

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Система розумного будинку в автономному режимі на основі нейронної
мережі керування
(тема)

Виконав: студент 2 курсу, групи СКСм-22-1

Солодовник Ю.Ю.
(прізвище, ініціали)

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

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

Освітня програма _____
Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник проф. Немченко Н.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Чумаченко С. В.
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Автоматизації проектування обчислювальної техніки _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 Комп'ютерна інженерія _____
(шифр і назва)

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

Освітня програма _____ Спеціалізовані комп'ютерні системи _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«03» вересня 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Солодовнику Юрію Юрійовичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Система розумного будинку в автономному режимі на основі нейронної мережі керування _____

затверджена наказом по університету від _____ 03.11.2023 р. № _____ 1288СТ _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 05.01.2024 р.

3. Вихідні дані до роботи _____

Raspberry Pi

Аналіз середовища

Платформи Visual Studio/VS code

Мови програмування C#, TS, Python

4. Перелік питань, що потрібно опрацювати в роботі _____

Штучний інтелект


Нейронні мережі

Створення обробки та класифікації

Створення застосунку











5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17 слайдів

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спец. частина	Проф. Немченко В.П.		05.01.24

7. Дата видачі завдання 02.09.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	02.09.2023-06.09.2023	Вик. 
2	Аналіз літератури за темою	08.09.2023-15.10.2023	Вик. 
3	Суть технічної проблеми	16.10.2023-20.10.2023	Вик. 
4	Постановка задачі	21.10.2023-25.10.2023	Вик. 
5	Існуючі методи для вирішення задачі	26.10.2023-28.10.2023	Вик. 
6	Розробка алгоритму застосунку	29.10.2023-05.11.2023	Вик. 
7	Програмна реалізація застосунку	06.11.2023-07.12.2023	Вик. 
8	Тестування застосунку	08.12.2023-15.12.2023	Вик. 
9	Оформлення пояснювальної записки	16.12.2023-23.12.2023	Вик. 
10	Оформлення графічної частини	23.12.2023-27.12.2023	Вик. 

Студент 
(підпис)

Керівник роботи  проф. Немченко Н.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить 78 сторінок, 22 рисунки, 9 джерел за переліком посилань.

IOT, RASPBERRY PI, ML, НЕЙРОННІ МЕРЕЖІ, АЛГОРИТМ LIGHTGBM, РОЗУМНИЙ БУДИНОК, FFNN, KERAS, ANGULAR, ASP.NET

Метою роботи є розгляд використання нейронної мережі для керування системою розумного будинку в автономному режимі а також їх практичне застосування. Досліджено існуючі методи та технології у сфері розумних будинків, а також застосування нейронних мереж у системах керування. Визначено потенційні області застосування нейронних мереж у керуванні розумним будинком та виявлено переваги та виклики їх використання.

У роботі розглянуто основні поняття та теоретичні засади нейронних мереж, включаючи їх структуру, принцип роботи, типи та алгоритми навчання. Наведено практичну реалізацію двох підходів які можуть працювати в симбіозі: Feedforward Neural Network для оцінки загального стану та застосування алгоритму LightGBM для класифікації пропонованих дій у разі невідповідності системи задовільному стану.

Для забезпечення взаємодії між системою та підключеними пристроями використовуються сенсори, що збирають дані про стан приміщення, температуру, освітлення тощо. Зібрані дані передаються до нейронної мережі, яка на основі отриманої інформації приймає рішення щодо керування системами розумного будинку.

Запропоновано реалізацію системи на базі двох модулів для аналізу та класифікації дій за показниками та застосунок для відображення інформації та керування системою. Для розгортання даної системи використано одноплатний комп'ютер Raspberry Pi 4.

ABSTRACT

The explanatory note to the qualification work contains 78 pages, 22 figures, 9 sources according to the list of references.

IOT, RASPBERRY PI, ML, NEURAL NETWORKS, LIGHTGBM ALGORITHM, SMART HOUSE, FFNN, KERAS, ANGULAR, ASP.NET

The purpose of the work is to consider the use of a neural network for controlling the smart home system in an autonomous mode, as well as their practical application. The existing methods and technologies in the field of smart buildings, as well as the application of neural networks in control systems, were studied. The potential areas of application of neural networks in the control of a smart house are determined and the advantages of their use are identified.

The work discovers the basic concepts and theoretical foundations of neural networks, including their structure, principle of operation, types and learning algorithms. The practical implementation of two approaches that can work in symbiosis is given: Feedforward Neural Network for assessing the general state and application of the LightGBM algorithm for classification of proposed actions in case of non-compliance of the system to a satisfactory state.

To ensure interaction between the system and connected devices, sensors are used that collect data on the state of the room, temperature, lighting, etc. The collected data is transmitted to the neural network, which, based on the received information, makes decisions about managing the smart home systems.

It is proposed to implement the system based on two modules for the analysis and classification of actions by indicators and applications for displaying information and managing the system. A Raspberry Pi 4 single-board computer was used to deploy this system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1 Визначення проблеми	12
1.2 Мета та завдання роботи.....	12
1.3 Об'єкт та предмет дослідження	13
1.4 Актуальність теми	14
2 ДОСЛІДЖЕННЯ В СФЕРІ РОЗУМНИХ БУДИНКІВ	16
2.1 Поняття розумного будинку	16
2.2 Стандарти та протоколи для розумних будинків	18
2.3 Огляд готових рішень для розумного будинку	20
2.3.1 Google Home та Google Nest	22
2.3.2 Amazon Alexa.....	23
2.3.3 Apple HomeKit	23
2.3.4 OpenHAB	25
2.3.5 Home Assistant	25
3 ОСНОВНІ ПОНЯТТЯ НЕЙРОННОЇ МЕРЕЖІ	27
3.1 Нейронні мережі структура та принцип роботи.....	27
3.1.1 Структура нейронної мережі	27
3.1.2 Принцип роботи нейронної мережі.....	28
3.2 Типи нейронних мереж	29
3.2.1 Перцептрон	30
3.2.2 Зворотнє поширення	30

3.2.3 Рекурентні мережі	31
3.2.4 Застосування типів нейронних мереж.....	32
3.3 Функції активації та їх вплив на роботу нейронних мереж	32
3.3.1 Сигмоїдна функція	33
3.3.2 Функція ReLU.....	33
3.3.3 Функція гіперболічного тангенсу	34
3.3.4 Інші функції активації.....	35
3.4 Алгоритми навчання нейронних мереж	36
3.4.1 Алгоритм зворотнього поширення помилки	36
3.4.2 Алгоритмічні методи оптимізації.....	37
3.4.3 Рекурентне навчання	37
3.4.4 Інші алгоритми навчання.....	38
4 ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У СИСТЕМАХ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ.....	40
4.1 Огляд сучасних рішень та технологій у сфері розумних будинків	40
4.2 Потенційні області застосування нейронних мереж у керуванні розумним будинком	41
4.3 Використання нейронних мереж у системах керування	43
4.3.1 Передбачуване керування.....	43
4.3.2 Управління енергоефективністю.....	45
4.3.3 Розпізнавання та ідентифікація	45
5 ПЕРЕВАГИ ТА ВИКЛИКИ РЕАЛІЗАЦІЇ СИСТЕМ РОЗУМНОГО БУДИНКУ НА БАЗІ НЕЙРОННИХ МЕРЕЖ	48
5.1 Адаптивність та самонавчання	48
5.2 Гнучкість і масштабованість.....	48

5.3	Здатність до вирішення складних проблем.....	48
5.4	Споживана потужність та енергоефективність	48
5.5	Безпека та конфіденційність даних	49
5.6	Інтеграція різних пристроїв та стандартів	49
5.7	Доступність та вартість	49
5.8	Надійність та скасування помилок	50
5.9	Потреба в великій кількості даних	50
5.10	Інтерпретованість результатів	50
6	РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ОЦІНКИ СЕРЕДОВИЩА.....	52
6.1	Нейронна мережа для аналізу стану середовища.	52
6.1.1	Застосування нейронних мереж для класифікації	52
6.1.2	Огляд архітектур нейронних мереж для класифікації	52
6.1.3	Процес навчання.....	53
6.2	Реалізація нейронної мережі для аналізу стану приміщення.....	54
6.2.1	Збір даних	54
6.2.2	Навчання нейронної мережі	58
6.3	Класифікація стану середовища	59
6.3.1	Алгоритм LightGBM	59
6.3.2	Програмна реалізація модулю класифікації	60
7	РЕАЛІЗАЦІЯ СИСТЕМИ РОЗУМНОГО БУДИНКУ НА БАЗІ НЕЙРОННОЇ МЕРЕЖІ.....	64
7.1	Загальна структура системи.....	64
7.2	Raspberry Pi.....	65
7.3	Бекенд частина	66
7.3.1	ASP.NET Core.....	67

7.3.2 Взаємодія з іншими компонентами системи	68
7.4 Аналіз стану показників середовища	69
7.4.1 Датчик температури та вологості. КУ-015	70
7.4.2 Датчик освітленості. ВН1750	71
7.5 Фронтенд частина.....	72
ВИСНОВКИ	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	77
Відомість кваліфікаційної роботи	78
ДОДАТОК А	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence (штучний інтелект);

ADC – Analog-to-digital converter (аналого-цифровий перетворювач);

BPTT – Backpropagation Through Time (алгоритм зворотнього поширення помилки);

CoAP – Constrained Application Protocol (протокол обмеженого застосування);

DI – Dependency Injection (ін'єкція залежностей)

ELU – Exponential Linear Unit (експоненціальна лінійна одиниця);

GPIO – General Purpose Input/Output (введення/виведення загального призначення);

GOSS – Gradient-based One Side Sampling (одностороння вибірка на основі градієнта);

IOT – Internet Of Things (інтернет речей);

MQTT – Message Queuing Telemetry Transport (телеметричний транспорт черги повідомлень);

MVC – Model-View-Controller (модель-представлення-контролер);

NEAT – NeuroEvolution of Augmenting Topologies (нейроеволюція наростаючої топології);

NN – Neural network (нейронна мережа);

ReLU – Rectified Linear Unit (Зрізаний лінійний вузол);

RESTful API – Representational State Transfer (передача представницького стану);

RNN – Recurrent Neural Networks (Рекурентні нейронні мережі);

SPA – Single page application (односторінковий застосунок).

ВСТУП

Розумні будинки стали все більш поширеним явищем в сучасному світі, де технології Інтернету речей (IoT) [1] забезпечують зручність, комфорт і ефективність в повсякденному житті. Проте, ефективне керування розумним будинком вимагає розробки інтелектуальних систем керування, які здатні адаптуватись до потреб користувачів і оптимізувати енергоспоживання.

Розумний будинок, на основі нейронної мережі може аналізувати, прогнозувати та приймати рішення на основі зібраних даних про звички та поведінку користувачів, а також інформації про стан середовища. Це дозволяє забезпечити оптимальне керування системами освітлення, опалення, вентиляції та інших пристроїв, забезпечуючи комфорт, енергоефективність і безпеку.

Метою дослідження є проведення огляду існуючих рішень у сфері розумних будинків, а також дослідження використання нейронних мереж у системах керування, структуру та принцип роботи нейронних мереж, функції активації та алгоритми навчання нейронних мереж. Проведено огляд сучасних рішень та технологій, визначено потенційні області застосування нейронних мереж у керуванні розумним будинком, а також розглянуто переваги та недоліки використання нейронних мереж у системах розумних будинків. Розробка прототипу та аналіз системи розумного будинку, яка базується на нейронній мережі та може функціонувати автономно без доступу до хмарних сервісів.

Дослідження у цій області має великі перспективи, оскільки розумні будинки можуть значно покращити якість життя людей, забезпечуючи комфортні умови проживання. Розробка ефективної нейронної мережі для керування розумним будинком може відкрити нові перспективи в цій області та допомогти вдосконалити функціональність розумних будинків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення проблеми

Сучасний розвиток інформаційних технологій сприяє появі інноваційних систем керування розумними будинками, які забезпечують автоматизоване та інтелектуальне керування різними системами в приміщенні. Однак, існують проблеми, пов'язані з ефективним управлінням такими системами, зокрема освітленням. Однією з основних проблем є необхідність в ручному керуванні освітленням, що призводить до неефективного використання енергії та зайвого споживання ресурсів.

1.2 Мета та завдання роботи

Метою даного дослідження є розробка нейронної мережі для керування системою розумного будинку в автономному режимі. Основна мета полягає в розробці архітектури ефективною та надійною системи керування, яка забезпечує оптимальне функціонування розумного будинку шляхом автоматизованого управління різними параметрами, такими як освітлення, опалення, кондиціонування повітря, безпека тощо.

Для досягнення цієї мети поставлені наступні завдання:

- 1) аналіз літератури та попередніх досліджень у сфері розумних будинків та застосування нейронних мереж у системах керування. Цей крок дозволить уточнити існуючі рішення, визначити проблемні аспекти та встановити основні вимоги до розроблюваної системи;
- 2) визначення архітектури нейронної мережі, включаючи кількість шарів, кількість нейронів у кожному шарі та типи зв'язків між нейронами;
- 3) розробка оптимальної архітектури є ключовим етапом для досягнення ефективності та точності системи керування;

4) розробка алгоритмів навчання нейронної мережі, що дозволяють системі самостійно вивчати та адаптуватися до змінних умов роботи розумного будинку. Це включає в себе визначення функцій активації, методів оптимізації та критеріїв оцінки продуктивності мережі;

5) експериментальне тестування розробленої нейронної мережі за допомогою симуляційних моделей та реальних даних. Цей етап дозволить оцінити ефективність та точність системи керування в різних умовах та порівняти її з існуючими рішеннями;

6) аналіз результатів та формулювання висновків щодо досягнення поставлених цілей, переваг та обмежень розробленої нейронної мережі в контексті системи керування розумного будинку;

7) інтеграція нейронної мережі в систему;

8) реалізація застосунку для моніторингу та налаштування системи;

Виконання цих завдань сприятиме розробці надійної та ефективної системи керування розумним будинком на основі нейронної мережі, що забезпечить зручність, енергоефективність та безпеку для користувачів.

1.3 Об'єкт та предмет дослідження

Об'єктом дослідження є система розумного будинку, яка включає в себе різноманітні підсистеми та компоненти, призначені для автоматизованого керування різними аспектами побутового середовища. До складу такої системи можуть входити елементи освітлення, опалення, кондиціонування повітря, безпеки, енергозбереження та інші, які взаємодіють між собою та зовнішнім середовищем.

Предметом дослідження є розробка нейронної мережі для керування системою розумного будинку в автономному режимі. Основними аспектами предмету дослідження є аналіз теоретичних засад нейронних мереж, визначення їх архітектури, алгоритмів навчання та експериментальне тестування в контексті розумних будинків. Розроблена нейронна мережа має

виконувати функцію ефективного та автоматизованого керування різними системами розумного будинку на основі збору та аналізу даних, прийняття рішень та виконання відповідних дій для досягнення заданих цілей. Створення застосунку для зручної взаємодії з даною системою.

Дослідження об'єкту та предмету дослідження спрямоване на вдосконалення системи керування розумним будинком шляхом впровадження нейронної мережі, що в свою чергу сприятиме підвищенню рівня комфорту розумних будинків.

1.4 Актуальність теми

Актуальність дослідження нейронної мережі для керування системою розумного будинку в автономному режимі базується на низці ключових факторів.

1) зростаюча популярність розумних будинків свідчить про необхідність вдосконалення систем керування, які забезпечують різні функціональні можливості та покращують якість життя мешканців. За даними звіту Grand View Research [2], ринок розумних будинків прогнозується досягти значної вартості до 2028 року, що свідчить про зростаючий інтерес до цього сектора;

2) інтеграція нейронних мереж у системи керування розумним будинком може принести значні переваги. Нейронні мережі мають потужний потенціал у вирішенні складних завдань прогнозування, адаптації до змінних умов та оптимального управління системою. Застосування нейронних мереж у системах розумного будинку може сприяти зниженню енергоспоживання, підвищенню безпеки, оптимізації комфорту та зручності для мешканців;

3) швидкий розвиток технологій, зокрема штучного інтелекту та інтернету речей, створює нові можливості для застосування нейронних мереж у розумних будинках. Інтернет речей, що передбачає підключення різних пристроїв та сенсорів до мережі, сприяє збору великого обсягу даних.

Використання нейронних мереж у системах керування розумним будинком може допомогти аналізувати ці дані, робити прогнози та здійснювати прийняття рішень на основі зібраних даних, що сприятиме автоматизації та покращенню функціональності системи;

Отже, актуальність даної теми впливає з необхідності досягнення більшої ефективності, безпеки та комфорту у розумних будинках, а також з потенціалу нейронних мереж у досягненні цих цілей. Дослідження в галузі розробки нейронної мережі для керування системою розумного будинку може привести до нових інновацій та покращень у цій сфері.

2 ДОСЛІДЖЕННЯ В СФЕРІ РОЗУМНИХ БУДИНКІВ

2.1 Поняття розумного будинку

Розумний будинок – це комплексна система, яка об'єднує різні компоненти та пристрої для забезпечення автоматизації та управління будинком. У розумному будинку центральною точкою керування є централізована система, яка об'єднує різні компоненти та пристрої. Ця система може включати в себе центральний контролер, сенсори, виключні пристрої, системи безпеки та багато іншого. Також може включати в себе різні системи, такі як системи освітлення, опалення, безпеки, мультимедіа тощо. Взаємозв'язок між цими системами дозволяє їх спільну роботу та взаємодію для забезпечення комфорту та безпеки мешканців. Наприклад, система розумного будинку може автоматично вимикати світло, коли мешканець залишає кімнату, або забезпечувати синхронізацію музики та відео між різними пристроями. Взаємозв'язок між цими компонентами забезпечує їх спільну роботу та взаємодію.

Для забезпечення зв'язку між компонентами розумного будинку використовуються бездротові технології, такі як Wi-Fi, Bluetooth, Zigbee тощо. Ці технології дозволяють компонентам спілкуватися між собою без необхідності фізичних з'єднань, що забезпечує більшу гнучкість та зручність.

У сучасному світі розумні будинки стають все більш популярними, оскільки вони пропонують розширені можливості з управління та контролю різних аспектів життя.

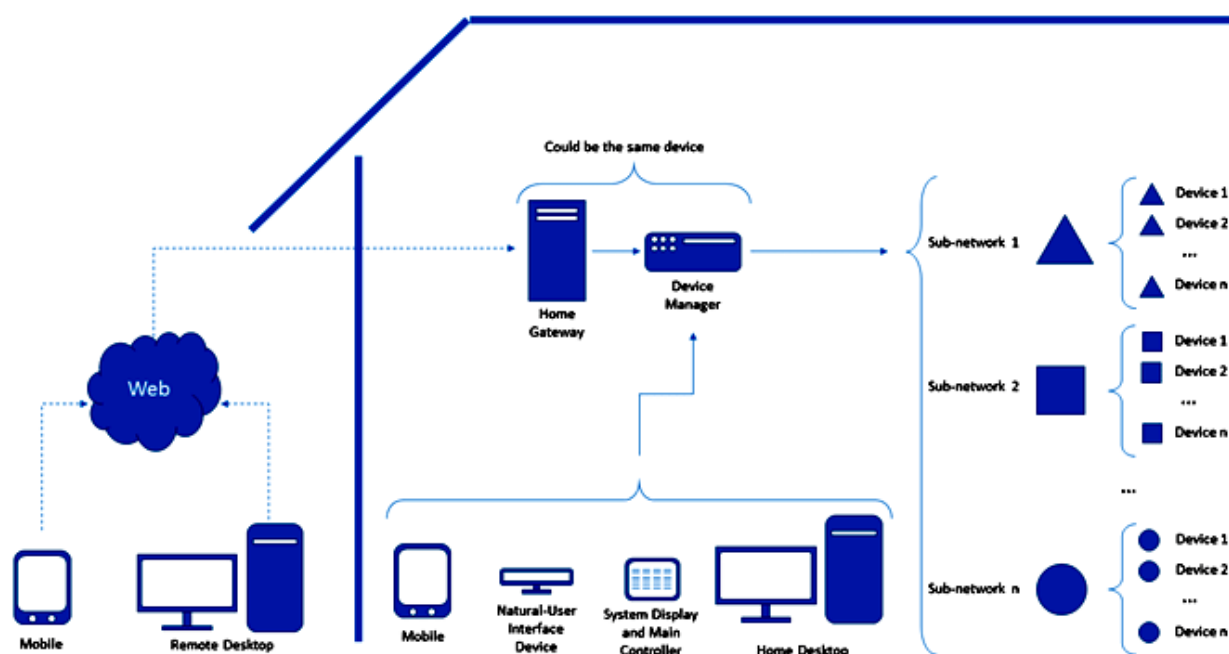


Рисунок 2.1 – Архітектура системи розумного будинку

Рисунок 2.1 представляє архітектуру системи розумного будинку, в якій зображені основні компоненти та їх взаємозв'язок.

Одним з ключових аспектів досліджень в сфері розумних будинків є аналіз функціональних можливостей цих систем. В статті «The Evolution Of Smart Home Technology» [3] проведений огляд еволюції розумних будинків з минулого до сьогодення, а також розглянуто потенційні перспективи розвитку. Відзначено, що розумні будинки здатні автоматизувати багато аспектів життя, такі як освітлення, безпека, енергоспоживання, кліматичні умови тощо. Це дозволяє забезпечити комфорт, безпеку та ефективність управління.

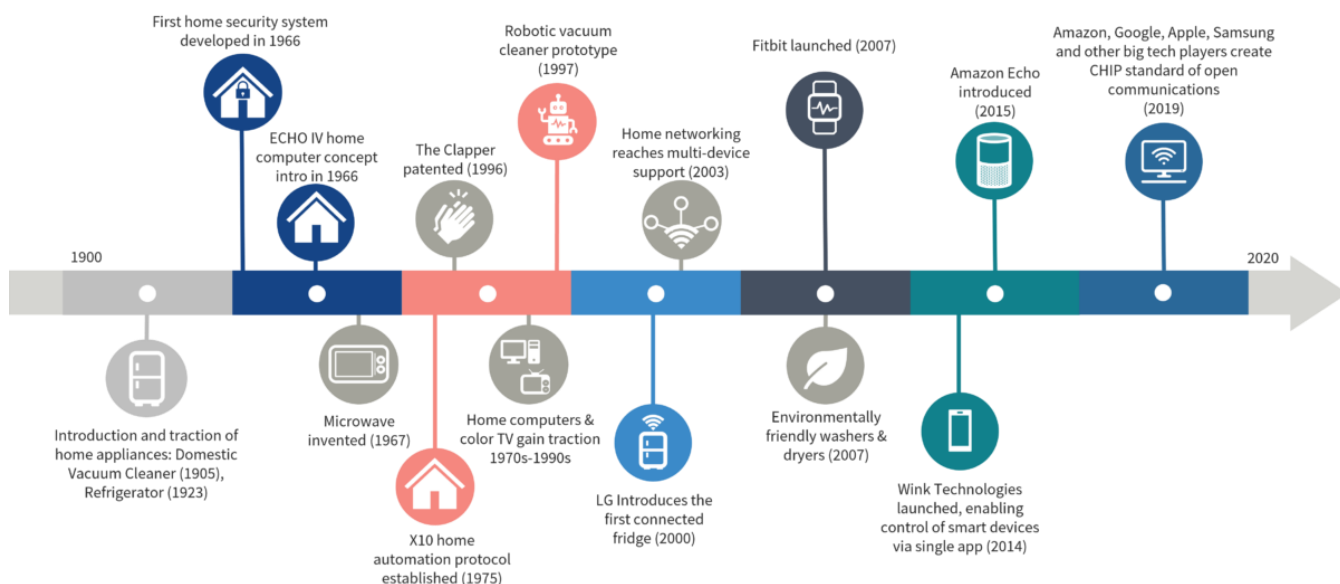


Рисунок 2.2 – Еволюція розумних будинків

Рисунок 2.2 демонструє еволюцію розумних будинків з минулого до сьогодення. Зображення відображає зміну управління та контролю різних аспектів життя від традиційних систем до сучасних розумних будинків.

2.2 Стандарти та протоколи для розумних будинків

Одним з ключових аспектів розумних будинків є забезпечення зв'язку та взаємодії між пристроями. В цьому контексті розглянуті різні протоколи та стандарти, що використовуються в системах розумних будинків. Протокол – це спосіб передачі сигналу про ту чи іншу дію від одного пристрою до іншого. Наприклад, один пристрій розпізнає голосову команду і наказує іншому увімкнути або вимкнути світло. При виборі пристроїв для автоматизованої екосистеми важливо розуміти різницю між протоколами, оскільки кожен із них має свої плюси і мінуси залежно від мети використання. Протоколи дозволяють забезпечити надійну та безпечну взаємодію між пристроями розумного будинку, а також підтримують сумісність з різними виробниками. Деякі приклади:

1) MQTT (Message Queuing Telemetry Transport) є легким протоколом передачі повідомлень, який дозволяє ефективно обмінюватись даними між пристроями у розумному будинку. Він забезпечує надійну доставку повідомлень та можливість підписки на різні теми;

2) RESTful API є архітектурним стилем для створення веб-сервісів, що може бути використаний для взаємодії з пристроями розумного будинку через Інтернет. Він дозволяє здійснювати операції зчитування та запису даних з різних пристроїв;

3) CoAP (Constrained Application Protocol) є протоколом передачі даних, спеціально розробленим для пристроїв з обмеженими ресурсами, таких як датчики в розумних будинках. Він забезпечує ефективну передачу даних за низькими вимогами до мережевих ресурсів;

Ці групи можуть бути «відкритими», і тоді їх протоколами може скористатися широке коло осіб. Відмінним прикладом такого протоколу є KNX та Z-wave.

В інших випадках групи «закриті», і тоді вони дозволяють використовувати їхній протокол лише певному виробнику або декільком на основі придбаної ними ліцензії на використання. Як приклад такого випадку можна назвати Zigbee.

Zigbee є одним з популярних стандартів бездротової мережі, який використовується для з'єднання різних пристроїв у розумному будинку. Він забезпечує низьке споживання енергії та можливість взаємодії між пристроями різних виробників.

Z-Wave є ще одним популярним стандартом для розумних будинків. Він працює на основі радіочастоти і забезпечує високу надійність та широкий діапазон покриття.

Wi-Fi є широко використовуваним стандартом для бездротового з'єднання пристроїв у будинку. Він забезпечує швидку передачу даних, але може бути більш енергозатратним порівняно з іншими стандартами.

Таблиця 2.1 – Порівняння різних технологій та рішень у сфері розумних будинків

Технологія	Опис	Переваги	Недоліки
ZigBee	Бездротовий протокол зв'язку	Енергоефективність	Обмежений дальній зв'язок
Z-Wave	Бездротовий протокол зв'язку	Широкий дальній зв'язок	Висока вартість
Wi-Fi	Бездротовий протокол зв'язку	Висока швидкість передачі	Велике споживання енергії

2.3 Огляд готових рішень для розумного будинку

Сфера розумних будинків знаходиться на шляху активного розвитку, і на сьогоднішній день існує велика кількість рішень та технологій, які можуть бути використані для створення розумних будинків.

Системи автоматизації будинку (Home Automation Systems) включають в себе різні технології та пристрої для управління освітленням, опаленням, кондиціонуванням повітря, безпекою, розвагами та іншими аспектами будинку. Популярні системи включають Apple HomeKit, Google Home та Amazon Echo. Вони забезпечують інтеграцію зі смарт-пристроями та дозволяють управляти будинком за допомогою голосових команд або мобільних додатків. Порівнюючи ці системи, можна враховувати їх сумісність з різними пристроями, зручність використання та наявність додаткових функцій.

Мережі зв'язку (Communication Networks) відіграють важливу роль у розумних будинках, оскільки вони забезпечують зв'язок між різними пристроями та системами. Одним з популярних варіантів є Wi-Fi, який забезпечує широкий охоплення та високу швидкість передачі даних.

Інші варіанти включають Bluetooth, Zigbee, Z-Wave та інші бездротові та проводові протоколи. Порівнюючи ці мережі, важливо враховувати їх стійкість до перешкод, швидкість передачі даних, вартість та можливість інтеграції з іншими пристроями.

Системи безпеки (Security Systems) включають в себе відеоспостереження, датчики руху, датчики витoku газу та вуглекислого газу, датчики протиопікання та багато інших пристроїв, які забезпечують захист будинку та мешканців. Популярні системи включають Nest Secure, Ring Alarm та SimpliSafe. Порівнюючи ці системи, важливо враховувати їх точність датчиків, можливість інтеграції з іншими пристроями, можливість сповіщення та доступ до системи віддалено.

Енергоефективність (Energy Efficiency) є важливим аспектом розумного будинку, і існують рішення та технології, спрямовані на зниження споживання енергії та оптимізацію її використання. Це включає в себе автоматичне регулювання освітлення та температури, використання енергоефективних пристроїв та систем управління енергією. Порівнюючи ці рішення, важливо враховувати їх енергоефективність, можливість налаштування та зручність використання.

Враховуючи розглянуті рішення та технології, можна зробити висновок, що в сфері розумних будинків існує широкий спектр готових рішень для автоматизації та управління різними аспектами будинку.

При виборі конкретної системи важливо враховувати потреби та вимоги користувачів, сумісність з іншими пристроями, доступність додаткових функцій та вартість.

Таблиця 2.2 – Порівняння деяких популярних систем з різних аспектів

Система	Автоматизація	Освітлення	Температура	Система безпеки	Енергоефективність
Amazon Echo	Так	Так	Так	Ні	Так

Google Home	Так	Так	Так	Ні	Так
Apple HomeKit	Так	Так	Так	Так	Так
Samsung SmartThings	Так	Так	Так	Так	Так
Philips Hue	Так	Так	Ні	Ні	Так
Nest	Так	Ні	Так	Так	Так

Таблиця 2.2 надає загальний огляд деяких важливих функцій рішень для розумних будинків.

Дослідження в сфері розумних будинків надають важливу інформацію щодо поточного стану цієї області та пропонують підґрунтя для подальших досліджень. Враховуючи результати вищезгаданих протоколів, можна визначити напрямки подальшої роботи, спрямовані на поліпшення функціональності та ефективності систем розумних будинків.

2.3.1 Google Home та Google Nest

Google Home та Google Nest є популярними платформами для систем розумного будинку, які використовують нейронні мережі для автоматизації та керування пристроями. Ці системи надають користувачам можливість керувати освітленням, термостатами, безпековими системами та іншими пристроями з допомогою голосового управління. Вивчення принципів роботи цих систем може надихнути на створення подібної системи розумного будинку.

Основними функціями цих систем є:

- голосове управління: Вони дозволяють користувачам керувати пристроями та системами за допомогою голосових команд. Це робить процес автоматизації більш доступним та зручним;
- освітлення та клімат-контроль: Користувачі можуть налаштовувати

освітлення вдома та регулювати температуру за допомогою цих платформ. Це дозволяє зекономити енергію та створити комфортні умови у приміщенні;

– безпека: Google Nest також пропонує рішення для безпеки будинку, включаючи відеоспостереження та датчики безпеки. Це дозволяє користувачам віддалено контролювати та моніторити стан свого будинку;

– інтеграція з іншими пристроями: Ці платформи підтримують інтеграцію з іншими розумними пристроями та сервісами, що розширює їх функціональність.

2.3.2 Amazon Alexa

Amazon Echo та голосовий асистент Alexa – інша популярна система для систем розумного будинку. Alexa використовує нейронні мережі для розпізнавання голосових команд користувачів та виконання відповідних дій, таких як управління освітленням, медіа та іншими пристроями. Вивчення роботи Alexa може сприяти розумінню того, як інтегрувати нейронні мережі для голосового управління в системі розумного будинку.

2.3.3 Apple HomeKit

Apple HomeKit – це платформа для систем розумного будинку, розроблена компанією Apple. Вона використовується для налаштування, комунікації та керування розумними пристроями вдома з використанням пристроїв Apple. HomeKit надає користувачам можливість автоматичного виявлення таких пристроїв і налаштування їх взаємодії.

Користувачі можуть створювати кімнати, об'єкти та дії в рамках системи HomeKit, що дозволяє активувати автоматизацію вдома за допомогою голосових команд до Siri або через додаток Home. Для розробників HomeKit надає можливість створювати складні додатки для керування аксесуарами на високому рівні.

HomeKit був вперше представлений на iPhone та iPad у версії iOS 8 17 вересня 2014 року. Ця платформа дозволила стороннім додаткам взаємодіяти з пристроями HomeKit за допомогою Siri та надавати віддалений доступ через центральні хаби вдома.

HomeKit був створений з метою спростити задачі вдома та забезпечити користувачам інструменти для адаптації різних можливостей будинку до їхніх конкретних потреб. Управління підключеними пристроями вдома здійснюється через протокол HomeKit Accessory Protocol (HAP). HomeKit надсилає повідомлення на підключені пристрої, які містять поля, що ідентифікують конкретний аксесуар та його категорію. Кожна категорія має код, який використовується для ідентифікації пристрою, а також ідентифікується за номером Global State Number (GSN). Як і у більшості пристроїв Apple, використовуються протоколи Apple Continuity Protocols, які включають бездротові технології, такі як Bluetooth і Wi-Fi.

HomeKit використовує протоколи Bluetooth та Wi-Fi, а також Thread на HomePod і деяких моделях Apple TV для підключення та комунікації з пристроями. Виробники пристроїв, сумісних з HomeKit, повинні брати участь у програмі MFi (Made for iPhone/iPad). Apple також надавала документацію для створення аксесуарів HomeKit для членів програми розробників Apple.

Початково всі продукти на основі HomeKit повинні були включати обов'язковий шифруючий копроцесор. Проте вимога була змінена у iOS 11, яке додало підтримку шифрування на програмному рівні. Обладнання, вироблене без підтримки HomeKit, може бути активоване для використання через "шлюзовий" продукт, такий як місто, який підключає ці пристрої до служби HomeKit.

HomeKit в основному конкурує зі стандартами розумних будинків від Amazon та Google. На жовтень 2019 року Apple перераховує 450 пристроїв, сумісних з HomeKit, порівняно з 10 000 для Google та 85 000 для Amazon.

2.3.4 OpenHAB

OpenHAB – це програмне рішення, розроблене на платформі Java, яке об'єднує компоненти для автоматизації будівель в єдину платформу незалежно від виробника пристроїв та протоколів використання. Ця система має велику спільноту розробників і базується на фреймворку Eclipse SmartHome.

OpenHAB 2 вдосконалено для зручності користувача під час налаштування та конфігурації пристроїв. В порівнянні з першою версією, в другому поколінні платформи було впроваджено численні покращення.

Це програмне рішення є незалежним від операційної системи та може бути легко розширене за допомогою додаткових прив'язок до різних технологій і протоколів. OpenHAB розроблено з відкритою ліцензією Eclipse і має відкритий вихідний код. Інтерфейси користувача підтримують веб-браузери, а також мобільні пристрої на платформах Android та Apple iOS.

Вивчення відкритого коду OpenHAB надає можливість користувачам розробляти власні системи розумного будинку з аналогічними функціями та можливостями інтеграції. Ця платформа дозволяє створювати різноманітні автоматизовані сценарії для керування будинком за допомогою зручного інтерфейсу.

2.3.5 Home Assistant

Система Home Assistant є інструментом для автоматизації та керування розумним будинком. Ця система володіє спектром можливостей, оскільки вона інтегрується з понад тисячею різноманітних пристроїв і сервісів, надаючи змогу контролювати багато аспектів побуту.

Після ініціалізації, Home Assistant автоматично проводить сканування домашньої мережі для визначення вже відомих пристроїв, що дозволяє легко налаштувати їх взаємодію з системою.

Після успішної інтеграції всіх пристроїв в вашому будинку, можна використовувати високорозвинений механізм автоматизації Home Assistant для створення складних сценаріїв, які роблять ваше життя більш комфортним і ефективним. Наприклад, налаштувати включення світла при заході сонця або при поверненні додому, або отримувати сповіщення, коли гаражна двері залишаються відкритими.

Крім того, Home Assistant надає можливість розширити функціональність за допомогою різних додатків. Можна встановлювати інші додатки, які допомагають у керуванні будинком, наприклад, AdGuard – DNS – базовий блокувальник реклами, або використовувати сторонні двигуни автоматизації, такі як NodeRed.

Однією з ключових переваг Home Assistant є те, що вся інформація та дані залишаються локальними. Система взаємодіє з пристроями локально і використовує хмарні сервіси лише як альтернативний варіант у випадку відсутності локального з'єднання. Це гарантує високий рівень конфіденційності і захищає дані користувачів від доступу сторонніх осіб.

3 ОСНОВНІ ПОНЯТТЯ НЕЙРОННОЇ МЕРЕЖІ

3.1 Нейронні мережі структура та принцип роботи

Нейронні мережі є ключовим інструментом у сфері штучного інтелекту та систем керування. Вони засновані на біологічних принципах роботи людського мозку і можуть моделювати складні нелінійні залежності між вхідними та вихідними даними [4].

3.1.1 Структура нейронної мережі

Нейронна мережа складається зі сполучених нейронів, що працюють разом для вирішення певного завдання. Кожен нейрон має свій вхідний та вихідний сигнали, і вони організовані в шари. Зазвичай нейронні мережі складаються з трьох основних типів шарів: вхідного шару, прихованих шарів і вихідного шару.

Вхідний шар отримує дані, які подаються на вхід мережі. Кількість нейронів у вхідному шарі відповідає кількості вхідних параметрів або ознак.

Приховані шари здійснюють обробку вхідних сигналів та виконують проміжні обчислення. Кількість прихованих шарів і кількість нейронів у кожному шарі можуть варіюватись залежно від складності задачі та потреб системи.

Вихідний шар надає результати роботи нейронної мережі. Кількість нейронів у вихідному шарі залежить від типу вихідної інформації, яку необхідно отримати.

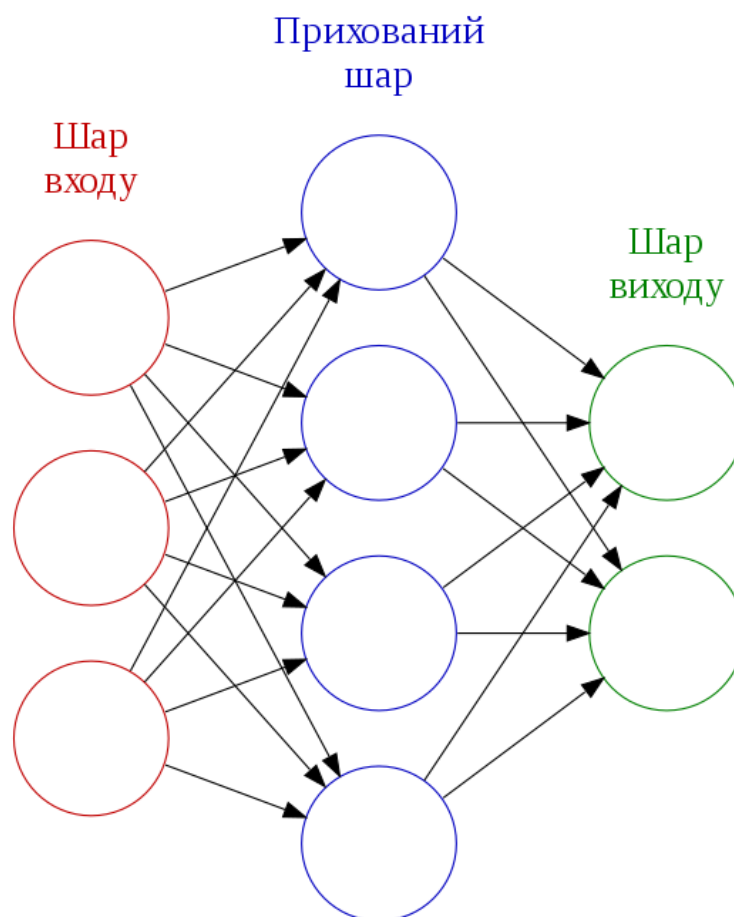


Рисунок 3.1 – Структура нейронної мережі

Структура нейронної мережі може бути різною в залежності від типу мережі та вирішуваної задачі. Існують такі типи нейронних мереж, як багат шарові перцептрони, зворотні поширення, рекурентні мережі, сверточні мережі та інші. Кожен тип мережі має свою специфіку та застосування.

3.1.2 Принцип роботи нейронної мережі

Нейронна мережа працює за принципом передачі сигналів через нейрони та обчислення вагових коефіцієнтів, що контролюють сили зв'язків між нейронами. Процес роботи нейронної мережі можна поділити на дві

фази: пряме поширення та зворотне поширення.

Пряме поширення. У цій фазі вхідні дані проходять через нейрони мережі від вхідного шару до вихідного шару. Кожен нейрон обчислює вагову суму вхідних сигналів, застосовує активаційну функцію та передає вихідний сигнал наступному шару. Цей процес повторюється до досягнення вихідного шару, де отримується фінальний результат.

Зворотне поширення. Після прямого поширення проводиться оцінка отриманого результату та коригування вагових коефіцієнтів для покращення точності. У цьому процесі використовується алгоритм зворотного поширення помилки, де розраховується помилка між отриманим результатом і очікуваним результатом. Потім ця помилка поширюється від вихідного шару до вхідного шару, коригуючи вагові коефіцієнти для зменшення помилки.

Таким чином, нейронні мережі використовуються для моделювання складних залежностей між вхідними та вихідними даними. Їх структура та принцип роботи дозволяють розв'язувати різноманітні задачі у багатьох галузях, включаючи системи керування розумними будинками.

3.2 Типи нейронних мереж

У сучасному світі нейронні мережі є одним із ключових інструментів у сфері машинного навчання та штучного інтелекту. Вони здатні до розв'язання складних задач, що потребують аналізу та взаємозв'язку великої кількості даних. У цьому пункті ми розглянемо різні типи нейронних мереж, зокрема перцептрон, зворотне поширення та рекурентні мережі, їх особливості та застосування.

Таблиця 3.1 – Порівняння типів нейронних мереж

Тип мережі	Опис
Багатошаровий	Складається з одного вхідного шару, одного або

перцептрон	декількох прихованих шарів та вихідного шару. Використовується для класифікації та регресії.
Зворотнє поширення	Використовується для навчання нейронної мережі за допомогою алгоритму зворотного поширення помилки. Використовується у багатьох задачах.
Рекурентна мережа	Має зв'язки, які створюють зациклену структуру та можуть зберігати інформацію про попередні стани. Використовується для обробки послідовностей.

3.2.1 Перцептрон

Перцептрон є одним з найпростіших типів нейронних мереж, але водночас він є основним будівельним блоком для більш складних архітектур. Вперше запропонований Френком Розенблаттом в 1957 році, перцептрон є моделлю штучного нейрона, який приймає на вхід декілька значень та генерує вихідний сигнал на основі внутрішньої ваги та функції активації. Його структура включає вхідний шар, приховані шари та вихідний шар.

Перцептрон здатний розв'язувати задачі класифікації, де вхідні дані можна подати у вигляді числових ознак. Наприклад, він може класифікувати зображення на основі піксельних значень або розпізнавати голосові команди на основі характеристик звукового сигналу. Перцептрон використовує функцію активації для прийняття рішення про активацію або пригнічення нейрона.

3.2.2 Зворотнє поширення

Зворотнє поширення є одним з найпоширеніших методів навчання нейронних мереж, особливо у глибоких нейронних мережах. Цей метод базується на принципі градієнтного спуску і дозволяє нейронній мережі

самостійно коригувати свої ваги для досягнення бажаного вихідного сигналу.

Процес зворотного поширення полягає в подачі вхідного сигналу до мережі, отриманні вихідного сигналу та порівнянні його з очікуваним вихідним сигналом. Далі обчислюється помилка (різниця між очікуваним та отриманим сигналом) та виконується алгоритм градієнтного спуску для зміни ваг нейронів таким чином, щоб зменшити помилку. Цей процес повторюється кілька разів з використанням тренувального набору даних, поки помилка не досягне заданого порогу або збурення.

Зворотнє поширення є потужним інструментом, оскільки воно дозволяє нейронній мережі самостійно вивчати складні залежності у вхідних даних. Воно застосовується в багатьох областях, включаючи комп'ютерне зорове сприйняття, мовний аналіз, рекомендаційні системи та багато інших.

3.2.3 Рекурентні мережі

Рекурентні мережі є спеціальним типом нейронних мереж, які мають здатність зберігати та використовувати інформацію з попередніх кроків часу. Це дає їм здатність моделювати послідовності та взаємозв'язки між елементами послідовності.

Структура рекурентної мережі включає вхідний шар, прихований шар та вихідний шар. Головною особливістю рекурентних мереж є наявність зворотного зв'язку, що дозволяє передавати інформацію назад у часі. Це досягається за допомогою рекурентних зв'язків, які дозволяють передавати вихідний сигнал попереднього кроку часу в якості вхідного сигналу наступному кроку.

Рекурентні мережі мають широке застосування у великій кількості задач, зокрема у мовному моделюванні, машинному перекладі, аналізі текстів та генерації послідовностей. Вони можуть ефективно моделювати залежності між словами у реченні, послідовностями акцій у часі та іншими динамічними структурами даних.

3.2.4 Застосування типів нейронних мереж

Перцептрони, зворотнє поширення та рекурентні мережі використовуються в багатьох областях, таких як комп'ютерне бачення, природна мова, рекомендаційні системи, автономні автомобілі та багато інших. Наприклад, в комп'ютерному баченні перцептрони та глибокі нейронні мережі здатні до розпізнавання об'єктів на зображеннях, виявлення облич та інших характеристик зображень.

У природному мовознавстві рекурентні мережі використовуються для машинного перекладу, аналізу настрою текстів, генерації тексту та багатьох інших завдань, пов'язаних з обробкою мови. В рекомендаційних системах нейронні мережі можуть моделювати смаки та персоналізовані преференції користувачів для надання рекомендацій на основі їхньої історії та взаємодії.

Використання різних типів нейронних мереж, таких як перцептрони, зворотнє поширення та рекурентні мережі, дозволяє нам моделювати складні залежності у вхідних даних та досягати високої точності у різних задачах машинного навчання.

Вони продовжують розвиватися та знаходити нові застосування у багатьох сферах, що допомагає зробити великі кроки в напрямку створення інтелектуальних систем, здатних аналізувати, розуміти та взаємодіяти з навколишнім світом.

3.3 Функції активації та їх вплив на роботу нейронних мереж

Функція активації визначає вихідний сигнал нейрона залежно від його вхідного сигналу. Коректний вибір функції активації впливає на швидкість навчання, стійкість моделі та здатність до апроксимації складних функцій.

3.3.1 Сигмоїдна функція

Однією з найпоширеніших функцій активації є сигмоїдна функція. Сигмоїдна функція має S-подібну форму та відображає значення у діапазоні між 0 та 1.

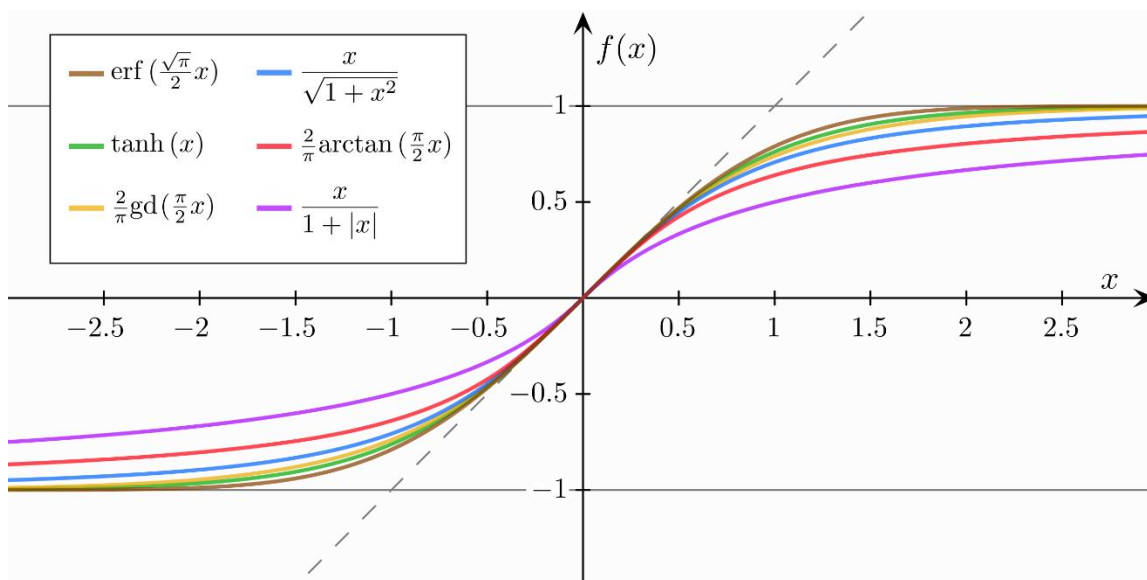


Рисунок 3.2 – Сигмоїдна функція

Вона використовується для моделювання нелінійних залежностей в нейронних мережах. Однак, у нейронних мережах глибокого навчання сигмоїдна функція поступово втрачає свою популярність через кілька обмежень. Зокрема, сигмоїдна функція може страждати від проблеми насичення, коли вхідні дані знаходяться у крайніх точках, де похідна функції стає дуже мала. Це призводить до зникнення градієнту та утруднює навчання моделі.

3.3.2 Функція ReLU

Ще одна популярна функція активації, яка широко використовується у нейронних мережах, є функція ReLU (Rectified Linear Unit). Функція ReLU

повертає 0 для від'ємних значень та саму вхідну величину для не від'ємних значень.

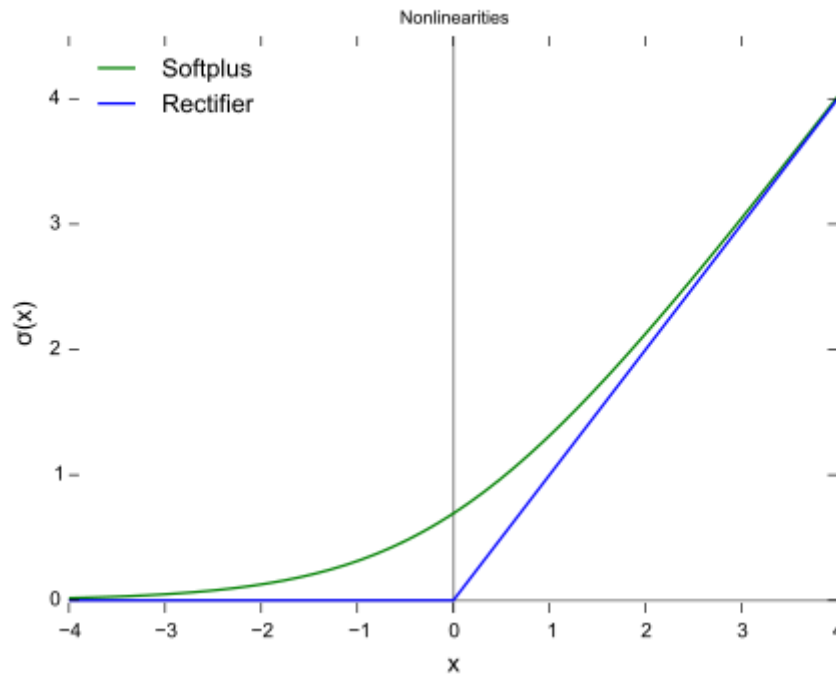


Рисунок 3.3 – Функція ReLU

Функція ReLU є лінійною на позитивній півосі і нелінійною на негативній півосі, що дозволяє їй моделювати складні нелінійні залежності. Її простота та ефективність робить її особливо корисною для глибоких нейронних мереж. Однак, функція ReLU має свої обмеження, зокрема, проблему «мертвих» нейронів. Якщо ваги нейрона настільки налаштовані, що вихідна активація завжди буде від'ємною, нейрон ніколи не активується і не приносить внесок у модель.

3.3.3 Функція гіперболічного тангенсу

Функція гіперболічного тангенсу (\tanh) є ще однією популярною функцією активації. Вона подібна до сигмоїдної функції, але відображає

значення у діапазоні від -1 до 1.

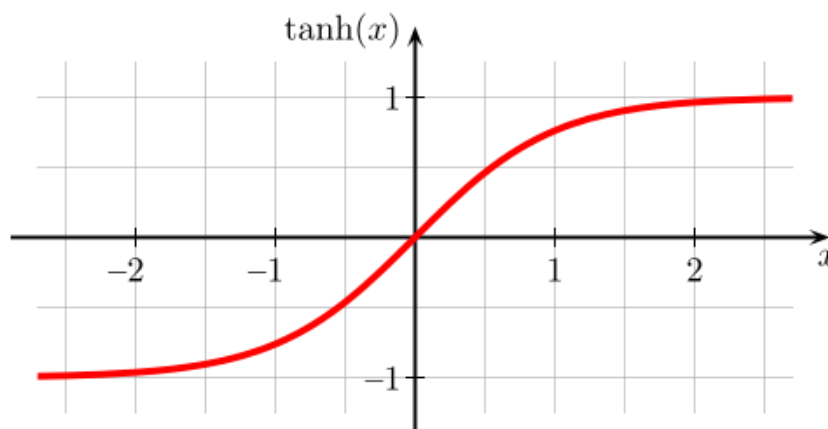


Рисунок 3.4 – Функція гіперболічного тангенсу

Функція \tanh є нелінійною та має симетричну форму. Так само як і сигмоїдна функція, вона може бути використана для моделювання нелінійних залежностей в нейронних мережах. Проте, вона також страждає від проблеми насичення, особливо коли вхідні дані знаходяться у крайніх точках.

3.3.4 Інші функції активації

Крім вищезгаданих функцій активації, існує також безліч інших, які можуть бути використані залежно від конкретних завдань та вимог моделі. Наприклад, функція Softmax часто використовується в задачах класифікації, де необхідно отримати ймовірності для кожного класу. Існують також функції активації, які були спеціально розроблені для вирішення конкретних проблем, наприклад, функція ELU (Exponential Linear Unit) або Leaky ReLU.

Коректний вибір функції активації важливий для успішної роботи нейронних мереж. Неможливо сказати, що одна функція активації підходить для всіх випадків. Вибір залежить від природи завдання, особливостей даних

та архітектури мережі. Часто вирішенням є використання комбінації різних функцій активації на різних шарах мережі.

3.4 Алгоритми навчання нейронних мереж

Алгоритми навчання нейронних мереж відіграють ключову роль у розвитку та використанні цих моделей. Оптимальний вибір алгоритму навчання може впливати на ефективність, швидкість та якість роботи нейронної мережі. У цьому розділі будуть розглянуті декілька популярних алгоритмів навчання нейронних мереж.

3.4.1 Алгоритм зворотнього поширення помилки

Одним з найпоширеніших алгоритмів навчання нейронних мереж є алгоритм зворотнього поширення помилки (Backpropagation). Цей алгоритм використовується для навчання нейронних мереж з учителем, де наявні пари вхідних даних та відповідних очікуваних вихідних значень. Мета полягає в тому, щоб мережа вчилася апроксимувати ці вихідні значення шляхом зміни ваг та зсувів нейронів.

Алгоритм зворотнього поширення помилки базується на градієнтному спуску, де використовується похідна функції втрати по вагах для оновлення їх значень у напрямку, що мінімізує втрату [5]. Процес навчання включає два основних кроки: пряме поширення сигналу (forward pass), коли вхідні дані проходять через мережу та визначаються вихідні значення, і зворотнє поширення помилки (backward pass), коли обчислюються градієнти втрати та ваги оновлюються.

Алгоритм зворотнього поширення помилки є потужним і ефективним способом навчання нейронних мереж, але він також має свої обмеження. Він може страждати від проблеми зникнення градієнту (vanishing gradient problem), коли градієнти занадто малі, що ускладнює процес навчання

глибоких мереж. Для подолання цієї проблеми використовуються модифікації алгоритму, такі як розповсюдження градієнту по часу (gradient propagation through time) для рекурентних мереж або методи нормалізації градієнту.

3.4.2 Алгоритмічні методи оптимізації

Окрім алгоритму зворотнього поширення помилки, існує також широкий спектр алгоритмічних методів оптимізації, які використовуються для навчання нейронних мереж. Ці методи базуються на різних принципах і можуть мати вплив на швидкість та якість збіжності мережі.

Один з таких методів – алгоритм стохастичного градієнтного спуску (Stochastic Gradient Descent, SGD). Він використовується для мінімізації функції втрати шляхом оновлення ваг з кожним навчальним прикладом. SGD є простим і ефективним методом, але він може мати проблему з швидкістю збіжності та може застрягати в локальних мінімумах.

Деякі покращені версії SGD включають метод імпульсу (Momentum), який надає прискорення процесу навчання шляхом накопичення швидкості оновлення ваг, та метод Адама (Adam), який комбінує ідеї імпульсу та адаптивного нормалізованого шкільного кроку (adaptive normalized learning rate).

3.4.3 Рекурентне навчання

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) є особливим типом нейронних мереж, які мають зв'язки між нейронами, що створюють циклічну структуру. Це дозволяє RNN зберігати інформацію про попередні стани та використовувати її для обробки нових вхідних даних.

Алгоритми навчання рекурентних мереж також відрізняються від алгоритмів для звичайних нейронних мереж. Одним з популярних алгоритмів

є алгоритм зворотнього поширення помилки через час (Backpropagation Through Time, BPTT), який розширює алгоритм зворотнього поширення помилки на часову вимірину рекурентними мережами. BPTT використовує той самий принцип оновлення ваг згортки градієнту, що й алгоритм зворотнього поширення помилки, але враховує залежності в часі.

3.4.4 Інші алгоритми навчання

Поміж інших алгоритмів навчання нейронних мереж варто відзначити генетичні алгоритми, які використовують еволюційний підхід до оптимізації ваг мережі, алгоритм відновлення нейронних мереж (NeuroEvolution of Augmenting Topologies, NEAT), що поєднує генетичні алгоритми з розростаючими структурами мережі, та алгоритм Хінтона (Hinton's Capsule Network), який використовує капсульні мережі для моделювання об'єктів у просторі.

Кожен з цих алгоритмів має свої переваги та обмеження, і вибір підходящого алгоритму навчання залежить від конкретних умов та вимог задачі. Дослідження в галузі алгоритмів навчання нейронних мереж постійно продовжуються з метою розробки нових методів, що забезпечують кращу збіжність, стійкість та швидкість навчання.

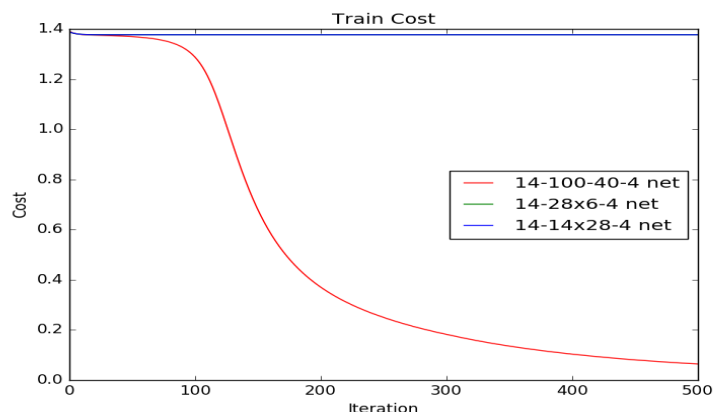


Рисунок 3.5 – Залежність функції втрати від ітерацій

Рисунок 3.5 демонструє графік залежності функції втрати від ітерацій під час навчання нейронної мережі з використанням алгоритму зворотнього поширення помилки. Видно, як втрата зменшується з кожною ітерацією, що свідчить про прогрес у навчанні моделі.

Таблиця 3.2 – Порівняння алгоритмів навчання нейронних мереж

Алгоритм	Опис
Зворотнє поширення помилки	Використовує градієнтний спуск для оновлення ваг мережі.
Стохастичний градієнтний спуск	Використовується для мінімізації функції втрати з кожним навчальним прикладом.
Адам	Комбінація методу імпульсу та адаптивного нормалізованого шкільного кроку.
Зворотнє поширення помилки через час	Розширення алгоритму зворотнього поширення помилки на рекурентні мережі.

4 ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У СИСТЕМАХ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ

4.1 Огляд сучасних рішень та технологій у сфері розумних будинків

Розумний будинок є одним з ключових напрямків розвитку сучасних технологій, який надає зручність, комфорт та ефективність у повсякденному житті людей. Він поєднує в собі різноманітні сенсори, пристрої та системи, які забезпечують автоматизоване керування різними аспектами життя в будинку, такими як освітлення, опалення, вентиляція, безпека, розподіл енергії та інші.

На сьогоднішній день існує велика кількість рішень та технологій, які використовуються у сфері розумних будинків. Деякі з них базуються на традиційних протоколах та системах керування, таких як протоколи зв'язку ZigBee, Z-Wave, Wi-Fi, а також системи автоматизації, які використовують мікроконтролери, реле та датчики для керування різними пристроями в будинку. Ці рішення забезпечують базовий функціонал розумного будинку та дозволяють контролювати його системи через мобільні додатки або спеціальні пульти.

Однак, з розвитком технологій і збільшенням обсягу доступної інформації, з'являються більш продуктивні та прогресивні рішення. Одним з них є використання нейронних мереж у системах керування розумним будинком. Нейронні мережі є потужним інструментом машинного навчання, який може аналізувати та використовувати велику кількість даних для прийняття рішень. Вони дозволяють розумному будинку «вчитися» зі свого досвіду та адаптуватися до змінних потреб та умов.

Огляд сучасних рішень та технологій у сфері розумних будинків показує, що нейронні мережі знайшли своє застосування в різних аспектах керування будинком. Вони можуть бути використані для прогнозування

енергетичних витрат, оптимізації розподілу енергії, автоматичного керування освітленням та опаленням залежно від умов та пріоритетів користувачів.

Одним з важливих аспектів використання нейронних мереж у розумних будинках є їхні можливості адаптації та самонавчання. Вони можуть навчатися на основі історичних даних про поведінку користувачів та змінювати свої стратегії для досягнення оптимальних результатів. Наприклад, нейронна мережа може вивчати звички та пріоритети користувачів, а потім пропонувати оптимальні сценарії керування будинком відповідно до їхніх потреб.

Крім того, нейронні мережі можуть бути використані для розв'язання складних задач, таких як розпізнавання голосу, обличчя та жестів. Це дозволяє забезпечити більш особисту та зручну взаємодію користувача з розумним будинком. Наприклад, система розпізнавання голосу може розуміти команди користувача та виконувати їх безпосередньо.

Для ефективного застосування нейронних мереж у системах керування розумним будинком необхідно вирішити деякі проблеми, пов'язані з обробкою великих обсягів даних, навчанням моделей та ресурсоемністю обчислень. Проте, з розвитком технологій обчислювальної техніки та алгоритмів машинного навчання, ці проблеми стають все менш значущими.

Огляд сучасних рішень та технологій у сфері розумних будинків свідчить про постійний розвиток та вдосконалення цього напрямку. Нейронні мережі відкривають нові можливості для розумних будинків, забезпечуючи більш гнучке та інтелектуальне керування. Застосування цих технологій може покращити якість життя людей, знизити споживання енергії та забезпечити більшу безпеку та комфорт у їхніх домівках.

4.2 Потенційні області застосування нейронних мереж у керуванні розумним будинком

Застосування нейронних мереж у системах керування розумним

будинком відкриває широкі перспективи для покращення функціональності, зручності та ефективності таких систем. Нейронні мережі дозволяють створити адаптивні та самонавчаючі системи, які здатні аналізувати дані, розпізнавати закономірності та приймати рішення на основі зібраних даних.

Однією з потенційних областей застосування нейронних мереж у керуванні розумним будинком є оптимізація енергоспоживання. Нейронні мережі можуть аналізувати споживання енергії у будинку, враховувати зовнішні фактори (наприклад, погоду, добові ритми тощо) та прогнозувати оптимальний режим роботи систем опалення, кондиціонування повітря, освітлення та інших електричних пристроїв. Це дозволяє досягти енергоефективності та зниження витрат на енергію, забезпечуючи комфортні умови для мешканців.

Ще однією важливою областю є покращення безпеки та захисту в системах розумного будинку. Нейронні мережі можуть бути використані для розпізнавання образів, що дозволяє виявляти незвичну або підозрілу активність у приміщенні або на території будинку. Наприклад, нейронна мережа може розпізнавати образи людей, транспортних засобів чи навіть дій та поведінки мешканців, сповіщаючи про можливі загрози або надавати доступ до будинку лише авторизованим особам. Це забезпечує високий рівень безпеки та захисту в розумних будинках.

Додатково, нейронні мережі можуть забезпечувати інтелектуальне управління системами в будинку. Вони можуть виконувати завдання розпізнавання голосу або жестів, що дозволяє мешканцям комунікувати з системою управління без потреби використовувати класичні інтерфейси. Нейронні мережі можуть навчитися розпізнавати індивідуальні особливості мешканців та виконувати персоналізовані команди, наприклад, налаштувати температуру, освітлення або відтворити улюблену музику.

4.3 Використання нейронних мереж у системах керування

Використання нейронних мереж у системах керування є однією з передових областей досліджень в галузі розумних будинків. Нейронні мережі, які засновані на принципах функціонування нервової системи, можуть вирішувати складні задачі керування та досягати високої ефективності в управлінні розумними будинками. В цьому пункті розглянемо використання нейронних мереж у системах керування розумним будинком та їх потенційні переваги.

Нейронні мережі використовуються у різних аспектах систем керування розумними будинками, включаючи передбачуване керування, управління енергоефективністю, розпізнавання та ідентифікацію.

Таблиця 4.1 – Порівняння енергозберігаючих алгоритмів управління в розумному будинку

Алгоритм	Відсоток енергозбереження
Передбачуване керування	25%
Адаптивне освітлення	15%
Оптимізація режимів	30%

4.3.1 Передбачуване керування

Передбачуване керування є важливою складовою систем розумного будинку. Нейронні мережі можуть бути навчені передбачати зміни в параметрах будинку на основі вхідних сигналів, таких як температура, освітлення, вологість, рух і багато інших. Це дозволяє системі керування адаптувати свої режими роботи, оптимізувати використання ресурсів та

забезпечувати комфорт для мешканців будинку.

Наприклад, нейронна мережа може бути навчена передбачати зміни температури в будинку на основі вхідних сигналів, таких як час доби, погодні умови, кількість осіб в будинку тощо.

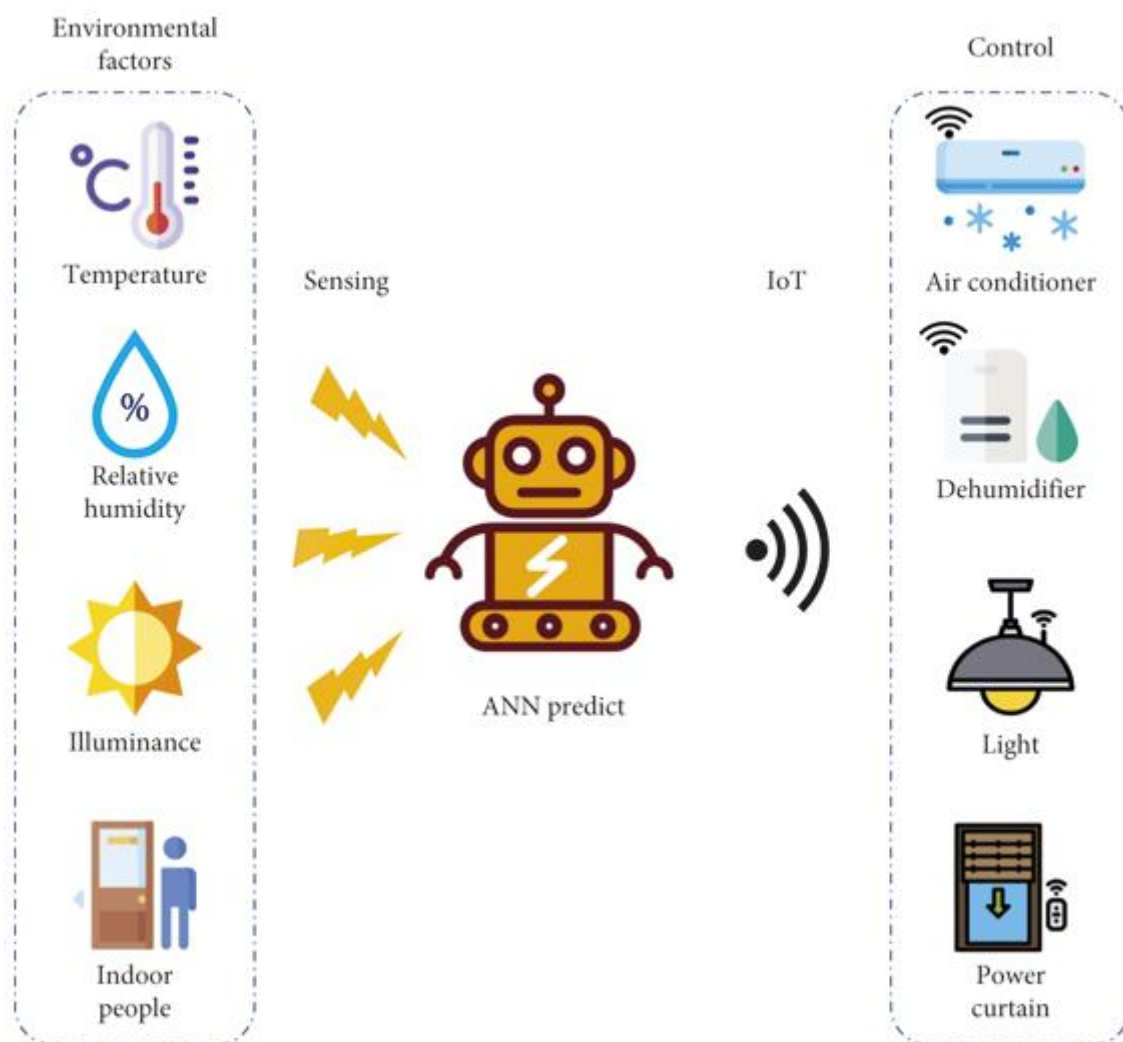


Рисунок 4.1 – Архітектура нейронної мережі для передбачуваного керування в розумному будинку

За допомогою таких передбачень система керування може налаштувати роботу систем опалення та кондиціонування повітря, забезпечуючи оптимальну температуру в будинку без зайвого споживання енергії.

Дослідження показали, що використання нейронних мереж для

передбачуваного керування може призвести до значного покращання енергоефективності розумних будинків та комфорту мешканців [1]. Результати досліджень свідчать про зниження споживання електроенергії на опалення та кондиціонування повітря на 20-30% при застосуванні нейронних мереж у системах передбачуваного керування.

4.3.2 Управління енергоефективністю

Однією з ключових проблем у розумних будинках є енергоефективність. Забезпечення оптимального використання енергії є важливим завданням для зменшення споживання електроенергії та вартості енергетичних ресурсів. В цьому контексті нейронні мережі можуть бути використані для розробки імплементації алгоритмів управління енергоефективністю розумних будинків.

Наприклад, нейронна мережа може бути навчена прогнозувати споживання електроенергії в будинку на основі різних факторів, таких як час доби, день тижня, активність мешканців та інші. За допомогою таких передбачень система керування може управляти енергозабезпеченням будинку, включаючи оптимальне використання альтернативних джерел енергії, регулювання режимів освітлення та електроприладів.

Дослідження показали, що використання нейронних мереж у системах управління енергоефективністю може призвести до значних економічних вигод, зменшення споживання електроенергії та викидів вуглецю. Застосування таких систем у реальних умовах дозволяє зменшити витрати на електроенергію до 30% та сприяє збереженню природних ресурсів.

4.3.3 Розпізнавання та ідентифікація

Ще однією важливою областю використання нейронних мереж у системах керування розумним будинком є розпізнавання та ідентифікація.

Нейронні мережі можуть бути навчені розпізнавати обличчя, голоси, жести та інші біометричні дані мешканців будинку для забезпечення безпеки та персоналізованого керування.

Наприклад, нейронна мережа може бути навчена розпізнавати обличчя мешканців будинку та автоматично відкривати двері, налаштовувати настрої освітлення та інших систем відповідно до їхніх персональних налаштувань. Це забезпечує зручність та безпеку для мешканців, а також можливість контролювати доступ до будинку.

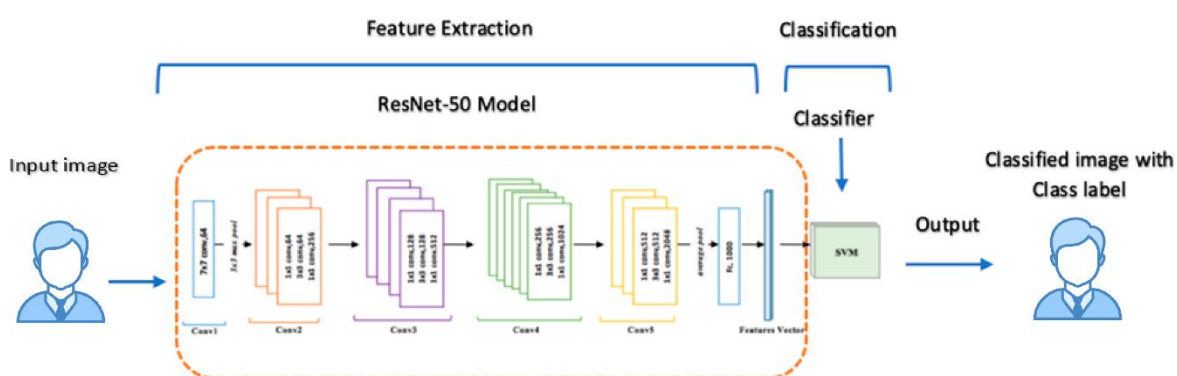


Рисунок 4.2 – Приклад структури нейронної мережі для ідентифікації

Дослідження в галузі розпізнавання та ідентифікації з використанням нейронних мереж продовжуються активно. Результати досліджень свідчать про високу точність та ефективність таких систем, що забезпечує високий рівень безпеки та персоналізованого сервісу.

Нейронні мережі здатні адаптуватися до змінних умов та самостійно навчатися на основі великих обсягів даних. Це робить їх ефективними інструментами для аналізу та керування розумним будинком. Одним з головних використань нейронних мереж у системах керування розумним будинком є аналіз даних з різних датчиків та пристроїв, що дозволяє розуміти поведінку мешканців та оптимізувати енергоефективність та комфортність приміщень.

Однією з ключових областей застосування нейронних мереж у

системах керування розумним будинком є розпізнавання образів та комп'ютерне зору. Нейронні мережі можуть використовуватися для автоматичного розпізнавання облич, виявлення руху та інших образів з відеопотоків, що дозволяє покращити безпеку та контроль за приміщеннями. Наприклад, система нейронних мереж може автоматично розпізнавати мешканців та запам'ятовувати їхні впізнавальні ознаки, що дозволяє забезпечити безпеку та автоматичне налаштування персоналізованих налаштувань для кожного мешканця.

Додатково, нейронні мережі можуть бути використані для розпізнавання голосу та природної мови, що дозволяє забезпечити голосове керування розумним будинком. Завдяки глибокому навчанню та обробці природної мови, система може розуміти команди та інструкції від мешканців та виконувати відповідні дії, такі як включення освітлення, регулювання температури або запуск побутових пристроїв.

Крім того, нейронні мережі можуть бути використані для прогнозування та управління енергоспоживанням розумного будинку. Вони можуть аналізувати дані про енергопостачання, зовнішній клімат, поведінку мешканців та інші фактори, щоб оптимізувати режими роботи різних пристроїв та систем, забезпечуючи енергоефективність та економію ресурсів.

Однак, використання нейронних мереж у системах керування розумним будинком також має свої обмеження. Перше за все, це вимагає значних обчислювальних ресурсів для тренування та роботи нейронних мереж. Такі системи потребують потужних процесорів та великої кількості пам'яті для ефективної роботи. Крім того, інтерпретація та пояснення рішень, прийнятих нейронними мережами, може бути складною задачею, що ставить виклик перед довірою до системи та її зрозумілістю для користувачів..

5 ПЕРЕВАГИ ТА ВИКЛИКИ РЕАЛІЗАЦІЇ СИСТЕМ РОЗУМНОГО БУДИНКУ НА БАЗІ НЕЙРОННИХ МЕРЕЖ

5.1 Адаптивність та самонавчання

Нейронні мережі можуть самостійно навчатися та адаптуватися до змінних умов і вимог системи розумного будинку. Вони здатні виявляти складні залежності та патерни в даних, що дозволяє їм ефективно адаптуватися до змін у споживанні електроенергії, оптимізувати роботу систем опалення та кондиціонування повітря, а також прогнозувати зрушення в споживанні енергії.

5.2 Гнучкість і масштабованість

Нейронні мережі можуть бути гнучкими і легко масштабуватись, що дозволяє їх використовувати для керування різними аспектами системи розумного будинку. Вони можуть працювати з великим обсягом даних та виконувати складні завдання керування з багатьма параметрами.

5.3 Здатність до вирішення складних проблем

Нейронні мережі мають потенціал для розв'язання складних проблем у сфері розумних будинків. Вони можуть аналізувати велику кількість даних з різних датчиків та пристроїв, виявляти аномальну поведінку та забезпечувати вчасну реакцію на виникнення проблем.

5.4 Споживана потужність та енергоефективність

Однією з ключових проблем у реалізації систем розумного будинку на

базі нейронних мереж є споживана потужність та енергоефективність. Нейронні мережі можуть вимагати значних обчислювальних ресурсів, що призводить до збільшення споживаної енергії. Це особливо важливо в системах розумного будинку, оскільки пристрої мають працювати безперервно.

5.5 Безпека та конфіденційність даних

Іншою важливою проблемою є безпека та конфіденційність даних користувачів системи розумного будинку. Використання нейронних мереж може вимагати обробки чутливої інформації, такої як відео або аудіопотоки, голосові команди користувачів та інші особисті дані. Забезпечення адекватного рівня захисту та конфіденційності даних стає важливою задачею в реалізації систем розумного будинку.

5.6 Інтеграція різних пристроїв та стандартів

Системи розумного будинку часто використовують різні пристрої від різних виробників з різними стандартами та протоколами зв'язку. Інтеграція всіх цих пристроїв у єдину систему може бути складною задачею, особливо коли вони використовують різні мови та формати даних. Н адодачу налаштування нейронних мереж вимагає експертних знань та досвіду. Правильне визначення архітектури мережі, підбір оптимальних гіперпараметрів та навчання моделі можуть бути складними завданнями, особливо для неспеціалістів у галузі машинного навчання

5.7 Доступність та вартість

Доступність та вартість систем розумного будинку на базі нейронних мереж може бути обмеженою для деяких користувачів. Нейронні мережі

можуть вимагати високотехнологічного обладнання та програмного забезпечення, що може бути дорогим. Зробити системи розумного будинку більш доступними та вартісно-ефективними є важливою задачею.

5.8 Надійність та скасування помилок

В системах розумного будинку, де нейронні мережі відповідають за прийняття рішень, важливо забезпечити надійність та можливість скасування помилок. Неправильне розпізнавання команд або помилки в роботі можуть призвести до негативних наслідків, включаючи втрату даних або некоректну роботу пристроїв.

Реалізація систем розумного будинку на базі нейронних мереж стикається з численними технічними, безпековими та практичними викликами. Для успішної розробки таких систем необхідно ретельно вивчити ці проблеми та розробити відповідні стратегії та рішення.

5.9 Потреба в великій кількості даних

Нейронні мережі потребують значної кількості даних для ефективного навчання та прогнозування. Вони можуть бути менш ефективними у випадках, коли обмежений обсяг доступних даних або коли дані мають високий рівень шуму або неоднорідностей

5.10 Інтерпретованість результатів

Нейронні мережі, особливо з високою складністю, можуть бути менш інтерпретованими з точки зору пояснення прийнятих рішень. Вони можуть бути розглянуті як «чорні скриньки», де важко визначити, як саме було прийняте рішення.

Таблиця 4.2 – Переваги та обмеження використання нейронних мереж у системах керування розумним будинком

Переваги	Обмеження
Самонавчання	Вимоги до обчислювальної потужності
Адаптація до змінних умов	Складність пояснення рішень
Аналіз великих обсягів даних	Етичні аспекти використання

Таким чином, використання нейронних мереж у системах керування розумним будинком є потужним інструментом, який дозволяє розв'язувати складні завдання аналізу та управління. Однак, необхідно ретельно розглядати їх переваги та обмеження, а також забезпечувати захист приватності та етичне використання даних. Розумний підхід до використання нейронних мереж у системах розумних будинків може відкрити нові можливості для створення інноваційних та ефективних рішень, що покращують якість життя мешканців

6 РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ОЦІНКИ СЕРЕДОВИЩА

6.1 Нейронна мережа для аналізу стану середовища.

Нейронні мережі є математичними моделями, які імітують роботу нейронів у мозку людини. Вони складаються з нейронів, які з'єднані між собою в шарах та взаємодіють один з одним. Кожен нейрон приймає вхідні сигнали, обчислює їхню вагу та передає результат до інших нейронів. Нейрони реагують на вхідні сигнали, видаючи вихідні значення. Мережа для класифікації виходячи із назви виконує функцію аналізу певного набору даних та асоціацією з певною групою

6.1.1 Застосування нейронних мереж для класифікації

В контексті систем розумного будинку, нейронні мережі можуть бути використані для класифікації даних та прийняття відповідних рішень. Наприклад, система розумного будинку може класифікувати зчитані датчиками дані про стан приміщення (температура, вологість, освітлення) та виконувати відповідні дії, такі як регулювання температури або вмикання світла.

6.1.2 Огляд архітектур нейронних мереж для класифікації

Для класифікації існують різні архітектури нейронних мереж, які можуть бути використані в системах розумного будинку. Деякі з них включають:

– звичайні нейронні мережі (Feedforward Neural Networks, FFNN): Це базова архітектура нейронних мереж, в якій дані проходять через шари нейронів з одного кінця в інший без зворотного зв'язку. Вони

використовуються для бінарної та багатокласової класифікації;

- згорткові нейронні мережі (Convolutional Neural Networks, CNN): Ця архітектура підходить для обробки великих обсягів візуальних даних, таких як зображення. Вони використовуються для завдань, де важлива просторова структура даних, наприклад, в розпізнаванні об'єктів або відеоспостереженні;

- рекурентні нейронні мережі (Recurrent Neural Networks, RNN): Ці мережі підходять для обробки послідовних даних, таких як текст або часові ряди. Вони здатні враховувати контекст та залежності між даними на різних часових кроках;

- довго-короткострокова пам'ять (Long Short-Term Memory, LSTM): Ця архітектура є варіантом RNN і особливо ефективна в уникненні проблеми зникнення та зростання градієнту. Вона широко використовується для обробки послідовних даних та тексту.

6.1.3 Процес навчання

Процес навчання нейронної мережі для класифікації включає в себе наступні кроки:

- підготовка даних. Вхідні дані, які будуть використані для навчання та тестування, повинні бути підготовлені та нормалізовані;

- створення архітектури мережі. Визначення кількості шарів та нейронів у кожному шарі, а також вибір активаційних функцій;

- компіляція моделі. Встановлення параметрів навчання, таких як функція втрат та оптимізатор;

- навчання моделі. Модель навчається на тренувальних даних, де ваги нейронів оновлюються для мінімізації функції втрат;

- оцінка моделі. Модель оцінюється на тестових даних для визначення її точності та продуктивності.

Після навчання модель може бути використана для класифікації нових даних. Вхідні дані проходять через модель, і вихід вказує на клас або

категорію, до якої належить вхід.

6.2 Реалізація нейронної мережі для аналізу стану приміщення

Дана система виконує лише одну функцію – оцінка загального стану приміщення за показниками освітленості, температури вологості та часовому проміжку.

6.2.1 Збір даних

Для навчання такої мережі початковим кроком є формування датасету. Він має мати вигляд набору вхідних значень та очікуваного результату. В якості вхідних значень використовуємо показники датчиків системи Temperature, Humidity, Illumination, PartOfTheDay. В якості вихідного значення використано булеве значення IsActionRequired. Такий датасет має налічувати велику кількість тестових даних. Тому більшість із них згенеровано за певними правилами.

Лістинг 6.1 – Фрагмент коду для генерації датасету за показниками температури

```
private static void GenerateTemperatureDataSet(int rowCount, StreamWriter writer)
{
    for (int i = 0; i < rowCount; i++)
    {
        int illumination = GetNormalRandomIllumination();
        double humidity = GetNormalRandomHumidity();
        PartOfDay partOfDay = GetRandomPartOfDay();
        DesiredAction desiredAction = DesiredAction.None;

        double temperature = GetRandomTemperature();
        if (humidity > MAX_NORMAL_TEMPERATURE)
        {
            desiredAction = DesiredAction.DecreaseTemperature;
        }
        if (humidity < MIN_NORMAL_TEMPERATURE)
        {
            desiredAction = DesiredAction.IncreaseTemperature;
        }

        writer.WriteLine($">{temperature:F2},{humidity:F2},{illumination},{(int)partOfDay},{(int)desiredAction}>");
    }
}
```

Для достовірності правильно сформованого датасету варто відобразити набір даних на графіку, що допоможе візуально побачити тенденцію залежності значення від вхідних параметрів.

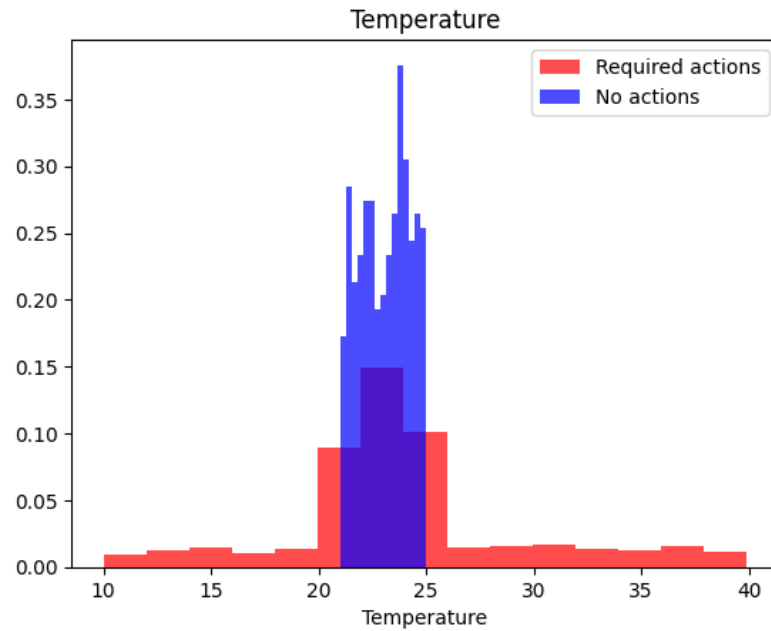


Рисунок 6.1 – Залежність температур в наборі

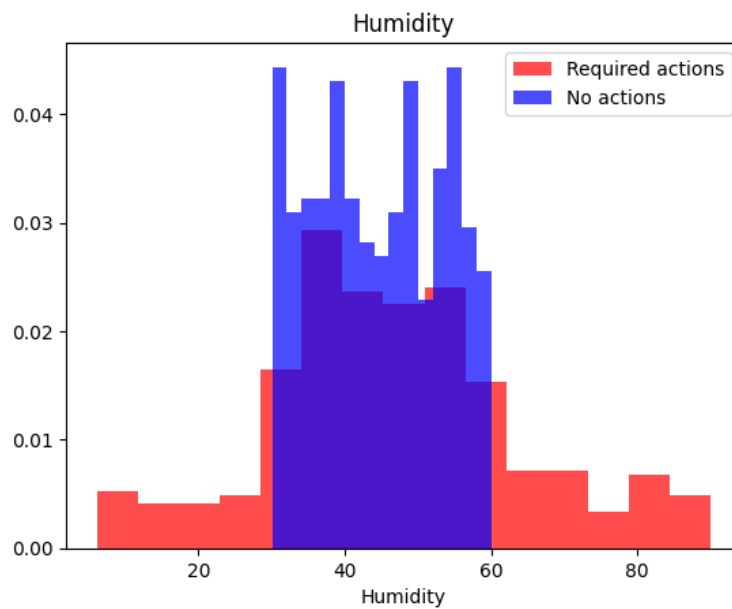


Рисунок 6.2 – Залежність значень вологості в наборі

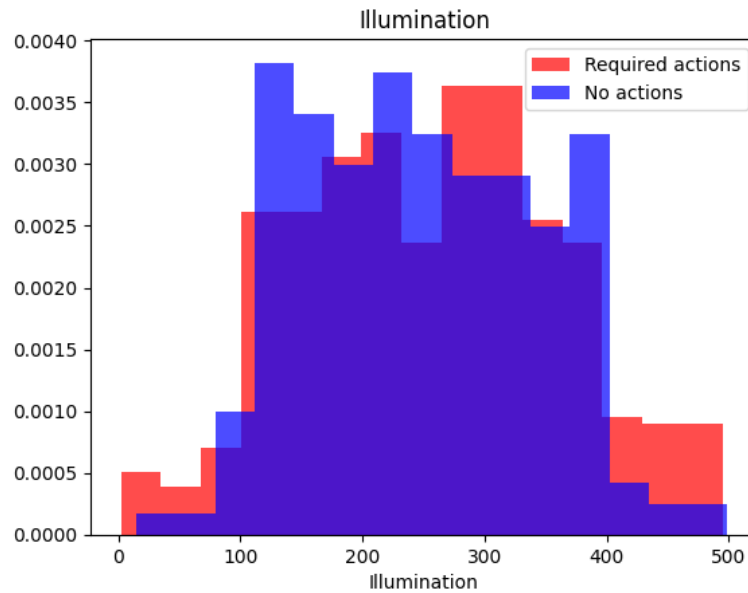


Рисунок 6.3 – Залежність значень рівня освітленості в наборі

На Рисунку 6.1 можна побачити залежність виходу від значень температур, де валідними вважаються 21-25 С°, для вологості це приблизно 30-60%, а для рівня освітленості 10-500, враховуючи, що це звичайне житлове приміщення, то показники відповідають дійсності.

Також датасет має також мати набір значень з виключеннями описаними власноруч. Результуючий набір тренувальних даних можна представити у вигляді таблиці.

Таблиця 6.1 – Частина тренувальної вибірки

Temperature, С°	Humidity %	Illumination, lux	PartOfTheD ay	IsActionRequir ed
30.72	48.71	362	4	1
22.04	34.19	309	4	0
21.94	44.09	274	3	0

22.31	84.68	176	3	1
-------	-------	-----	---	---

Для подальшого використання тренувальної вибірки вона має бути нормалізована та приведена до одного типу, в даному випадку float.

Лістинг 6.2 – Фрагмент коду для нормалізації вибірки

```
public static NormalizedDataSet GetNormalizedTrainingData(string csvFilePath = «»)
{
    if (string.IsNullOrEmpty(csvFilePath))
    {
        csvFilePath = _defaultTrainDataPath;
    }
    using (StreamReader reader = new StreamReader(csvFilePath))
    using (CsvReader csv = new CsvReader(reader, new
CsvConfiguration(CultureInfo.InvariantCulture)))
    {
        var records = csv.GetRecords<EnvironmentIndicatorRow>()
            .ToArray();
        int numRows = records.Length;
        int numCols = 4;
        float[,] X = new float[numRows, numCols];
        for (int i = 0; i < numRows; i++)
        {
            X[i, 0] = records[i].Temperature;
            X[i, 1] = records[i].Humidity;
            X[i, 2] = records[i].Illumination;
            X[i, 3] = (float)records[i].PartOfDay;
        }
        var normalizedDataSet = new NormalizedDataSet
        {
            X = X,
            Y = records.Select(r => r.IsActionRequired ? 1f : 0f)
                .ToArray()
        };
        return normalizedDataSet;
    }
}
```

6.2.2 Навчання нейронної мережі

Для навчання обрано шарову нейронну мережу (Feedforward Neural Network, FFNN). Модель створюється за допомогою бібліотеки Keras.NET. Це API нейронних мереж високого рівня для C# і F# через зв'язок Python, який може працювати поверх TensorFlow, CNTK або Theano, в даному випадку TensorFlow. Дана мережа має три шари:

Перший шар має 32 нейрони, другий 64, обидва шари використовують активаційну функцію ReLU (Rectified Linear Unit), яка активує нейрони, якщо вхідний сигнал більше за нуль і призводить до виходу нуля в іншому випадку.

$$f(x) = x^+ = \max(0, x)$$

Третій шар з одним нейроном використовує активаційну функцію sigmoid, яка призводить до виходу значення між 0 і 1 і використовується для бінарної класифікації.

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

При компіляції у якості функції оптимізації обрано стохастичний градієнтний спуск (Stochastic Gradient Descent). SGD, що є одним з популярних оптимізаторів для навчання нейронних мереж.

$$Q(w) = \frac{1}{n} \sum_{i=1}^n Q_i(w),$$

У якості функції втрат використовується двійкова перехресна ентропія (також відома як логарифмічні втрати або логарифмічні втрати) — це модельна метрика, яка відстежує неправильне позначення класу даних моделлю, штрафуючи модель, якщо в класифікації міток трапляються відхилення в імовірності. Низькі значення втрат журналу прирівнюються до значень високої точності.

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

При навчанні моделі її поділено на пакети по 4 приклади у кожному та 1000 епох. Після чого модель зберігається у json форматі окремо також зберігаються ваги.

Лістинг 6.3 – Фрагмент коду з методом навчання моделі

```
public static void TrainModel()
{
    var normalizedTrainingData = DataSetParser.GetNormalizedTrainingData();
    Nddarray x = np.array(normalizedTrainingData.X);
    Nddarray y = np.array(normalizedTrainingData.Y);

    var model = new Sequential();
    model.Add(new Dense(32, activation: «relu», input_shape: new Shape(4)));
    model.Add(new Dense(64, activation: «relu»));
    model.Add(new Dense(1, activation: «sigmoid»));

    model.Compile(optimizer: «sgd», loss: «binary_crossentropy», metrics: new
string[] { «accuracy» });
    model.Fit(x, y, batch_size: 4, epochs: 1000, verbose: 1);

    string json = model.ToJson();
    File.WriteAllText(_modelPath, json);
    model.SaveWeight(_modelWeightsPath);
}
```

6.3 Класифікація стану середовища

Для більш детального аналізу стану показників середовища реалізовано модель класифікації.

6.3.1 Алгоритм LightGBM

LightGBM є потужним методом ансамблю градієнтного підсилення, який використовується для завдань класифікації та регресії, і базується на деревах рішень. Він є дуже ефективним і оптимізованим для використання в розподілених системах, що робить його привабливим в бізнес-аналітиці та науці про дані [6].

Однією з ключових особливостей LightGBM є те, що дерева зростають, розглядаючи залишок від інших дерев. Тобто, в порівнянні з іншими методами, такими як XGBoost, LightGBM спочатку роздивляється умови для розгалуження, які надають максимальний приріст інформації (зазвичай званий «прирістом»).

Оскільки ростуть листками, дерева LightGBM можуть потенційно бути дуже глибокими і схильними до перенавчання, особливо на невеликих наборах даних. Щоб запобігти цьому, можна обмежити глибину дерева, що допоможе підвищити стійкість до перенавчання.

LightGBM використовує гістограму розподілу даних. Замість розділення кожної окремої точки даних, він використовує гістограму для поділу даних на біну. Це дозволяє оптимізувати обчислення і робить алгоритм ефективним навіть на розріджених даних.

Ексклюзивне об'єднання ознак: Ще однією цікавою особливістю LightGBM є можливість об'єднувати ексклюзивні ознаки для зменшення розмірності даних. Це полегшує обчислення і робить алгоритм більш швидким і ефективним.

Gradient-based One Side Sampling (GOSS) – це метод семплінгу даних, який використовується в LightGBM. Він вагує точки даних з більшими градієнтами вище, що допомагає покращити точність моделі. Також в GOSS деякі точки даних з меншими градієнтами випадково видаляються, а деякі залишаються, щоб зберегти точність моделі.

LightGBM став популярним інструментом у світі машинного навчання завдяки своїй високій продуктивності, здатності робити оптимальний вибір, і підходу, який допомагає уникнути перенавчання. Він часто використовується у задачах аналізу даних, класифікації тексту, розпізнаванні зображень, обробці даних супутникового спостереження та інших галузях науки та бізнесу.

6.3.2 Програмна реалізація модулю класифікації

Після початкового аналізу стану показників середовища в приміщенні, якщо стан є незадовільним, то є необхідність проаналізувавши поточні показники виконати необхідні дії. Для цього підходить описаний вище алгоритм LightGBM. Для реалізації такого модулю також потрібен тренувальний набір. Даний набір є схожим з тим, що використано для навчання моделі аналізу стану, але вихідним значенням є одна категорія з набору. В даному випадку, це є рекомендована дія.

Лістинг 6.4 – Список пропонованих дій у вигляді enum

```
public enum DesiredAction
{
    None = 0,
    IncreaseTemperature = 1,
    DecreaseTemperature = 2,
    TurnOnLight = 3,
    TurnOffLight = 4,
    TurnOnHumidifier = 5,
    TurnOffHumidifier = 6
}
```

В результаті сформовано .csv файл який має вигляд, показаний в Таблиці 6.2.

Таблиця 6.2 – Фрагмент датасету для класифікації рекомендованих дій

Temperature, C°	Humidity %	Illumination, lux	PartOfTheDay	DesiredAction
39.89	52.75	255	4	2
24.19	28.8	248	4	5
21.4	45.62	304	2	0
23.28	45.64	268	4	4

Пори тренуванні такої моделі також потрібно нормалізувати вхідну модель, для цього використано бібліотеку ML.NET, яка надає широкий функціонал для машинного навчання. Але на виході буде отримано модель з вірогідністю рекомендованої дії та інші інформаційні поля. Під час навчання формується пайплайн з кількістю листків 24, 7 дерев та 1022 біни.

Лістинг 6.5 – Метод створення пайплайну

```
public static IEstimator<ITransformer> BuildPipeline(MLContext mlContext)
{
    var pipeline = mlContext.Transforms.ReplaceMissingValues(new []
    {
        new InputOutputColumnPair(@"Temperature", @"Temperature"),
        new InputOutputColumnPair(@"Humidity", @"Humidity"),
        new InputOutputColumnPair(@"Illumination", @"Illumination"),
        new InputOutputColumnPair(@"PartOfTheDay", @"PartOfTheDay")
    })
    .Append(mlContext.Transforms.Concatenate(@"Features", new []
    {
        @"Temperature",
        @"Humidity",
        @"Illumination",
        @"PartOfTheDay"
    })))
    .Append(mlContext.Transforms.Conversion.MapValueToKey(outputColumnName:@"Desired
Action",inputColumnName:@"DesiredAction"))
    .Append(mlContext.MulticlassClassification.Trainers.OneVersusAll(binaryEstimator
:mlContext.BinaryClassification.Trainers.FastTree(new FastTreeBinaryTrainer.Options()
{
    NumberOfLeaves=24,
    MinimumExampleCountPerLeaf=2,
    NumberOfTrees=7,
    MaximumBinCountPerFeature=1022,
    FeatureFraction=0.929154850294031,
    LearningRate=0.225569522574451,
    LabelColumnName:@"DesiredAction",
    FeatureColumnName:@"Features"
})),labelColumnName: @"DesiredAction"))
    .Append(mlContext.Transforms.Conversion.MapKeyToValue(outputColumnName:@"Predict
edLabel",inputColumnName:@"PredictedLabel"));
    return pipeline;
}
```

В результаті 20 хвилинного тренування можна отримати готову модель з високим коефіцієнтом правильних предікшенів, яка також може бути збережена та використана в подальшому.

Top 5 models explored					
	Trainer	MicroAccuracy	MacroAccuracy	Duration	#Iteration
89	FastTreeOva	0.8667	0.9036	8.6	89
31	FastTreeOva	0.8536	0.8920	4.4	31
66	FastForestOva	0.8525	0.8905	6.5	66
96	FastTreeOva	0.8513	0.8923	7.9	96
14	FastTreeOva	0.8504	0.8921	3.1	14

Рисунок 5.1 – Точність навчання

7 РЕАЛІЗАЦІЯ СИСТЕМИ РОЗУМНОГО БУДИНКУ НА БАЗІ НЕЙРОННОЇ МЕРЕЖІ

7.1 Загальна структура системи

Дана система складається з чотирьох основних модулів (рисунок 7.1).

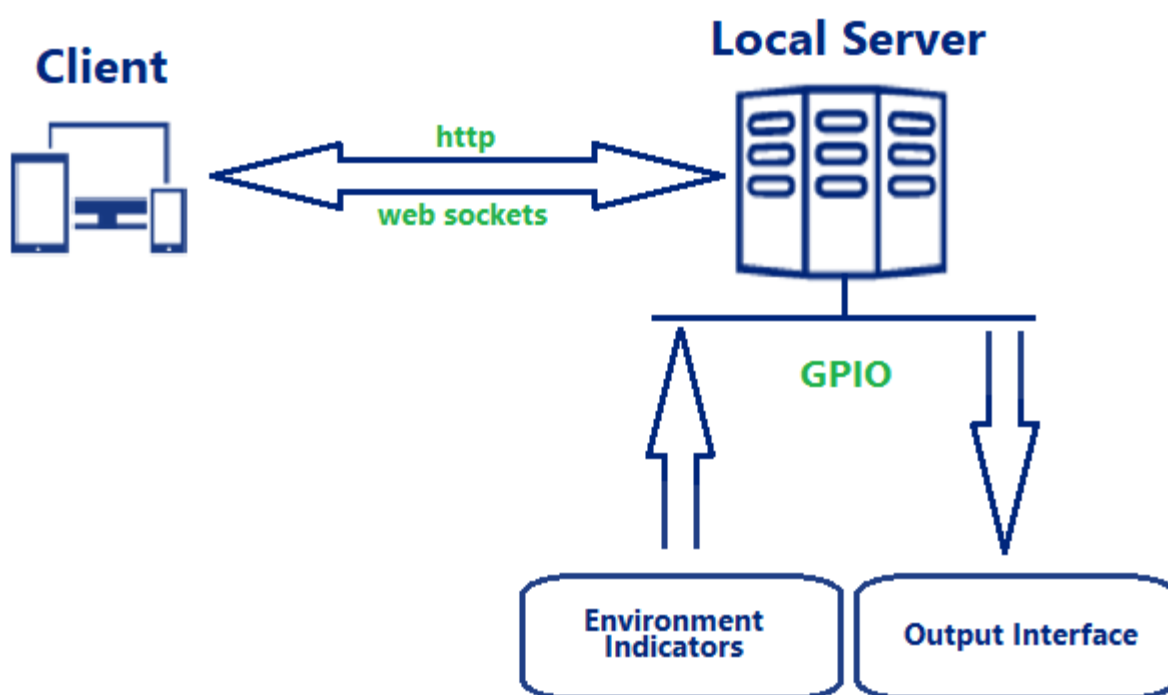


Рисунок 7.1 – Загальна структура системи розумного будинку

Основною частиною є локальний сервер, який виконує опитування датчиків середовища, проводить аналіз, класифікацію з використанням описаної вище нейронної мережі, оброблює запити клієнтської частини та може керувати вихідними пристроями підключеними до системи. Клієнтська частина у вигляді веб-застосунку запитує поточні данні, отримує нотифікації та виконує налаштування. До моніторингового модулю входять

цифрові індикатори освітлення, температури та вологості. У вихідному модулі можуть бути задіяні будь-які цифрові пристрої, які можуть впливати на показники індикації середовища. Юзер може налаштувати відправку повідомлень, або автоматичні дії у разі незадовільних показників.

Дана система розгорнута на базі одноплатного комп'ютера Raspberry Pi 4.

7.2 Raspberry Pi

Raspberry Pi містить усе, що потрібно для роботи комп'ютера – лише на одній платі. Графічний і центральний процесор знаходяться в одній інтегральній схемі. Інші компоненти, включаючи USB-порт, оперативну пам'ять і слот для SD-карти, припаяні. SD-карта зазвичай використовується для зберігання операційної системи та, можливо, деяких інших файлів.

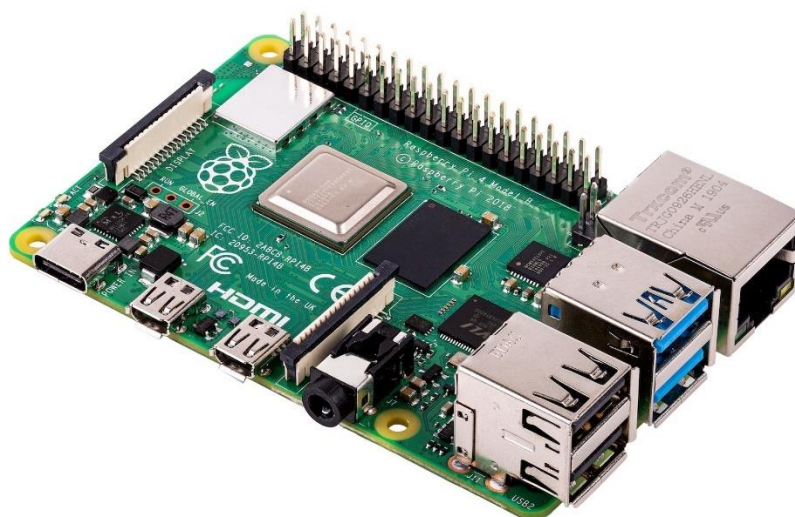


Рисунок 7.2 – Raspberry Pi 4

Фізичні характеристики Raspberry Pi відносно персональних комп'ютерів [7]. Це просто полегшена копія, але все ще надає стільки функцій, скільки хоче користувач. Якщо є необхідність додати нові модулі,

такі як додатковий слот USB, це простий випадок припаювання його до друкованої плати.

Будь-яка модель Raspberry Pi працює так само, як і будь-який інший комп'ютер. Все, що йому потрібно, це джерело живлення, дисплей та інтерфейс для вводу/виводу.

Однією з потужних особливостей Raspberry Pi є ряд контактів GPIO уздовж верхнього краю плати. GPIO означає введення/виведення загального призначення. Ці контакти є фізичним інтерфейсом між Raspberry Pi і зовнішнім оточенням. На найпростішому рівні їх можна розглядати як перемикачі.

Виводи GPIO дозволяють Raspberry Pi контролювати зовнішні пристрої, підключаючись до електронних схем. Pi може керувати світлодіодами, вмикати та вимикати їх, запускати двигуни та багато іншого. Він також може визначити, чи був натиснутий перемикач, температуру та світло. Ми називаємо це фізичними обчисленнями.

На Raspberry Pi є 40 контактів (26 контактів на ранніх моделях), і вони забезпечують різні функції.

7.3 Бекенд частина

У якості платформи для розробки бекенд частини обрано ASP.NET Core. Він дозволяє створювати масштабні та гнучкі проекти. Структура побудована за патерном MVC, тобто представлена трьома рівнями:

- рівень моделі. Відповідає за збереження даних;
- рівень бізнес логіки. Виконує основний функціонал обчислення формування даних;
- рівень представлення. Відповідає за функцію передачі даних клієнтській частині.

Для гнучкості і масштабованості розробки використано техніку DI (Dependency Injection), де створюється слабка залежність реалізації через

інтерфейси, що надає можливість простіше модифікувати існуючу логіку.

Лістинг 7.1 – Ін'єкція залежностей

```
builder.Services.AddScoped<IPredictionService, PredictionService>();  
builder.Services.AddScoped<IDeviceService, DeviceService>();  
builder.Services.AddScoped<ISettingsService, SettingsService>();
```

7.3.1 ASP.NET Core

ASP.NET Core, фреймворк веб-розробки з відкритим вихідним кодом, спеціально створений для створення сучасних хмарних програм. Він може похвалитися декількома перевагами порівняно зі своїм попередником, оригінальною платформою ASP.NET, наприклад підвищеною продуктивністю, підтримкою кількох платформ і спрощеним процесом розробки. На відміну від попередніх версій, які були обмежені для Windows, ASP.NET Core тепер є кросплатформною, розширюючи свою сумісність із macOS і Linux [8].

Початкова версія ASP.NET Core, представлена в 2016 році, ознаменувала собою комплексну переробку широко використовуваної системи веб-розробки Microsoft, ASP.NET. Ціллю цього редизайну було створення більш ефективної та модульної структури, адаптованої для розробки сучасних хмарних додатків.

Основні функції ASP.NET Core включають:

- сучасне міжплатформне середовище виконання, сумісне з Windows, Linux, Docker і macOS;
- підвищена продуктивність завдяки новому модульному дизайну, що сприяє покращенню кешування та оптимізації;
- спрощена розробка через уніфіковану структуру, яка об'єднує MVC і Web API в єдиний конвеєр;
- представлення нової моделі програмування Razor Pages, що спрощує розробку сторінково-орієнтованих програм.

7.3.2 Взаємодія з іншими компонентами системи

Основним протоколом взаємодії з клієнтською частиною є HTTP протокол, а форматом даних є JSON.

Лістинг 7.2 – Фрагмент методу контролера Device

```
[Route("GetTemperature")]
[HttpGet]
public IActionResult GetTemperature()
{
    var result = _deviceService.GetTemperatureValue();
    return Ok(result);
}
```

Для двостороннього зв'язку використано протокол WebSocket, що дозволяє серверу самому ініціювати відправку даних. В данному випадку використовується для відправки нотифікацій клієнту у випадку, якщо стан індикаторів навколишнього середовища незадовільний і потрібні додадкові дії. Фрґменти створення зв'язку та відпрака повідомлення наведені в Лістингу 7.3.

Лістинг 7.3 – Ініціація створення нового зв'язку

```
app.Use(async (http, next) =>
{
    if (http.WebSockets.IsWebSocketRequest)
    {
        var websocket = await http.WebSockets.AcceptWebSocketAsync();
        while (websocket.State == WebSocketState.Open)
        {
            MyWebSocketManager.AddWebSocket(websocket);
            var token = CancellationTokens.None;
            var buffer = new ArraySegment<byte>(new byte[4096]);
            var received = await websocket.ReceiveAsync(buffer, token);
        }
    }
    else
    {
        await next();
    }
});
```

7.4 Аналіз стану показників середовища

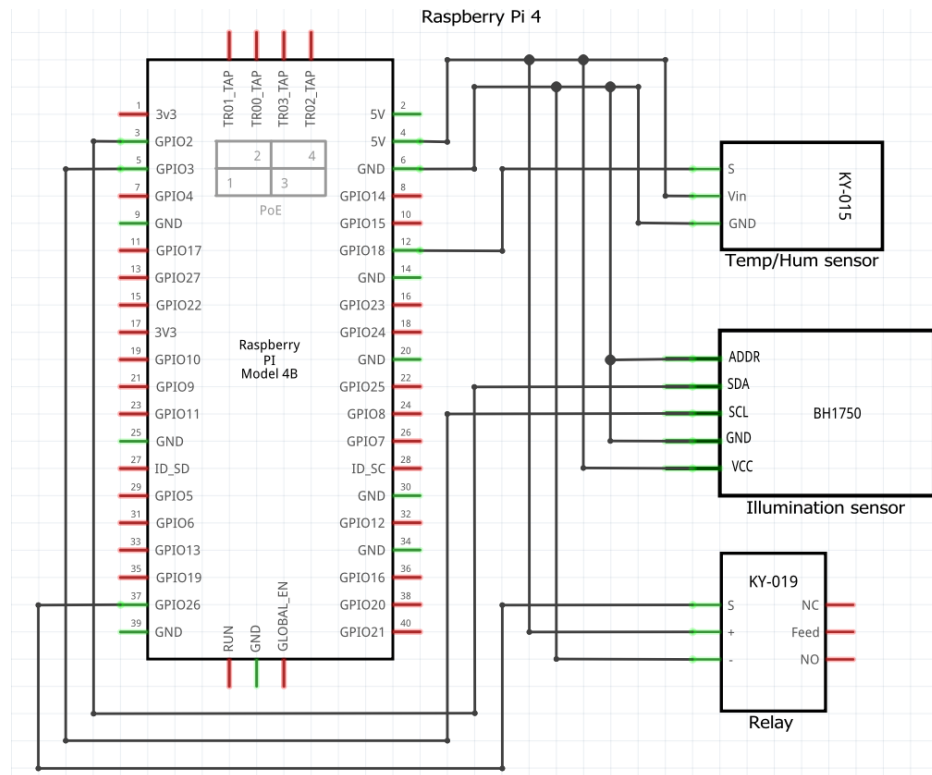


Рисунок 7.2 – Схема з'єднання пристроїв до плати

На Рисунку 7.2 представлена схема підключення цифрових датчиків температури/вологості, освітленості та реле для управління світлом.

Перевірка значень датчиків навколишнього середовища відбувається за допомогою періодичного опитування цифрових значень датчиків підключених до інтерфейсу GPIO. Для повторного виконання однакових дій використано бібліотеку Quartz, що дозволяє створювати джоби, які виконують певні дії з інтервалом або в конкретний момент часу, має дуже гнучкий функціонал для налаштування.

Лістинг 7.4 – Ініціація джоби для моніторингу датчиків середовища

```
IScheduler scheduler = await StdSchedulerFactory.GetDefaultScheduler();
await scheduler.Start();
```

```
IJobDetail environmentValuesReaderJob =
```

```

JobBuilder.Create<EnvironmentValuesReaderJob>().Build();

ITrigger environmentValuesReaderJobTrigger = TriggerBuilder.Create()
    .WithIdentity("environmentValuesReaderJobTrigger", _groupName)
    .StartNow()
    .WithSimpleSchedule(x =>
x.WithIntervalInMinutes(StorageService.EnvironmentValuesCheckInterval).RepeatForever()
)
    .Build();

```

Для взаємодії з фізичними пінами Raspberry Pi використано бібліотеку Device.Gpio, що дозволяє опитувати стан цифрових значень підключених пристроїв та генерувати вихідні значення а також надає готові класи, що представляють абстрактний опис існуючих датчиків за їх маркуванням.

Лістинг 7.5 – Фрагмент коду зчитування показників освітлення

```

var bh1750fviTask = Task.Run(async () =>
{
    using (Bh1750fvi sensor = new Bh1750fvi(_illuminationSensorDevice))
    {
        double illuminationValue = 0;
        for (var i = 1; i < StorageService.PinCheckCount; i++)
        {
            Illuminance illumination = sensor.Illuminance;
            if (illumination.Value > 0)
            {
                illuminationValue = illumination.Value;
            }
            Console.WriteLine($"BH1750: {illumination.Value}");
            await Task.Delay(StorageService.PinCheckDelay);
        }
        StorageService.IlluminationQueue.Enqueue(illuminationValue);
    }
});

```

7.4.1 Датчик температури та вологості. KY-015

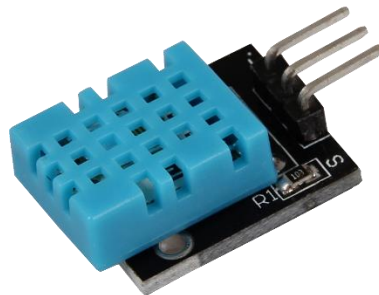


Рисунок 7.3 – KY-015

Модуль KY-015 складається з цифрового датчика вологості та температури DHT11 та резистора 1 кОм. DHT11 використовує внутрішній термістор та ємнісний датчик вологості для визначення умов навколишнього середовища, а внутрішній чіп відповідає за перетворення показань у послідовний цифровий сигнал.

Таблиця 7.1 – Технічні характеристики KY-015

Робоча напруга	від 3,3 до 5,5 В
Діапазон вимірювання вологості	від 20% до 90%
Точність вимірювання вологості	$\pm 5\%$
Дозвіл вимірювання вологості	1%
Діапазон вимірювання температури	від 0 °С до 50 °С
Точність вимірювання температури	$\pm 2\text{ °С}$
Дозвіл вимірювання температури	1 °С
Дальність передачі сигналу	20 м

7.4.2 Датчик освітленості. BH1750

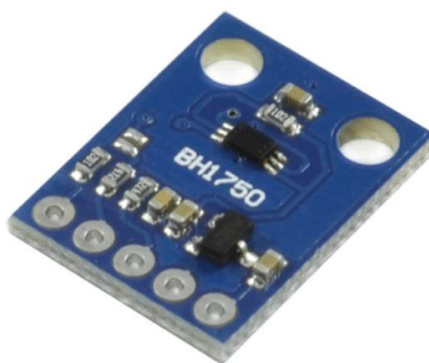


Рисунок 7.3 – BH1750

BH1750 це простий цифровий чіп датчика освітлення з I2C

інтерфейсом від японської компанії Rohm Semiconduct. Фотодіод виробляє напругу, напруга посилюється, надходить на аналого-цифровий перетворювач. Після ADC цифровий сигнал надходить у регістровий банк I2C. З регістрового файлу значення вираховується мікроконтролером I2C. У середині всього один 2-байтний регістр у форматі big endian. Всю схему тактує вбудований осцилятор на 320kHz.

Таблиця 7.2 – Технічні характеристики BH1750

Діапазон вимірів освітлення	1 - 65535 Lx
Напруга живлення	2.4 - 3,6 V
I2C rate	0< - 400 kHz

7.5 Фронтенд частина

Клієнтська частина застосунку представляє собою веб-застосунок та створена на базі фреймворку Angular [9]. Angular – це популярна інтерфейсна платформа веб-додатків з відкритим кодом, яку підтримує Google і спільнота розробників. Він написаний на TypeScript і дозволяє розробникам створювати динамічні односторінкові веб-програми (SPA). Застосунок є адаптивним і може бути використаний на пристроях з різними параметрами екрану. У якості підсилення вигляду використано Angular Materials, що дозволяє використовувати вже готові компоненти, анімації, діалогові вікна, що полегшує роботу з стилями застосунку.

Для більш швидкого завантаження застосунку розділено на модулі, що надає можливість швидше завантажувати основний модуль та відображати потрібний контент швидше. Модулі фактично представляють собою сторінки з меню навігації на кожній.

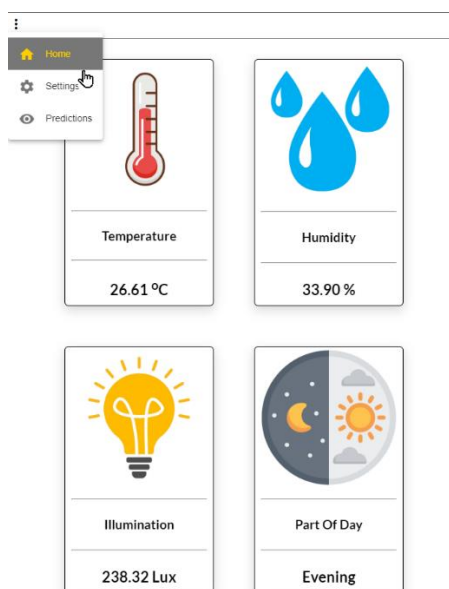


Рисунок 7.4 – Основна сторінка

Основна сторінка представляє собою список поточних показників, при натисканні на кожен картку можна отримати більш детальну інформацію останніх значень.

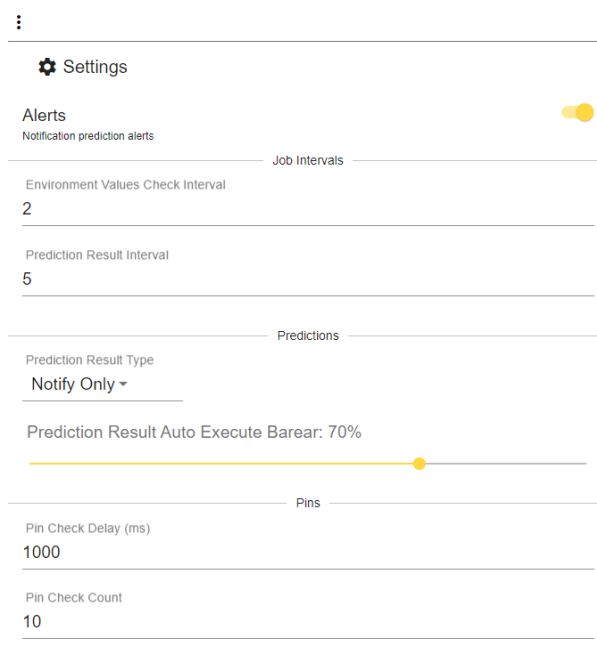


Рисунок 7.5 – Сторінка налаштувань

На сторінці налаштувань користувач може встановити бажані параметри роботи системі такі як: отримання пуш повідомлень, частоту опитування датчиків та частоту автоматичних класифікацій стану та поріг активації, тощо.

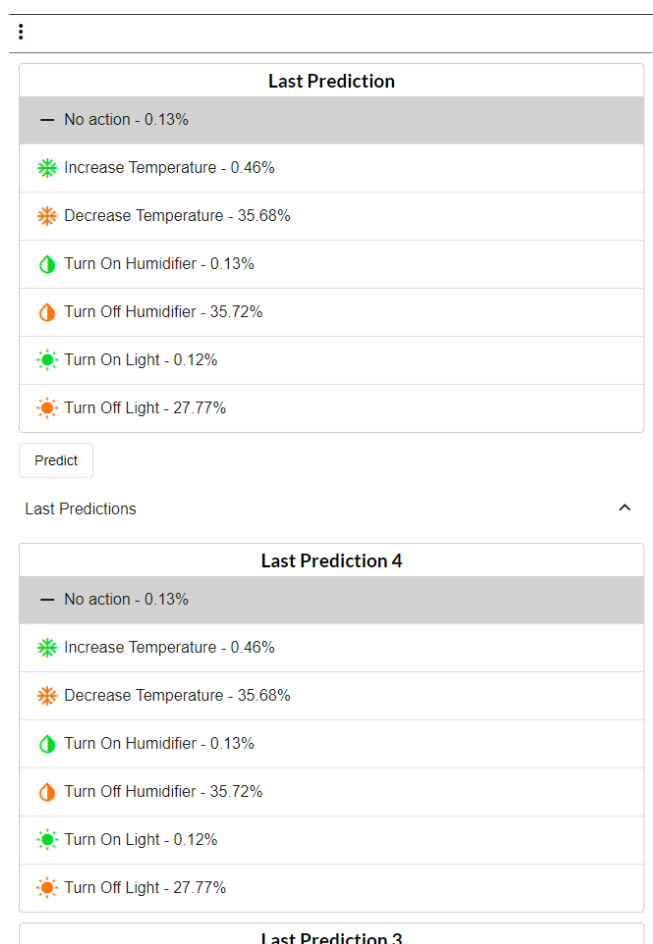


Рисунок 7.6 – Сторінка аналізу показників

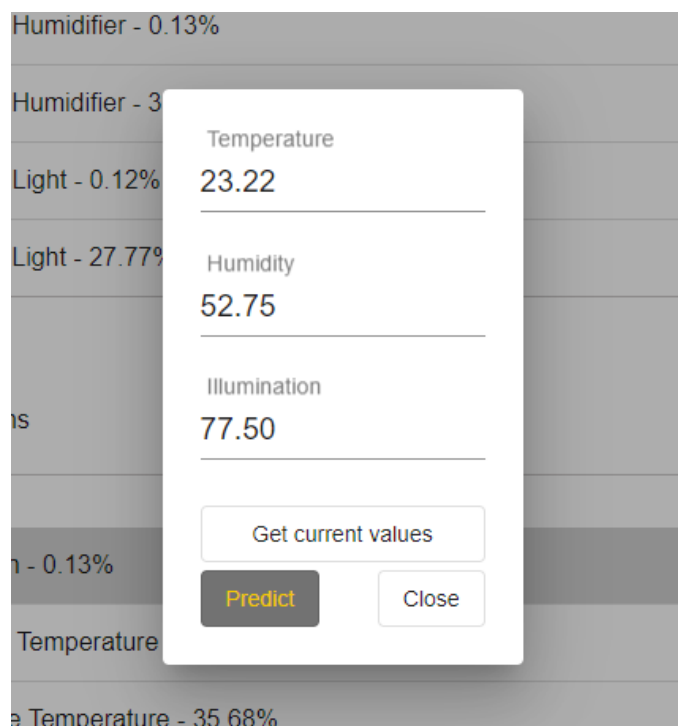


Рисунок 7.7 – Форма для передбачення власноруч

Сторінка аналізу показників показників дозволяє переглянути останні передбачення зроблені системою у відсотковому співвідношенні. Користувач може обрати одну з наведених опцій, після чого може підтвердити дію через модальне вікно. Також надано можливість робити передбачення власноруч за поточними або власними показниками оточення (рисунок 7.7). Також при наявності нових передбачень користувач побачить повідомлення в застосунку або нотифікацію (рисунок 7.8).

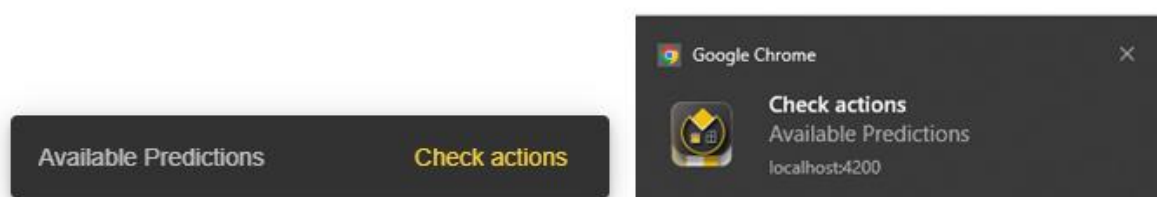


Рисунок 7.8 – Види нотифікацій користувача

ВИСНОВКИ

У ході дослідження було проведено огляд використання нейронних мереж у системах керування розумним будинком. Даний напрямок має великий потенціал у поліпшенні функціональності та ефективності таких систем.

Застосування нейронних мереж у системах керування розумним будинком може призвести до значного покращення енергоефективності. Нейронні мережі здатні аналізувати дані про споживання енергії та зовнішні фактори, що дозволяє розробити оптимальні стратегії регулювання систем опалення, кондиціонування повітря, освітлення та інших електричних пристроїв. Це не тільки знижує витрати на енергію, але й забезпечує комфортні умови для мешканців.

Нейронні мережі також відіграють важливу роль у покращенні безпеки та захисту в системах розумного будинку. Вони можуть виявляти незвичну або підозрілу активність на основі аналізу образів та поведінки. Це дозволяє розпізнавати небезпечні ситуації та сповіщати про можливі загрози.

Важливо зазначити, що використання нейронних мереж у системах розумного будинку також стикається з певними викликами та обмеженнями. Одним з них є необхідність великого обсягу даних для тренування та налагодження мережі. Також важливо враховувати проблеми приватності та захисту даних, оскільки нейронні мережі можуть працювати з чутливою інформацією про мешканців.





Загалом, на основі проведеного дослідження можна зробити висновок, що застосування нейронних мереж у системах керування розумним будинком є перспективним напрямком розвитку. Вони забезпечують покращення енергоефективності, безпеки та комфорту для мешканців. Проте, для успішної реалізації таких систем необхідно вирішити проблеми, пов'язані з обробкою великого обсягу даних, приватністю та захистом інформації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices / Andrew Minter. – Бермінгем: Packt Publishing, 2017 – 378 с.
2. Smart Home Market Size, Share & Growth [Електронний ресурс] – Режим доступу : www / URL: <https://www.grandviewresearch.com/industry-analysis/smart-homes-industry> (дата звернення: 27.04.2023).
3. The Evolution Of Smart Home Technology [Електронний ресурс] – Режим доступу : www / URL: <https://uksmarthomes.co.uk/the-evolution-of-smart-home-technology/> (дата звернення: 27.04.2023).
4. Нейронні мережі / Саймон Хайкін – Вільямс, 2020 – 1104 с.
5. Дослідження алгоритмів навчання нейронних мереж в задачах прогнозування/ І. О. Калініна /Наукові праці [Чорноморського державного університету імені Петра Могили]. Комп'ютерні технології, – 2009. – Т. 117, Вип. 104. – С. 160-171. Режим доступу : www / URL: <https://lib.chmnu.edu.ua/pdf/naukpraci/computer/2009/117-104-16.pdf>
6. Bhave R. Practical Machine Learning with LightGBM and Python: Explore Microsofts Gradient Boosting Framework to Optimize Machine Learning. Packt Publishing, Limited, 2021. 228 с.
7. Fontichiaro K., Severance C. R. Raspberry Pi. Cherry Lake Publishing, 2013.
8. Esposito D. Programming ASP. NET Core, Programming ASP. NET Core. Microsoft Press, 2018.
9. Freeman A. Your First Angular App. Pro Angular. Berkeley, CA, 2017. С. 5–30. URL: https://doi.org/10.1007/978-1-4842-2307-9_2 (дата звернення: 26.12.2023).

Відомість кваліфікаційної роботи

«Система розумного будинку в автономному режимі на основі нейронної мережі керування»

	Прізвище та ініціали відповідальної особи	Підпис	Дата
<p>Роботу виконав студент групи СКСм-22-1</p> <p>Структура кваліфікаційної роботи: – пояснювальна записка <u>78</u> с.; – графічний матеріал <u>17</u> арк..</p>	Солодовник Ю. Ю.		12.01.24
Керівник роботи	Немченко В.П.		12.01.24
<p>Перевірка на плагіат здійснена. Оригінальність авторського тексту складає <u>95</u> %</p>	Литвинова Є.І.		12.01.24
Нормоконтроль проведено :	Немченко В.П.		12.01.24