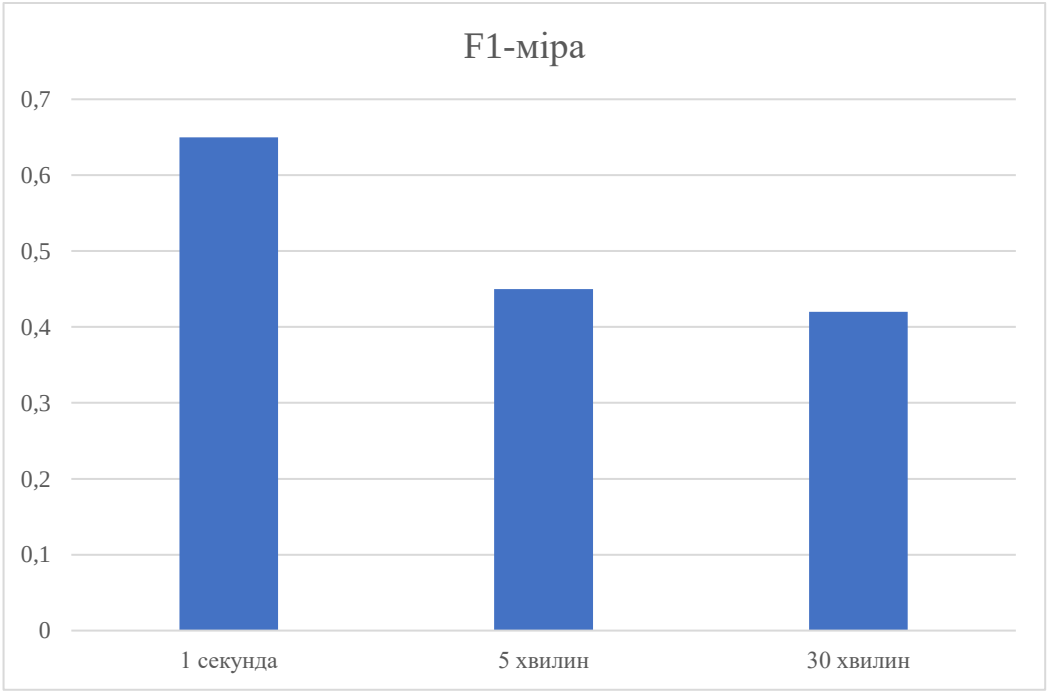

Рис. 2.2 Показник F1-міра

Рис. 2.3 Показник точність

Результати точності схожі, передбачення з часом 1 секунда має найвищу точність, за ними слідують 5 та 30 хвилин, які мають майже однаковий показник. Низька загальна точність може пояснюватися конкретним розбиттям даних і великою кількістю помилкових значень.

2.3. Побудова математичної моделі

Управління розумним будинком може бути оптимізоване за допомогою різних математичних моделей та алгоритмів для зменшення збоїв та оптимізації роботи системи. Однією з основних математичних моделей для цього є метод оптимізації, такий як лінійне програмування, який може бути застосований для мінімізації збоїв у системі розумного будинку.

Формули оптимізації поділяються на наступні:

1. Функція цілі: у математичній моделі оптимізації для розумного будинку потрібно визначити функцію, яку необхідно мінімізувати або максимізувати. Наприклад, це може бути функція, що враховує сумарні витрати на енергію та інші ресурси, або функція, яка враховує рівень комфорту мешканців будинку.

2. Обмеження: врахування обмежень є важливою частиною оптимізаційної моделі. Це можуть бути обмеження на використання енергії, температурні рамки

для систем опалення та кондиціонування повітря, часові обмеження для роботи певних пристроїв тощо.

3. Змінні: математичні змінні визначають параметри, якими можна маніпулювати для досягнення оптимального результату. Наприклад, це можуть бути рівні використання енергії різними пристроями, розподіл ресурсів між певними функціями будинку, регулювання температури тощо.

4. Функція витрат: це може бути функція, що описує загальні витрати, такі як енергетичні витрати або фінансові витрати на управління розумним будинком.

5. Оптимізаційний алгоритм: для знаходження оптимального рішення потрібен алгоритм, який може обчислювати значення змінних, щоб мінімізувати (або максимізувати) функцію цілі при заданих обмеженнях.

Ці формули та моделі можуть бути складними і залежать від конкретного типу розумного будинку, його систем та поставлених завдань оптимізації.

Якщо основною ціллю є зменшення кількості збоїв у системі розумного будинку, можна сформулювати функцію цілі, яка спрямована на мінімізацію цих збоїв. Для цього можна використати підхід, що враховує частоту виникнення збоїв або певний індекс надійності системи.

Формула може виглядати так:

$$minJ\left(x\right) = min\sum_{i=1}^{n} B_i \cdot p_i \tag{2.17}$$

де x – це вектор параметрів, які можуть впливати на кількість збоїв або на їхні ймовірності, B_i – це кількість збоїв для кожного окремого виду збою, а p_i – ймовірність виникнення кожного з цих збоїв.

Ця сума уявляє собою суму кількості збоїв, зважену їхньою ймовірністю виникнення, що може бути використана як метрика для оцінки загального рівня збоїв у системі розумного будинку.

Система обмежень при цьому має вигляд:

$$x_i \le T_i \tag{2.18}$$

$$\sum_{i=1}^{n} x_i \ge R \tag{2.19}$$

$$\sum_{i=1}^{n} x_i \le C \tag{2.20}$$

$$x_i \ge M_i \tag{2.21}$$

$$x_i \cdot p_i \ge D_i \tag{2.22}$$

Формула (2.18) вказує на те, що значення змінної x_i (яка може відповідати, наприклад, робочому часу пристрою або стану) не повинне перевищувати певного порогового значення T_i. Змінна x_i у цьому випадку відповідає конкретному параметру, що контролюється для запобігання відмови.

Формула (2.19) вказує на те, що сума значень параметрів x_i (може відповідати кількості резервних пристроїв або дублювання систем) повинна бути більшою або рівною R, щоб забезпечити надійність системи. Тут x_i використовується для представлення кількості конкретних пристроїв або систем, які готові взяти на себе функції в разі відмови основних.

Формула (2.20) вказує на обмеження на загальну кількість пристроїв чи їхній обсяг, які мережа може підтримувати без перевантаження та збоїв.

Формула (2.21) вказує на те, що значення параметру x_i повинне бути більше або рівним M_i, щоб забезпечити регулярне обслуговування та уникнення відмов.

Формула (2.22) вказує на обмеження щодо надійності пристрою x_i з урахуванням ймовірності відмови p_i та мінімальної надійності D_i.

Змінні x_i відображають різні параметри системи, які впливають на надійність та стабільність роботи розумного будинку. Такі змінні можуть відповідати кількості пристроїв, робочому часу, ймовірності відмови тощо. Ці формули можуть бути доповнені залежно від конкретних вимог та характеристик системи розумного будинку.

Крім того в розумному будинку слід враховувати загальні обмеження:

1. Енергоефективність:

- Потужність споживання ≤ Максимально допустима потужність (Наприклад, потужність споживання ≤ 10 кВт) – система має споживати енергію в межах максимально допустимої потужності для забезпечення енергоефективності.

2. Стабільність температури:
   - Температура приміщень повинна знаходитися у визначених межах Мінімальна температура ≤ Температура приміщення ≤ Максимальна температура (18°C ≤ Температура приміщень ≤ 25°C), щоб забезпечити комфорт у приміщеннях та оптимальну роботу систем опалення/охолодження.
3. Безпека:
   - Виявлення пожежі=Вчасно виявлено – система виявляє пожежу вчасно для запобігання значним руйнуванням та загрозам для мешканців.
4. Системи безпеки:
   - Сигнал тривоги ≤ Час реакції (Сигнал тривоги ≤ 30 секунд) – система безпеки має надсилати сигнал тривоги та реагувати на нього визначеним часом для запобігання небезпеці.
5. Керування витратами ресурсів:
   - Споживання води ≤ Дозволений ліміт споживання води (Споживання води ≤ 150 літрів на день) – система має ефективно використовувати водні ресурси.

3 РОЗРОБКА МЕТОДИКИ ПРОГНОЗУВАННЯ ЗБОЇВ В РОЗУМНОМУ БУДИНКУ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

3.1 Огляд платформи прогнозування збоїв

Платформа прогнозування збоїв передбачає існування платформи IoT, як показано у верхній частині рис.3.1. Зокрема, вона передбачає наявність двох джерел даних, одного для оперативних, а іншого для історичних і статичних даних, що нагадує архітектуру Lambda, головна ідея якої полягає в тому, що дані спочатку обробляються в режимі реального часу (операційний потік), а потім направляються на зберігання та аналіз (історичний потік). Це дозволяє системі працювати ефективно і в режимі реального часу, і в історичному контексті, забезпечуючи аналіз та висновки на основі накопичених даних.

Платформа IoT отримує дані з середовища IoT, які можуть бути отримані від датчиків і виконавчих механізмів (дані про навколишнє середовище), або безпосередньо від пристроїв, наприклад, про використання процесора і пам'яті (дані моніторингу). Крім того, платформа IoT зберігає загальну інформацію про пристрої (дані про пристрій). Дані про пристрій отримуються, коли пристрій реєструється в середовищі IoT, використовуючи, наприклад, компонент реєстрації пристрою. Процес може бути до певної міри автоматизований, але він все одно вимагає ручного втручання. Платформи IoT зазвичай використовують брокер повідомлень, який отримує дані з всіх різних пристроїв в середовищі IoT. Він може використовуватися як джерело реальних даних. Крім того, може існувати база даних, яка постійно зберігає отримані дані з середовища і статичні дані пристроїв. Зазвичай, ці два основні компоненти є частиною платформ IoT, як це видно на прикладі багатоцільової платформи для зв'язування та забезпечення, яка реалізує брокер Mosquitto та базу даних MongoDB. Однак важливо забезпечити лише ці два джерела даних незалежно від типів компонентів, що використовуються у фоновому режимі. Платформа IoT може явно надавати API-компоненту доступ до своїх

даних, або ж можна встановити пряме з'єднання з брокером і базою даних. У нижній частині рис.3.1 показано склад мережі, який повинен мати можливість встановлювати з'єднання з різними типами платформ IoT та джерелами даних. Компонент адаптера відокремлює платформу прогнозування збоїв від конкретних реалізацій і отримує дані, які будуть використовуватимуться моделями. Брокер повідомлень використовується для координації та постійного збереження даних після окремих етапів обробки.

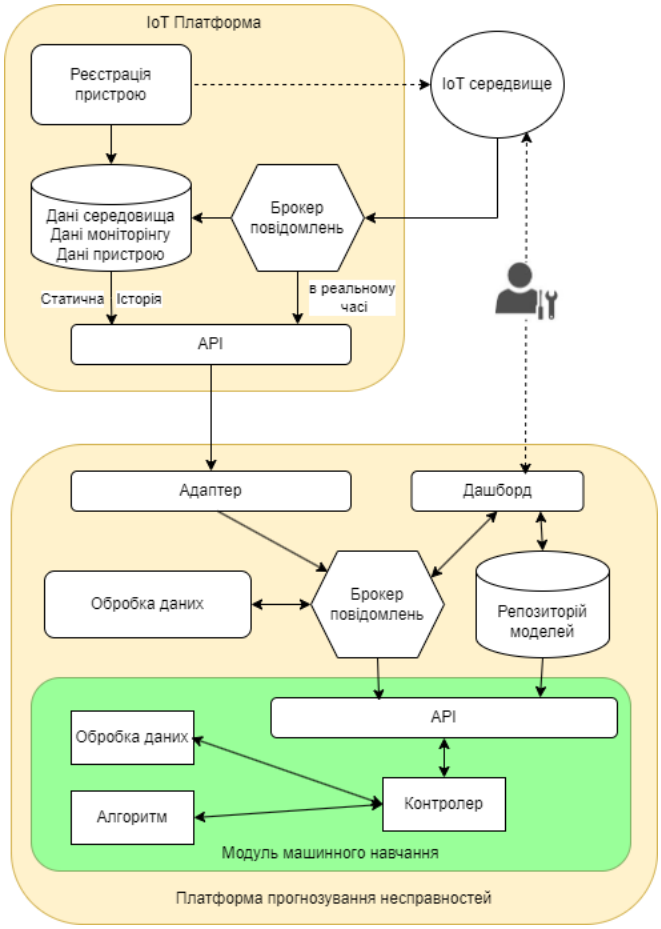

Рис.3.1 Архітектура платформи прогнозування збоїв. Суцільні лінії позначають управління та потік даних, пунктирні лінії для ручних завдань

Після того, як дані отримані адаптером, вони надсилаються до брокера і зберігаються в певній групі без змін. На наступному етапі дані можуть бути можуть бути оброблені в компоненті обробки даних, перш ніж вони будуть використані моделями. Він може бути стандартизованими за допомогою попередньо

визначеного шаблону даних, збагаченими інформацією, якої бракує, перетворені у певний формат або адаптовані в будь-який інший спосіб. Стандартизація даних за допомогою шаблону призводить до узгодженості даних, що в подальшому полегшує реалізацію моделей. Оброблені дані надсилаються і зберігаються знову на іншу групу, де вони можуть бути повторно використані моделями. Цей компонент необхідний при роботі з різнорідними даними і проводить обробку, необхідну для всіх моделей. Якщо припустити, що дані надходять у вигляді записів з позначками часу, він може бути використаний для вилучення значущої інформації з позначки часу, наприклад, годину дня, день тижня, свято тощо. Іншим потенційним варіантом використання, у випадку розділених груп пристроїв, є фільтрація даних за пристроями і надсилання їх до відповідної групи.

Кожна модель машинного навчання інтегрована як модуль, що складається з чотирьох основних компонентів: API, компонента обробки даних, контролера та власне алгоритму машинного навчання. API має однаковий склад для всіх модулів і складається з джерела повідомлень і двох споживачів повідомлень. Розрізняють споживача даних, який споживає попередньо оброблені дані, і споживача команд, який споживає команди, надіслані з інформаційної панелі. Джерело повідомлень надсилає результати моделі та інформацію про стан або прогрес брокеру повідомлень. Інтерфейс обміну повідомленнями відокремлює модулі машинного навчання від платформи. Хоча на попередньому кроці існує загальний компонент обробки даних. Тому в кожен модуль інтегровано додатковий компонент обробки даних, який має підготувати дані для конкретного алгоритму. Сам алгоритм є ядром модуля. Він проходить процес навчання, результатом якого є модель, що використовується для прогнозування. Алгоритм може бути реалізований різними способами і з різними бібліотеками, але він повинен забезпечувати інтерфейс для навчання і прогнозування. Нарешті, всі компоненти керуються контролером, який виконує кілька функцій. Він обробляє команди, отримані ззовні, і виконує відповідні дії, отримує дані, за потреби пересилає їх до компонента обробки, а потім – до алгоритму. Контролер може реалізовувати автоматизоване гіперпараметричне налаштування для пошуку оптимальних параметрів, тобто

найоптимальнішої конфігурації алгоритму на основі наданих даних. Контролер також зберігає і завантажує дані в базу даних або файлову систему. Він може реалізовувати планувальник для періодичного створення резервних копій, перенавчання алгоритму, якщо онлайн-навчання неможливе, або переналаштувати параметри. Він може оцінювати поточний стан обробки, а також надсилати прогнози моделі виробнику повідомлень.

Репозиторій моделей описує сховище загального призначення, спільне для всіх моделей. Сховище може бути базою даних, файловою системою або їх комбінацією. Основною метою є зберігання, опис та конфігурація алгоритмів, гіперпареметрів після налаштування або повних моделей машинного навчання, тобто внутрішні параметри після навчання. Він також може зберігати будь-яку іншу інформацію, що стосується моделі. Доступ до нього здійснюється через API-компоненти модулів і через інформаційну панель.

Після того, як модель машинного навчання готова і починає робити прогнози, відповідний модуль надсилає результати у визначену ним групу. Дашборд збирає всі прогнози від різних моделей, обробляє їх і візуалізує результати. Окрім груп для прогнозування, кожен модуль має групу, призначену для команд, що надсилаються з дашборду. Крім того, можуть бути додаткові групи, які використовуються модулями для надсилання інформації про стан та хід роботи. Інформаційна панель повинна надавати можливість взаємодіяти з модулями та виконувати окремі кроки вручну. Ці кроки можуть включати: імпорт історії, запуск автоматичного налаштування гіперпараметрів, навчання окремих алгоритмів, запуск і зупинка окремих моделей прогнозування, збереження та завантаження моделей тощо. Крім того, інформаційна панель має доступ до сховища моделей для отримання та відображення відповідної інформації про модель в інтерфейсі користувача. Інформаційну панель також можна використовувати для вставки інформації під час процесу реєстрації моделі. Нарешті, супровідник стежить за дашбордом і виконує ремонтні роботи або заміни в IoT-середовище. Також може бути реалізована система сповіщень, яка інформує техпідтримку, якщо якийсь пристрій вийшов з ладу або досяг критичних порогових значень на основі прогнозів.

3.2 Інтеграція платформи прогнозування збоїв в систему розумний будинок

На момент підключення платформи прогнозування до платформи IoT платформа IoT могла вже існувати та збирати певну кількість даних. Дані можна використовувати для початкового навчання алгоритмів.

Платформа прогнозування дозволяє отримати старі дані з бази даних, пройти загальний етап обробки та зберегти їх за певною темою. Після цього процес навчання алгоритмів можна розпочати індивідуально. Початкова підготовка є важливою, оскільки вона дозволяє плавно почати прогноз моделі зі значущими результатами прямо на початку. Але його також можна пропустити, особливо якщо до цього моменту не було зібрано жодних даних. Це призведе до навчання на льоту, або у вигляді онлайн-навчання, або перепідготовки через певні проміжки часу. Ще одна важлива перевага старих даних полягає в тому, що їх можна використовувати для налаштування скалерів, необхідних деяким моделям. Скалери зазвичай потребують обчислення внутрішніх параметрів, таких як мінімальне, максимальне або середнє значення, перш ніж їх можна буде застосувати до вхідних живих даних. Виконання цієї конфігурації на льоту є більш громіздким процесом, який, ймовірно, призведе до поганих результатів на початку. Крім цих переваг, існує обмеження, яке необхідно враховувати. Потенційно величезний обсяг даних, який міг би зберігатися в базі даних платформи IoT, не міг бути завантажений в пам'ять відразу. Тим не менш, у більшості випадків обробка даних у пакетів або навіть вибору частини даних може бути достатньо, щоб налаштувати масштабувальники, а також навчити алгоритми для надання значущих результатів на початку.

На другому кроці, перед початком будь-яких прогнозів моделі, платформа прогнозування встановлює з’єднання з брокером повідомлень на платформі IoT і починає отримувати живі дані. Знову ж таки, дані проходять етап обробки та потрапляють на певну групу. На цьому етапі прогнозування можна розпочати окремо для кожної моделі. Модуль отримує дані та за потреби обробляє їх далі. Потім модель робить прогноз, і контролер надсилає його до відповідної групи

прогнозів. Зрештою, прогнози витягуються та візуалізуються на інформаційній панелі. Дані, що проходять через систему, завжди зберігаються для кожної групи, за винятком груп, призначених для команд, а також інформації про статус або прогрес.

Отже, весь алгоритм інтеграції платформи прогнозування збоїв в систему розумний будинок, можна представити у вигляді чотирьох кроків:

Крок 1. Підключення до системи IoT та використання наявних даних:

- Встановлення зв'язку платформи прогнозування з системою IoT.
- Використання наявних даних для початкового навчання алгоритмів.

Крок 2. Отримання нових даних:

- Підключення платформи прогнозування до брокера повідомлень на платформі IoT.
- Отримання та обробка нових живих даних.
- Структурування даних за певною темою.

Крок 3. Прогнозування за допомогою моделей:

- Початок прогнозування за допомогою окремих моделей.
- Отримання, обробка та передача прогнозів до відповідних груп прогнозів.

Крок 4. Візуалізація та зберігання результатів:

- Витягнення та візуалізація прогнозів на інформаційній панелі.
- Зберігання даних для подальшого використання та оновлення моделей.

Такий алгоритм дозволяє ефективно працювати з даними, які вже існують та отримувати нові дані для постійного вдосконалення прогнозних моделей.

Процес інтеграції інформаційної системи в розумний будинок розпочинається з аналізу функціональних вимог та деталізації потреб (рис. 3.2). Наступним етапом є розробка спеціалізованих модулів, які забезпечують взаємодію інформаційної системи з розумним будинком через встановлення специфічних зв'язків та інтерфейсів.

Ці модулі інтегруються з вже існуючими пристроями та системами в розумному будинку, використовуючи мережі та комунікаційні протоколи для

обміну даними та керування. Важливим етапом є тестування цілісності та ефективності взаємодії, а також виправлення можливих помилок та несправностей.

Після успішного інтегрування системи у розумний будинок відбувається процес впровадження, коли система стає активною частиною будинкового середовища. Однак це лише початок: подальша підтримка, оптимізація та постійне вдосконалення системи відіграють ключову роль у забезпеченні її довготривалої та ефективної роботи в розумному будинку, адаптуючись до змін потреб та умов.

Рис. 3.2 Процес інтеграції інформаційної системи в розумний будинок

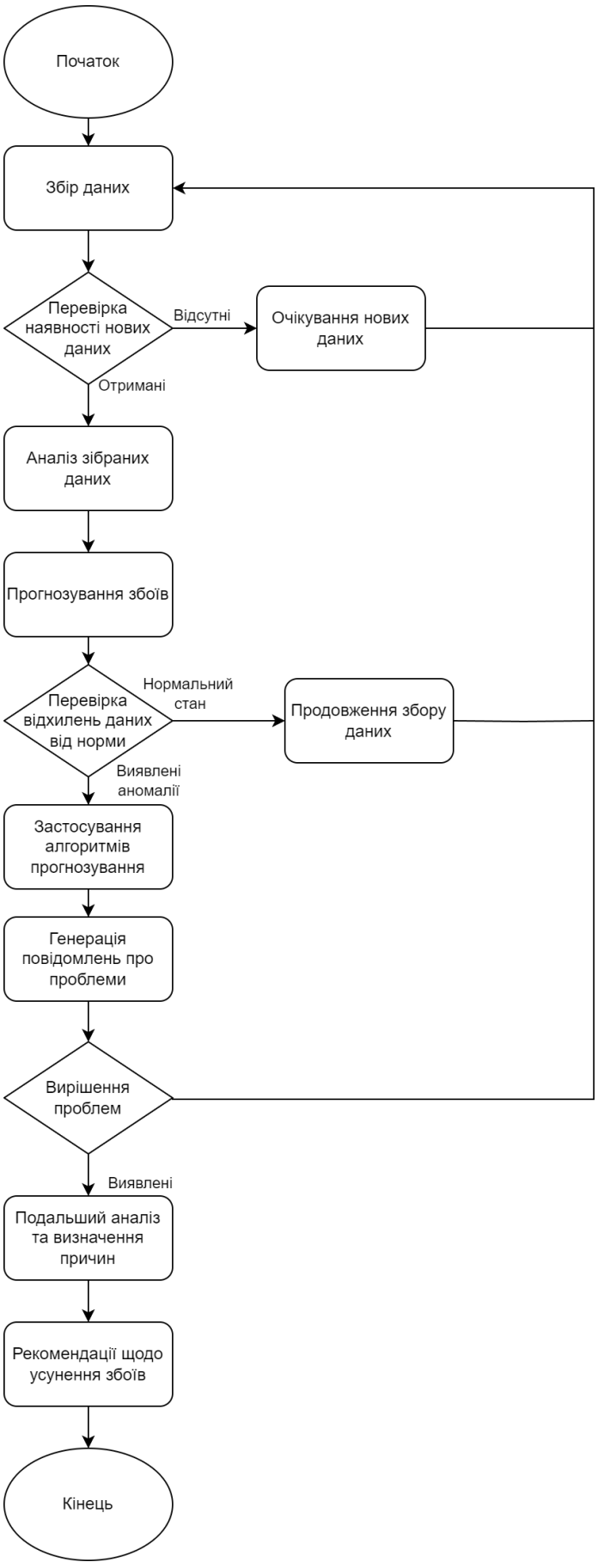

Рис. 3.3 Алгоритму функціонування розумного будинку з програмою прогнозування збоїв на основі машинного навчання

На рис. 3.3 представлена блок-схема алгоритму функціонування розумного будинку з програмою прогнозування збоїв на основі машинного навчання. Запропонований алгоритм можна поділити на наступні блоки:

1. Початок – звичайно позначає початок послідовності дій.
2. Збір даних – блок, що відповідає за перевірку та збір нових даних. Якщо дані відсутні, система очікує нові дані. Якщо дані отримані, вони проходять аналіз.
3. Прогнозування збоїв – тут відбувається перевірка даних на відхилення від норми. Якщо все в нормі, система продовжує збір даних. У випадку виявлення аномалій, застосовуються алгоритми прогнозування для визначення проблем та генерації повідомлень про них.
4. Вирішення проблем: якщо проблеми виявлені, система проводить подальший аналіз, визначає причини та рекомендації щодо їх вирішення.

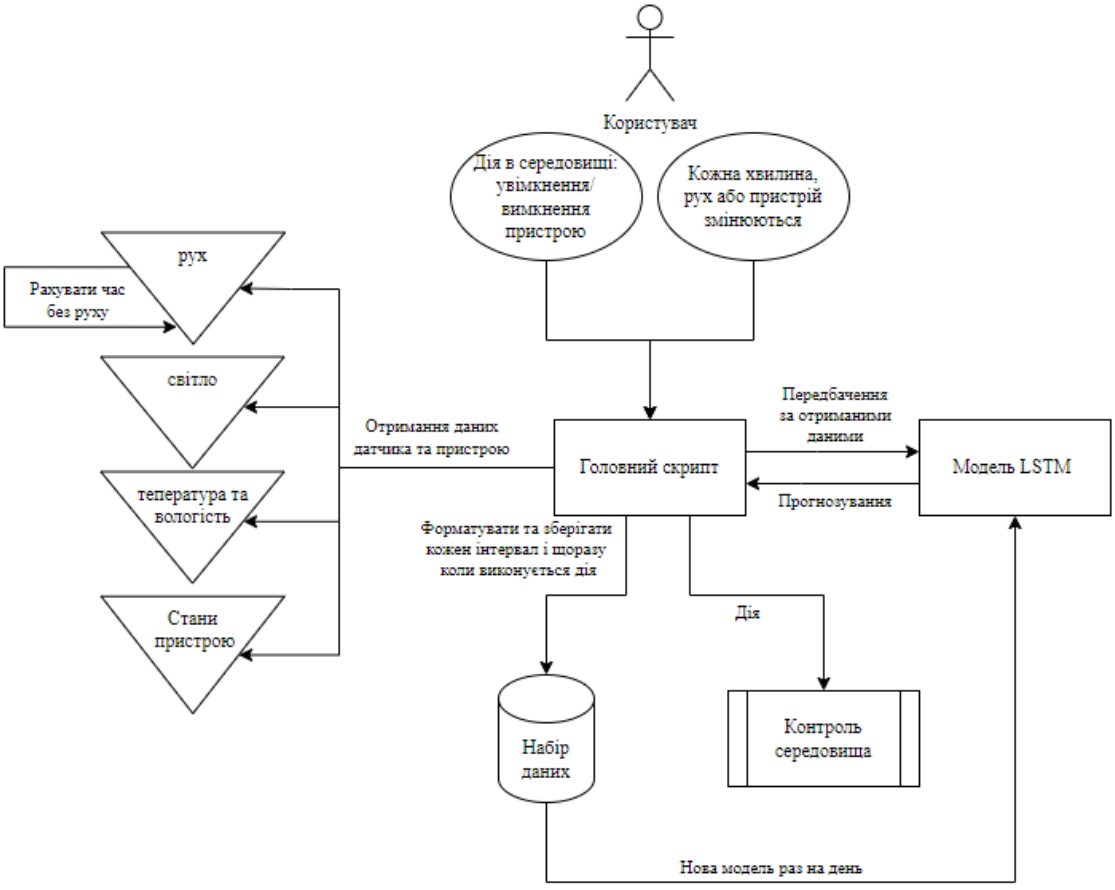

Рис. 3.4 Діаграма потоку

Кожен блок відповідає за певну частину функціоналу системи прогнозування збоїв у розумному будинку. Вони допомагають у зборі, аналізі та обробці даних, виявленні можливих проблем та наданні рекомендацій щодо їх усунення. Така система дозволяє розумному будинку бути більш автономним та реагувати на потенційні збої чи проблеми в реальному часі.

Після принаймні одного дня збору даних система може почати вивчати та визначати патерни. Набір даних розділяється на навчальний та оціночний набори. Навчальний набір складає 75% від загальної кількості даних, а решта 25% – оціночний. Це розбиття дозволяє приблизно оцінити точність моделі. Після створення моделі вона зберігається і може бути використана для оцінки нових даних. Система продовжує збирати дані про середовище.

Діаграма потоку на рисунку 3.4 показує весь робочий процес системи при інтеграції збору даних з впровадженням моделі. Блоки «рух», «світло», «температура та вологість», «стани пристрою», «головний скрипт» представляють основний процес моніторингу середовища в робочому потоці. Моніторинг використовується як для збору даних, для блоку «дія в середовищі», «набори даних», так і для прийняття рішень системи, позначених блоками «кожна хвилина, рух або зміни пристрою», «модель LSTM», «контроль середовища».

3.3 Опис програмного забезпечення

Платформу прогнозування збоїв було реалізовано за допомогою генератора JHipster1. JHipster створює повну веб-програму, яка забезпечує керування користувачами, адаптивний інтерфейс користувача, моніторинг та інші переваги. Реалізація прототипу використовує бекенд Java, реалізований за допомогою Spring2 фреймворк і інтерфейс Angular3. Крім того, він зберігає дані в базі даних MongoDB і використовує Kafka4 як брокер повідомлень. Інформаційна панель прогнозування збоїв представлена на рис. 3.5.

Програмне забезпечення для прогнозування збоїв в розумному будинку використовує джерела даних у вигляді багатоцільової платформи зв'язування та

забезпечення, включаючи MongoDB і Mosquitto. Запропонована розробка включає адаптер багатоцільової платформи, який створює зв'язок з базою даних MongoDB та окремий зв'язок із брокером Mosquitto. Щоб показати процеси навчання та прогнозування, було інтегрувано згенерований набір даних у базу даних та брокер. На панелі представлено моделі та їх функції, а також дві загальні функції, які застосовуються до всіх моделей.

Перша загальна функція відповідає за імпорт старих даних з бази даних та зберігання їх згідно з певною групою без змін. Для цього використовується Kafka – швидка потокова платформа, яка забезпечує якісне зберігання даних та дозволяє створювати та обмінюватися потоковими даними між різними системами та модулями, надаючи можливість обробки та аналізу цих даних у реальному часі.

Після цього дані проходять обробку в загальному компоненті, реалізованому в Java. З урахуванням використання згенерованого набору даних, вони вже попередньо оброблені, за винятком видалення ідентифікаторів MongoDB з кожного запису. Це виконується шляхом визначення шаблону даних без ідентифікатора MongoDB та відповідним зіставленням вхідних даних під час видалення ідентифікатора.

Оброблені дані знову зберігаються в іншій групі. Друга загальна функція активує процес моніторингу, встановлюючи зв'язок з брокером Mosquitto та отримуючи поточні дані. Дані в реальному часі проходять той самий шлях обробки, що й старі дані, поки їх не буде знову збережено після обробки. На жаль, наразі не існує механізму синхронізації, і ці дві функції можуть створювати дублікати, якщо вони виконуються одночасно.

Модель машинного навчання можна інтегрувати в платформу прогнозування збоїв у три етапи. Фрагмент архітектури платформи прогнозування збоїв і компонентів, які стосуються інтеграції, показано на рис. 3.5.

Щодо інтеграції моделі машинного навчання у платформу прогнозування збоїв, це може бути зроблено у три етапи. Спочатку, модель розробляється як окремий модуль машинного навчання. Цей модуль має вбудований виробник Kafka для передачі прогнозів та два споживачі Kafka для обміну даними та командами.

Важливо підібрати адекватні назви для груп, які використовуються в цьому контексті.

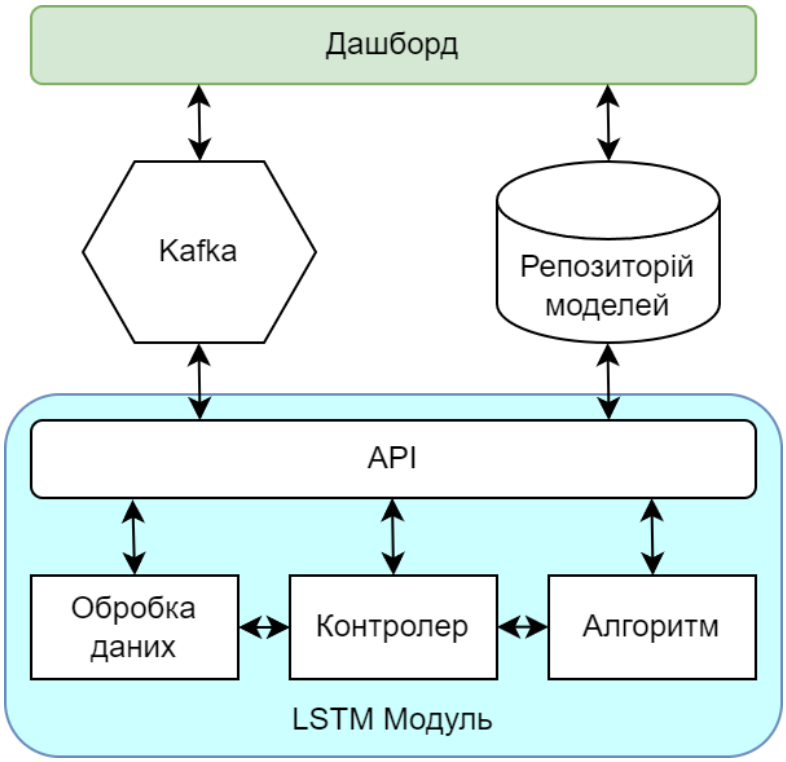

Рис. 3.5 Архітектура платформи прогнозування збоїв і компонентів

На наступному етапі модель реєструється в інформаційній панелі платформи, надаючи різноманітну інформацію про модель: назву, опис, тип та назви груп, які використовуються для комунікації та передачі прогнозів. Дані моделі зберігаються у репозиторії моделі, що дозволяє зручно управляти, переглядати, редагувати та видаляти їх. Область реєстрації моделі надає зручний інтерфейс для управління зареєстрованими моделями та додавання нових.

На завершальному етапі розробки модуля проводиться його незалежна інтеграція, що дає можливість взаємодіяти з інформаційною панеллю через систему Kafka. Для демонстрації була використана проста модель «Довга короткочасна пам'ять» з послідовністю одного значення відмови. Обрано дану модель через її підтримку онлайн-навчання та прогнозування часових рядів. Не менш важливим показником вибору моделі є проведений в другому розділі аналіз ефективності методів машинного навчання, який показав, що модель «Довга короткочасна

пам'ять» дає найкращу ефективності в процесі прогнозування. Цей модуль реалізовано на Python та інтегровано за допомогою описаних вище кроків. Він використовує абстрактний API-компонент з виробником та споживачами Kafka, а також попередньо визначені абстрактні методи для створення інтерфейсу. Це забезпечує однорідну реалізацію API у всіх моделях. Крім цього, для моделі «Довга короткочасна пам'ять» був створений спеціальний компонент обробки даних, який трансформує функції, готує послідовності та використовує масштабувальник для нормалізації даних. Останні два компоненти моделі – контролер і алгоритм. Контролер обробляє команди та здійснює відповідні дії, керуючи компонентами в моделі, а також запускаючи та припиняючи процес навчання та прогнозування алгоритму.

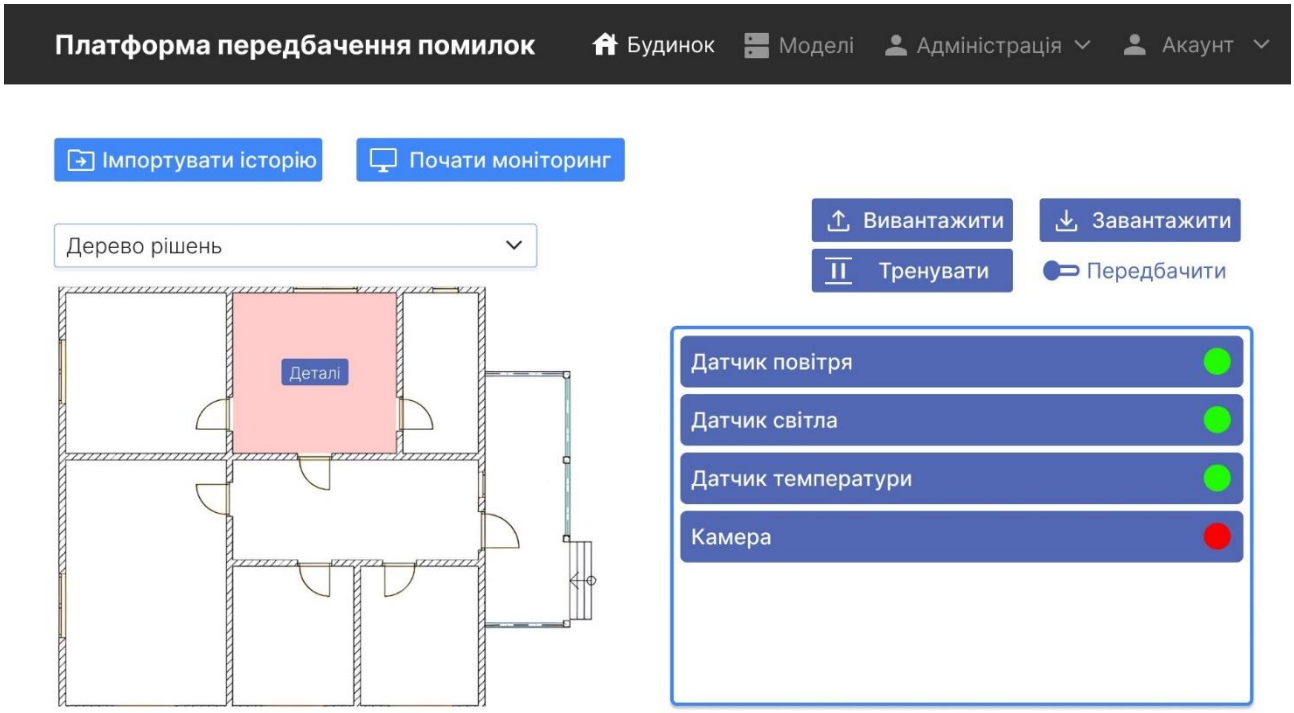

Рис. 3.6 Розроблене програмне забезпечення

Під час ініціалізації інформаційної панелі усі існуючі моделі та їхні прогнозні історії завантажуються, як це показано на рис. 3.6. Кожна модель має чотири взаємопов'язані функції. Для навчання моделей вони переходять до групи з обробленими даними. У цій реалізації обмежена кількість записів даних

використовується для одночасного завантаження всіх даних. Однак записи отримуються обмеженою кількістю або в партіях. Після завершення навчання можна починати прогнозування. Якщо для конкретної моделі передбачено онлайн-навчання, прогноз можна розпочати миттєво. Крім того, кожну модель разом з відповідним масштабувальником можна зберегти у вигляді файлів і завантажити у будь-який момент.

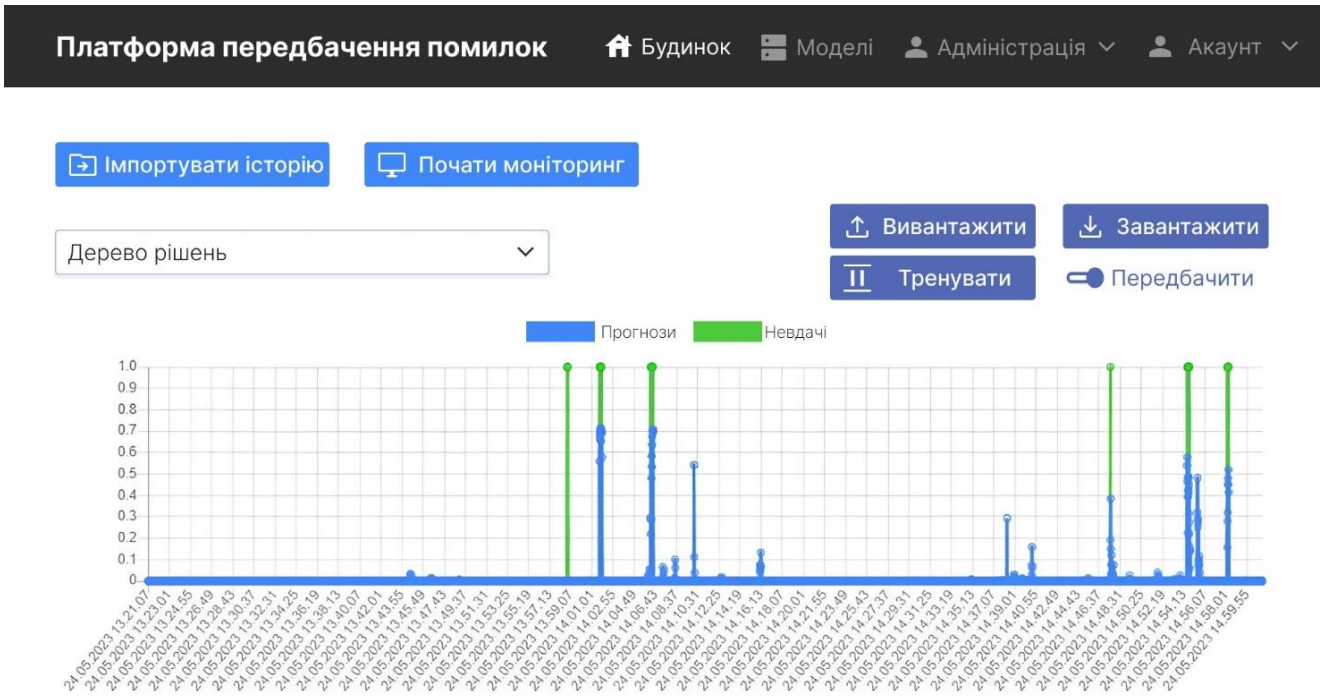

Рис. 3.7 Ініціалізація інформаційної панелі

Основний алгоритм роботи інформаційної системи можна описати наступним чином:

1. Ініціалізація та завантаження моделей:
    - Під час запуску системи завантажуються всі наявні моделі та їхні прогнозні історії.
    - Це відбувається для подальшого аналізу та використання цих моделей у процесі прийняття рішень.

Рис. 3.8 Алгоритм роботи інформаційної системи

2. Навчання моделей:
    - Моделі можна навчати, переходячи до відповідної групи з попередньо обробленими даними.
    - Обмежена кількість записів даних використовується для одночасного навчання, але можливість отримувати дані обмеженою кількістю чи партіями забезпечує ефективність процесу.
3. Проведення прогнозів:
    - Після завершення процесу навчання система готова до проведення прогнозів.
    - Нові дані, що надходять в ту саму групу, записуються для подальшого прогнозування.
    - Якщо для конкретної моделі передбачено онлайн-навчання, прогноз може бути здійснений миттєво.
4. Управління та збереження моделей:
    - Кожну модель та її відповідний масштабувальник можна зберегти як файли.
    - Це дає змогу у будь-який момент завантажити вже навчені моделі для подальшого використання.
5. Інтерфейс для взаємодії з користувачем:
    - Інформаційна панель надає можливість взаємодії з моделями, їхніми прогнозами та управлінням процесом навчання/прогнозування.
    - Користувач може взаємодіяти з системою через цей інтерфейс, здійснюючи операції з моделями та отримуючи результати прогнозів.

Кожен блок відображає окремий етап роботи інформаційної системи. Блок-схема на рис. 3.8 відображає послідовність кроків в процесі роботи системи, де кожен етап веде до наступного, управляючи різними аспектами, такими як завантаження моделей, навчання, прогнозування, управління та взаємодія з користувачем.

3.4. Дослідження ефективності введення блоку машинного навчання

Використання методів машинного навчання в системі розумного будинку дає значні переваги. Вони забезпечують систему здатністю аналізувати великі обсяги даних та виявляти патерни, які можуть передувати збоям. Це прогнозує можливі проблеми та надає можливість уникнути їх або готуватися заздалегідь. Призначена превентивна дія дозволяє системі автоматично приймати заходи для уникнення збоїв, наприклад, регулювати роботу пристроїв чи виконувати резервне копіювання на основі прогнозування.

Таблиця 3.1

Порівняльна характеристика систем розумного будинку з та без використання методів машинного навчання

Характеристика	З машинним навчанням	Без машинного навчання
Прогнозування збоїв	Прогнозування можливих проблем та уникнення їх заздалегідь	Реактивна реакція на збої після їх виникнення
Превентивні заходи	Автоматичні дії для уникнення збоїв	Відсутність автоматичних заходів перед збоями
Швидке виявлення та виправлення	Оперативна реакція на збої та їх виправлення	Реакція на збої після їх виникнення
Адаптація до змін	Здатність системи адаптуватися до змін у середовищі	Менша здатність до адаптації до змін
Оптимізація енергоспоживання	Ефективне використання енергії за допомогою автоматики	Обмежена здатність до енергоефективності
Вимоги до ресурсів	Більші вимоги до обчислювальних ресурсів та якості даних	Менші вимоги до обчислювальних ресурсів та даних

Системи з машинним навчанням швидше виявляють та виправляють збої, що дозволяє системі реагувати оперативніше, зменшуючи вплив на зручність користування. Ці алгоритми можуть також навчитися адаптуватися до змін в оточенні, наприклад, автоматично регулювати температуру в будинку враховуючи зміни погоди або уподобання мешканців. Оптимізація енергоспоживання є ще однією перевагою – системи можуть адаптуватися до звичок мешканців та ефективно використовувати енергію, що може призвести до значного зниження витрат. Такий підхід покращує якість обслуговування, забезпечує ефективне функціонування системи та підвищує комфорт користувачів.

Таблиця 3.2

Показники порівняння системи розумного будинку з та без використання методів машинного навчання

Показник	**З машинним навчанням**	**Без машинного навчання**
Кількість даних (за тиждень)	10 000	1 000
Кількість збоїв (за тиждень)	3	5
Швидкість виявлення збоїв	5 хвилин після виникнення	30 хвилин після виникнення
Реакція на збій	Автоматичне виправлення	Ручна реакція
Прогнозування збоїв	Так	Ні

Отже, без машинного навчання система допустила 5 збоїв:

1. Втрата зв'язку – проблеми з підключенням у бездротових мережах.
2. Несправність датчиків – пошкодження несправність датчика температури, що призвело до відсутності даних.
3. Помилки в програмному забезпеченні – призвело до неправильної роботи системи.

4. Несправність автоматизації – система автоматизації не реагувала на сигнали.

5. Енергетичні проблеми –проблеми із живленням пристроїв.

Система з машинним навчанням уникнула:

1. Прогнозування втрати зв'язку – за допомогою аналізу попередніх даних про стан мережі та сигналу, модель машинного навчання передбачила можливі проблеми у зв'язку та активувала механізми відновлення зв'язку до того, як це сталося.

2. Раннє виявлення несправностей датчиків – модель виявила аномальність у звичних показниках датчиків, що свідчило про можливу несправність. Вчасна реакція на ці сигнали допомогла вжити заходів до виникнення серйозних проблем.

Система не змогла уникнути:

1. Помилки в програмному забезпеченні – машинне навчання може покращити виявлення деяких помилок у програмному забезпеченні, але не завжди може передбачити або уникнути виникнення складних програмних помилок або критичних вразливостей.

2. Несправність автоматизації –певні види несправностей або блокування автоматизованих процесів можуть бути поза зонами передбачення системи машинного навчання, особливо якщо вони пов'язані із складними інтеракціями між пристроями.

3. Енергетичні проблеми – втрата електропостачання або енергетичні проблеми можуть виникнути без попередніх сигналів або змін у звичних показниках, що робить їх складними для передбачення.

Машинне навчання допомагає уникнути певних проблем через аналіз попередніх даних та розпізнавання патернів, але не може передбачити абсолютно всі можливі сценарії, особливо якщо вони виникають із певних непередбачуваних чинників або втручань третіх осіб.

Система розумного будинку, яка використовує методи машинного навчання, виявляється кращою порівняно із системою без цієї технології (як показали результати дослідження показники розумного будинку з використанням методів

прогнозування збоїв дають в середньому на 22% кращий результат порівняно з аналогічною системою без прогнозування – рис. 3.9). Машинне навчання дозволяє системі адаптуватися до змін у середовищі та вимог користувачів, швидше реагувати на нові умови та оптимізувати використання ресурсів. Це сприяє покращенню ефективності системи в управлінні енергією, комфортом, безпекою та задоволенням користувачів.

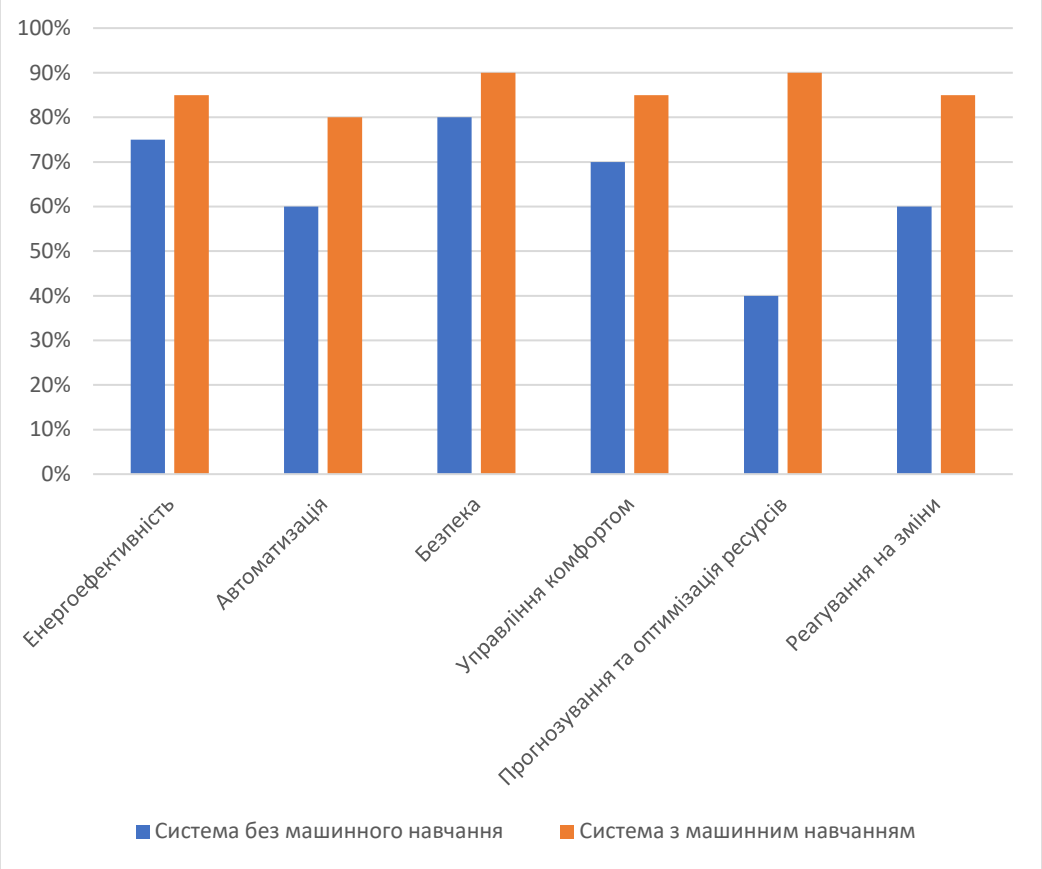

Рис. 3.9 Показники системи розумного будинку з та без використання методів машинного навчання

Машинне навчання дозволяє системі прогнозувати збої та уникати їх, що забезпечує більшу надійність та довговічність роботи системи. Такий підхід також забезпечує зростання рівня автоматизації, допомагаючи системі виконувати рутинні завдання без втручання користувача. Загалом, система з машинним навчанням залишається кращим вибором завдяки своїм можливостям у прогнозуванні, оптимізації та адаптації до змін, що дозволяє їй забезпечувати ефективніше та зручніше управління розумним будинком.

ВИСНОВКИ

Мета магістерської роботи, яка полягає в розробці ефективної системи прогнозування можливих несправностей та збоїв в інтелектуальних системах розумного будинку повністю досягнута.

Результати дослідження показали широкий спектр сучасних технологій, датчиків та систем управління, що застосовуються в розумних будинках. Огляд показав, що наявні технології мають потенціал для підвищення зручності, безпеки та енергоефективності.

Дослідження виявило кілька потенційних збоїв у розумних системах, зокрема пов’язаних зі з’єднанням, сенсорами та програмним забезпеченням. Це дає змогу заздалегідь підготувати систему до управління такими ситуаціями.

Аналіз доступних методів машинного навчання вказує на їх потенціал у прогнозуванні та управлінні ризиками у розумних будинках. Розглянуті моделі показали високу точність у передбаченні збоїв.

Розроблена методика прогнозування збоїв на основі методів машинного навчання дозволяє ефективно передбачати та управляти можливими ризиками.

Загальний аналіз показав позитивний вплив введення систем машинного навчання у роботу розумного будинку. Підвищення надійності та здатності до адаптації до змінних умов відзначається.

Розроблене програмне забезпечення має важливе практичне значення та дозволяє реалізувати представлену в роботі платформу прогнозування збоїв.

Для подальшого розвитку цієї роботи можуть бути розглянуті наступні напрямки:

- Розширення обсягу аналізу збоїв та розробка більш деталізованих моделей прогнозування.
- Вдосконалення методів машинного навчання для підвищення точності передбачення збоїв.
- Поглиблене дослідження впливу введення систем машинного навчання на функціонування розумного будинку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Базак Ю.К. «Порівняння методів машинного навчання для прогнозування збоїв в розумному будинку». Тези доповіді на Науково-практична конференція «Проблеми комп'ютерної інженерії». Збірник тез. – К.: ДУІКТ, 2023. с. 125-127

2. Бойко А. М. Моделювання автоматизованої системи оперативного управління параметрами "розумного будинку" в середовищI Proteus [Електронний ресурс] / А. М. Бойко, В. Б. Дроменко // Технології та дизайн. - 2020. - № 2 (35). - Режим доступу: http://nbuv.gov.ua/UJRN/td_2020_2_16.

3. Бондаренко, Я. С. Посібник до вивчення дисципліни "Статистичний аналіз даних" [Текст] / Я.С. Бондаренко, С.В. Кравченко. – Д: Ліра, 2018. – 40 с.

4. Гурєєва, К. М., Кудін, О. В., & Лісняк, А. О. (2018). Огляд методів машинного навчання в задачі прогнозування фінансових часових рядів. *Computer Science and Applied Mathematics*, (2), 18-28. http://journalsofznu.zp.ua/index.php/comp-science/article/view/1224

5. Жебка В.В., Базак Ю.К., Сторчак К.П. Особливості прогнозування збоїв в розумному будинку на основі методів машинного навчання // Телекомунікаційні та інформаційні технології. 2023. № 4 (81), с. 4-12.

6. Кононова К. Ю. Машинне навчання: методи та моделі: підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка» / К. Ю. Кононова. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.

7. Котунова, Д. Г. Огляд DIY елементів для систем «Smart Home» / Д. Г. Котунова, О. М. Павловський // XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», 13-14 травня 2020 р., м. Київ, Україна : збірник праць конференції. – Київ : КПІ ім. Ігоря Сікорського, 2020. – С. 35–38. 22.

8. Математичний аналіз: Кратні, криволінійні та поверхневі інтеграли, гармонічний аналіз, частина 3. [Електронний ресурс] : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад. Ю. Є.

Бохонов. – Електронні текстові дані (1 файл: 2,76 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 109 с.

9. Моніт Я.В. Система «Розумний будинок» з відкритим програмним забезпеченням/ Я.В.Моніт // XIX науково-технічна конференція студентів та молодих учених «Гіротехнології, навігація, керування рухом та конструювання авіаційно-космічної техніки», 15-16 лютого 2016 р. – К.: «Політехніка», 2016. – С. 43-44

10. О. О. Синявська, П. В. Слюсарчук. Ряди Фур'є. Навчальний посібник для студентів спеціальностей математика, прикладна математика, статистика. – Ужгород, 2015. – 70 с.

11. Пулеко І. В. Єфіменко А. А. Архітектура та технології Інтернету речей: навч. посіб. / І.В. Пулеко, А.А. Єфіменко. – Електронні дані. – Житомир: Державний університет «Житомирська політехніка», 2022. – 234 с.

12. Рубан І.В., Мартовицький В.О., Партика С.О. Класифікація методів виявлення аномалій в інформаційних системах // Системи озброєння і військова техніка, No. 3(47), 2016. 47.

13. Рудик І.І. Виявлення аномалій в комп'ютерній мережі на основі нейромережних технологій // Штучний інтелект, №. 2, 2012.

14. Статистичний аналіз даних вимірювань: навч. посіб. / Єременко В.С., Куц Ю.В., Мокійчук В.М., Самойліченко О.В. – К.: НАУ, 2013.– 320 с.

15. Технологія розумного будинку: як AI створює простір, комфортний для життя [Електронний ресурс]. – Режим доступу до ресурсу: https://www.everest.ua/tehnologiya-rozumnogo-budynku-yak-ai-stvoryuye-prostirkomfortnyj-dlya-zhyttya/

16. Харченко В. О. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.

17. Що таке розумний будинок? Все що потрібно знати про систему Розумний Дім [Електронний ресурс]. – Режим доступу до ресурсу: https://bron.ua/article/schotake-rozumnij-budinok-vse-scho-potrbno-znati-pro-sistemu-rozumnij-dm/5/

18. C. Zhang, M. Wang, M. Zhang, D. Wang, C. Song, L. Guan, Z. Liu. “Adaptive Failure Prediction Using Long Short-term Memory in Optical Network”. In: 2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC). 2019, pp. 1–3.

19. E. Westkämper, D. Spath, C. Constantinescu, J. Lentes. Digitale Produktion. SpringerLink : Bücher. Springer Berlin Heidelberg, 2013.

20. H. Zhuang, Y. Zhao, X. Yu, Y. Li, Y. Wang, J. Zhang. “Machine-Learningbased Alarm Prediction with GANs-based Self-Optimizing Data Augmentation in Large-Scale Optical Transport Networks”. In: 2020 International Conference on Computing, Networking and Communications (ICNC). 2020, pp. 294–298.

21. Huawei Technology Co, "Smart Home," 2015. [Online]. Available: http://www.huawei.com/minisite/iot/en/smarthome.html . [Accessed 7 May 2018].

22. J. Mineraud, O. Mazhelis, X. Su, S. Tarkoma. “A gap analysis of Internet-ofThings platforms”. In: Computer Communications 89-90 (2016). Internet of Things Research challenges and Solutions, pp. 5–16.

23. M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, J. Loncarski. “Machine Learning approach for Predictive Maintenance in Industry 4.0”. In: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). 2018, pp. 1–6

24. M. Rafiuzzaman, J. Gascon-Samson, K. Pattabiraman, S. Gopalakrishnan. “Failure Prediction in the Internet of Things Due to Memory Exhaustion”. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC ’19. Limassol, Cyprus: Association for Computing Machinery, 2019, pp. 292–301.

25. M. Rafiuzzaman, J. Gascon-Samson, K. Pattabiraman, S. Gopalakrishnan. “Failure Prediction in the Internet of Things Due to Memory Exhaustion”. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. SAC ’19. Limassol, Cyprus: Association for Computing Machinery, 2019, pp. 292–301.

26. N. Pandey, O. P. Verma, A. Kumar. “A framework for usage pattern–based power optimization and battery lifetime prediction in smartphones”. In: Personal and Ubiquitous Computing (2019), pp. 1–16

27. O. Vermesan, P. Friess, eds. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers Series in Communication. Aalborg: River, 2013.

28. R. Pinciroli, L. Yang, J. Alter, E. Smirni. *The Life and Death of SSDs and HDDs: Similarities, Differences, and Prediction Models*. 2020. arXiv: 2012.12373

29. T. Arsan, "Smart Systems: From design to implementation of embedded Smart Systems," 2016. [Online]. Available: http://ieeexplore.ieee.org.focus.lib.kth.se/document/7753420/.

30. X. Sun, K. Chakrabarty, R. Huang, Y. Chen, B. Zhao, H. Cao, Y. Han, X. Liang, L. Jiang. “System-Level Hardware Failure Prediction Using Deep Learning”. In: Proceedings of the 56th Annual Design Automation Conference 2019. DAC ’19. Las Vegas, NV, USA: Association for Computing Machinery, 2019.